
SCANCO MEDICAL

Finite Element-software

Version 1.03

SCANCO Medical AG
Auenring 6-8
8303 Bassersdorf
Switzerland
tel. +41 1 837 0710
fax. +41 1 837 0713

SCANCO USA Inc.
10 Lantern Lane
Wayne, PA
19087
USA
tel/fax +1 910 296 2056

July 2003

Main features.....	4
Typing conventions for this document	4
Installation	4
Quick start (VMS:).....	4
Introduction	5
Section A: Preparations	5
A.1. Selection of a volume of interest (VOI).....	5
A.1.1 VOI for the characterization of continuum level mechanical behavior in relation to macro-morphology.....	5
A.1.2 VOI for tissue-level stress/strain analyses	6
A.2. Image processing	7
A.2.1 Voxel size and accuracy.....	7
A.2.2 Filtering and segmentation.....	8
A.2.3 Removal of unconnected parts	8
A.2.4 Component labeling	8
Section B: Preprocessing and solving	9
B.1 Starting the program	9
B.1.1 Interactive.....	9
B.1.2 Batch/background mode.....	9
B.2 Input parameters	11
B.2.1 Pre-defined standard problems.....	13
B.2.2 Calculation of overall properties of cubic volumes of interest (VOI's).....	17
B.2.3 Boundary conditions and cputime.....	18
B.3 Input files	18
B.3.1 The <i>fea_file_name.bcd</i> file	18
B.3.1.1 The code parameter	18
B.3.1.2 The range parameters	19
B.3.1.3 The value coefficients.....	20
B.3.1.4 Format of the <i>fea_file_name.bcd</i> file	21
B.3.1.5 Examples:	22
B.4 Output files	23
B.4.1 The <i>fea_file_name.listing</i> file	23
B.4.2 The <i>fea_file_name-i.restart</i> file.....	24
B.4.3 The <i>fea_file_name.post</i> file.....	24
Section C: Post-processing	25
C1. Starting the program	25
C.1 Input parameters	25
1.1 Choice of mechanical parameters	26
C.2 Input files	28
C.2.1 The <i>post_file_name.post</i> file	28
C.3 Output files	28
C.3.1 The <i>post_file_name.postlist</i> file	28
C.4 (VMS:) Output IPL objects.....	32
C.4.1 The output AIM object.....	32
Section D: Tools.....	32
D.1 Examine post and restart files.....	32
Section E: advanced options.....	32
E.1 the <i>Scanco_fe.ini</i> file	32
E.2 Exchange of binary files between platforms	33
References	34

Appendix A. (VMS:) Logicals and symbols defined for the FE-program.....	35
Appendix B. (VMS:) Using the IPL_FE_VM.COM command file for batch processing.....	36
Appendix C. (UNIX:) Sample input file for the definition of input parameters.....	38
Appendix D. Boundary condition definitions for the predefined standard problems	39
Appendix E. Sample listing file.....	43
Appendix F. Sample postlist file	45
Appendix G. Sample scanco_fe.ini file (with VMS settings).....	47
Parameter index	48

Main features

- Fast solver for the calculation of structure-dependent elastic properties of voxel structures
- FE-models created directly from voxel data in a fully automated way
- Fully integrated in IPL
- Contains a library with boundary conditions for the simulation of standard tests
- Can solve models with on the order of 1M elements (theoretical max. 715M)
- Models can obtain up to 255 different isotropic elastic materials
- In- and output files can be exchanged between platforms, enabling solving on large parallel UNIX systems and preprocessing/visualization on VMS workstations

Typing conventions for this document

In this document, the following typing conventions are adopted.

<i>Courier</i>	name of a parameter that must be replaced by a user defined name or value
Arial	name of a program variable
<i>italic</i>	string that must be typed literally as printed
[unit]	square brackets indicate units (e.g. [MPa], [mm])
(VMS:) text in blue	represents platform-specific options. The platform for which this option is available will be indicated before the text in brackets.

Installation

(VMS:)

With a standard installation the logicals and symbols required by the FE-software are defined in the file DISK1:[MICROCT]MICROCT_FE.COM which is called by the file DISK1:[MICROCT]LOGIN.COM. An overview of logicals and symbols defined for the FE-solver is given in appendix A.

The FE-software requires two additional files: SCANCO_FE.INI and SCANCO_FE_STANDARD.BCD that must be stored in the directory defined by the SCANCO_FE_DIR logical. With a standard installation this logical points to the directory DISK1:[MICROCT.DATABASE]. It is possible to define another location for the SCANCO_FE_STANDARD.BCD file (see section D).

In order to run the program in batch-mode, two extra command files are required. The first file: DISK1:[MICROCT.MAIN]UCT_FE_ANALYSIS.COM contains the commands to submit the job to the batch queue. The second command file: DISK1:[MICROCT.MAIN]IPL_FE_VM.COM contains the IPL commands for the FE-analysis. The latter file needs to be edited to set the right parameter values.

(SGI_UNIX:)

There is no version of IPL for this platform. Instead, the FE-solver is supplied as a separate stand-alone program. The IPL commands are replaced by executables with the same name: “fe_solve”, “fe_post” and “fe_exa”. In the following it is assumed that these files are stored in a directory \$HOME/scanco/fe_64/, but other directories are allowed as well. The FE-software requires the definition of an environment variable SCANCO_FE_DIR that points to the directory where two files required by the program: “scanco_fe.ini” and “standard.bcd” can be found. It is recommended to store all files in the same directory in which case this variable should be defined by typing: *setenv SCANCO_FE_DIR \$HOME/scanco/fe_64/*. In order to execute the programs, the directory where the executable can be found should be included in the path or the full path should be included with the name of the executable. For example, the solver can be executed by typing: *\$HOME/scanco/fe_64/fe_solve*.

Quick start (VMS:)

Open a DECTERM window.

Within this window, type: “iplfe”

Within IPL:

- read in an char-type aim file using the *read* command
- start the FE-solver using the *fe_solve* command
- when the calculations are done, start the postprocessing with the *fe_post* command
- save the generated ‘con’ object to an aim file using the *write* command
- exit IPL

Use an editor to read the contents of the generated listing and postlist files

Use the 3D visualization program to visualize the generated aim file

Introduction

The IPL finite element (FE) software enables the calculation of structure-dependent elastic properties of voxel structures. It can be used for two purposes. First, to simulate experimental tests for the calculation of overall (apparent) properties of the structure. As an example, it can be used to simulate a compression test on a bone specimen to calculate the apparent stiffness of the specimen. The program has a library from which the boundary conditions for standard experimental tests can be chosen, but it is also possible to specify user-defined test conditions. Second, to calculate tissue-level mechanical loading conditions (e.g. stresses, strains) for external loads working on the structure. For example, the software can be used to calculate the tissue-level stresses and strains in a bone that is loaded by an external force at the joint surface.

FE-analysis within IPL consists of two steps. During the first step (involving 'preprocessing' and 'solving') selected voxels are converted to brick elements in a finite element model and a finite element problem with boundary conditions as specified by the user is solved. The results of this step, the calculated deformation field and reaction forces, are stored in a separate file. During the second step ('post processing') this file will be read to calculate a user selected mechanical parameter (e.g. stress or strain), which is accordingly stored in an AIM object. This AIM object can be further analyzed using routines build into IPL and can be displayed using the 3D visualization tool.

The software employs the so-called 'voxel conversion technique' to create finite element models from voxel data¹. With this technique voxels in the model are converted to equally shaped brick elements such that the resulting FE-model has exactly the same geometry as the reconstruction it is derived from. The elements in the resulting FE-model can represent up to nine different isotropic elastic materials. This enables, for example, the specification of different material properties for voxels representing cortical bone tissue, trabecular bone tissue and cartilage. FE-models generated in this way will generally consist of a very large number of elements (the number of elements is the same as the number of converted voxels). In order to solve such large problems within reasonable limits of cputime and memory, the software implements a special-purpose iterative (conjugate gradient) solver in combination with an element-by-element matrix-vector multiplication scheme². The solving stage will take, by far, most of the cputime and memory. Common problems will contain 0.1M - 10M elements and the required cputime for solving such large problems will usually be on the order of several hours or days whereas memory requirements can be on the order of several hundreds of megabytes. Although the code will permit models up to ~715M elements/nodes a more realistic maximum with present generation workstation will be 1-10M elements (depending on the type of problem and the boundary conditions).

This manual is subdivided in three main sections: 'preparations', 'preprocessing and solving' and 'post processing' and a fourth section describing some advanced options. The first section described the steps that are necessary to create a meaningful model to be used as a basis for the FE-model. The second and third section describe the commands, input and output files used by the FE-software.

Section A: Preparations

A.1. Selection of a volume of interest (VOI)

In the present context, the volume of interest (VOI) is the cubic region defined by the AIM object that is used as a basis for the FE-model. The specimen within the VOI, however, does not have to be cubic. The actual choice of the VOI will depend on the goal of the analysis.

A.1.1 VOI for the characterization of continuum level mechanical behavior in relation to macro-morphology

At the continuum level, also called the apparent level, the structure is considered as a continuous material with material properties that represent the average properties of a representative volume. Such a representative volume must hold a large number of microstructures, in order to provide sufficiently averaged continuum quantities. For cancellous bone a representative volume should be at least five intertrabecular lengths in size (approximately 3-5 mm)³. It is not possible to define meaningful elastic properties for cancellous bone specimens smaller than this minimum size.

The criteria for the VOI for the simulation of compression tests are much the same as those for real test experiments. Accurate and meaningful results can be obtained only if the specimen is reasonably homogenous and if at least two plan-parallel surfaces are available to which the loading can be applied. This

implies that a complete determination of anisotropic properties requires a cubic specimen that can be tested in its three orthogonal directions. In the case of other specimen geometries only a limited number of elastic parameters can be calculated. For example, in the case of cylindrical specimens, meaningful elastic properties can be obtained only for the longitudinal direction.

In order to use the predefined standard problems for the simulation of compression tests, the surfaces that are loaded must coincide with the surfaces of the cubic VOI. If this is not the case, the VOI will be automatically cropped until at least one element connects to each surface. In this situation, however, there is no guarantee that a state of homogenous load transfer is reached and results can be unreliable. Examples of valid and invalid specimen geometries are given in Fig. A.1. Note that complete determination of anisotropic behavior is possible only for specimens that completely occupy a cubic VOI.

A.1.2 VOI for tissue-level stress/strain analyses

In this situation an arbitrary shaped object is located somewhere within the cubic VOI. As an example, the object of interest can be a whole bone reconstructed in a rectangular voxel grid thus leaving a large amount of empty space around the bone. Analyses of this type will generally require the definition of user-defined boundary conditions to specify realistic external loading conditions for the reconstructed object. Although it is possible to specify boundary conditions for individual element faces or nodes, it can be a time consuming process to identify the appropriate elements/nodes and the magnitude/direction of the boundary condition to be applied. In order to simplify this task, it can be useful to have a surface of the VOI intersecting a part of the object to which boundary conditions are applied. In that case boundary conditions can be specified for a smooth surface aligned with the voxel grid, which, can be easily accomplished with the present FE-implementation. As an example, if displacements at the distal end of the proximal part of a femur need to be suppressed, it is useful to crop the original VOI such that one of its faces intersects the distal end of the bone at an appropriate location. Similarly, it can be useful to artificially add a cup or cartilage layer that intersects a face of the VOI to simplify the application of loading conditions to a femoral head (e.g. see reference 4).

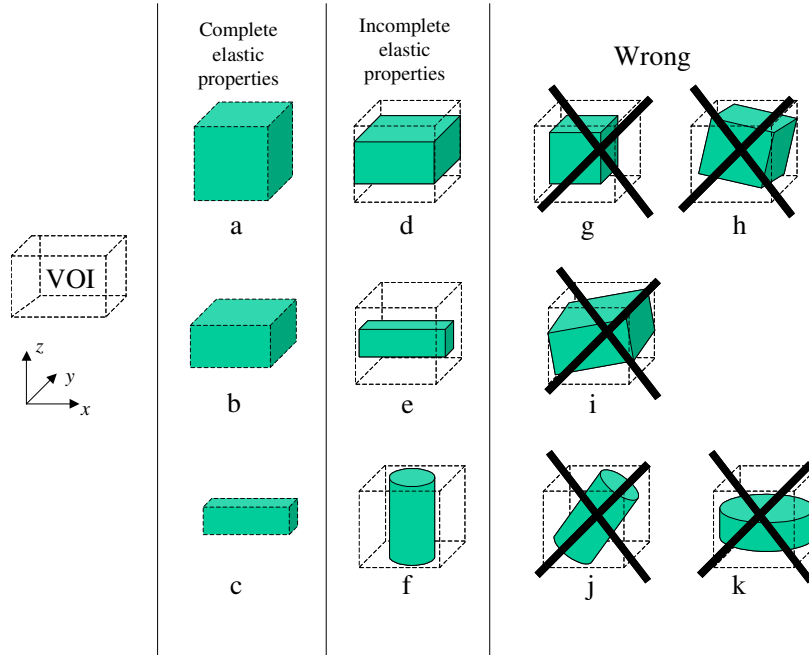


Fig. A.1 In order to use the predefined standard problems for the simulation of compression/tensile tests, the geometry of the specimen within the cubic VOI must be such that at least two plan-parallel opposing surfaces are available that coincide with the surfaces of the VOI to apply the loads. In the case of cubic specimens (a-c) that fill the entire VOI the complete anisotropic elastic properties of the specimen can be calculated. If the specimen occupies a part of the VOI, elastic properties can only be calculated for the directions where opposite faces coincide with the faces of the VOI (x- and y-direction for d, x-direction for e and z-direction for f). Note that a complete calculation of elastic properties can be obtained in these situations by cropping the VOI to the volume occupied by the specimen. In case the specimen is positioned in the VOI such that it has only one or no faces in common with the VOI faces, or if the specimen is rotated, no meaningful elastic properties can be calculated. In these situations, the specimens must be rotated in the VOI and/or the VOI must be cropped to obtain a situation as indicated in cases a-f.

A.2. Image processing

A.2.1 Voxel size and accuracy

In general, results of FE-analyses will be more accurate with decreasing element size. The number of elements in the FE-model, however, largely determines the cputime needed to solve the problem. The choice of the resolution thus will be a trade-off between accuracy and cputime. As a rule of thumb, it has been stated that a FE-model should have at least 4 elements through a cross-section of the micro-structural components that form the structure⁵. For trabecular bone this implies that the element size, and thus the voxel size, should be about 50 microns or less. The effect of element size on the accuracy of the results has been investigated in a several studies^{6, 7}. Generally speaking, and depending on the goal of the analysis, these studies have indicated that for some sites (femur) accurate results can be obtained with larger element sizes (> 100 microns) whereas for other sites results already start to deteriorate at a resolution of 50 microns.

Another issue related to the accuracy of the results is the fact that the created models will have 'jagged' surfaces, which will result in numerical artefacts in the FE-solution. It has been shown that this can lead to oscillating values for the calculated stresses and strains, in particular at the bone surfaces⁸. A reduction of element size will not reduce this error; no matter how small the element are, the surface will remain 'jagged'. Consequently, one should keep in mind that the FE-solution near the surface can be inaccurate due to these numerical artefacts. Although this is usually not a relevant issue for analyses aiming at the determination of continuum-level elastic properties (the results of which are hardly affected by these artefacts due to the averaging of tissue stresses and strains), it can be important when the goal is the determination of tissue level

loading conditions. It is important to note, however, that the accuracy of histograms of the stress/strain distribution in the material will be improved with decreasing element size since, with decreasing element size a smaller fraction of the elements will be located near the surface and thus less elements are subject of these artefacts^{1, 4}. It is noted as well that the effect of this artefact can be somewhat reduced during the post-processing stage using the interpolation option (see section C.1).

In order to reduce the number of elements in a model, to a reasonable number (~ 1M elements) it can be necessary to increase the voxel size by resampling the reconstruction data. To avoid aliasing artifacts, it is best to apply the resampling procedure on the unsegmented data. Resampling can be done within IPL using the *scale_elsize* command.

A.2.2 Filtering and segmentation

After choosing an appropriate voxel size, the usual Gauss filtering procedure can be applied to reduce noise artifacts. Filtering can be done using the IPL *gauss_lp* command. The choice of the filter parameters is largely dependent on the amount of noise in the data and no general guidelines can be given other than that care should be taken that the data does not become too much blurred.

An initial FE-model is then obtained by segmentation of the data using the IPL *threshold* command. The threshold value can have an important effect on the volume fraction of the generated model. Since the volume fraction is the first determinant for the elastic properties, the choice of the threshold value can also have a large effect on the results of the FE-analyses. Consequently, the threshold level must be chosen carefully, if possible by comparing the volume fraction of the final model with that measured with another calibrated technique (e.g. Archimedes principle or QCT measurements).

A.2.3 Removal of unconnected parts

The model created after the segmentation step will consist of a 'main structure' and several loosely or unconnected substructures. Loosely connected structures are structures that are connected to the main structure by only one node or two nodes. In theory, such loosely or unconnected substructures are not allowed in FE-models since they lead to a singular stiffness matrix. Due to the use of the iterative solver and an element-by-element matrix-vector multiplication, however, the existence of loosely or unconnected parts does not necessarily pose a problem. As long as these parts are unloaded (i.e. no forces are applied), the problem can still be solved. Nevertheless, there are two good reasons to remove the loosely and unconnected parts from the model. First, since they do not carry load they do not contribute to the stiffness of the model and thus represent a redundant amount of material that can obscure relationships between structure and stiffness. Second, the element-by-element multiplications are performed for these elements as well, thus taking an unnecessary amount of extra cputime. It thus is recommended to remove the loosely and unconnected parts from the structure. This can be accomplished within IPL using the component labeling option invoked by the *cl_rank_extract* command. After labeling all voxels, a unique component label can be assigned to the voxels connected to the main structure (default value=127) and the unconnected parts can be removed.

In particular when using low-resolution images, it is possible that a substantial part of the structure (more than a few percent) is removed. This implies that the volume fraction of the remaining structure is reduced. In these situations, it can be advantageous to iteratively reduce the threshold value used for the segmentation step until the volume fraction of the final model equals the reference volume fraction⁷.

A.2.4 Component labeling

The present code allows the use of up to 9 different materials. To identify the different materials, a different component label should be given to voxels of different materials. As an example, voxels representing cortical bone can be labeled as '1', voxels representing trabecular bone as '2' and those representing cartilage as '3'. In order to use different materials, it will generally be necessary to create a different AIM object for each material (e.g. one for cortical bone, one for trabecular bone and one for cartilage). The component value of the non-zero voxel in each of the objects can be set using the IPL *set_value* command. The objects can be combined into one new object with different component labels for the different materials using the IPL *concat* command. This combined object then can be used as a basis for the FE-model

Section B: Preprocessing and solving

B.1 Starting the program

The program can run interactively as well as in batch/background mode. In the following, it is explained how to start the program for the different modes and platforms that are supported.

B.1.1 Interactive

(VMS:) Assuming that the logical and symbol described in the installation section have been set, the IPL program that contains the FE-solver can be started by typing *iplfe* in a terminal window. Before invoking the FE-solver, a segmented AIM object of character type (1-byte per voxel) should be loaded in memory e.g. by using the *IPL read* command. This AIM object represents the geometry of the FE-model. The value of each voxel is interpreted as a component number in the model where a non-zero component value represents material, and a zero-value an empty hole. Using the procedures described in the previous section it is possible to create AIM files with more than one component to represent more than one material (e.g. bone, marrow, cartilage). Due to the 1-byte type of the object, component values can range only from -128 to +127. Each component value can be converted to a specific material. For models with 3 or less different materials, the command *fe_solve3* can be used to start the FE-solver. For models with more than 3 materials, use the command *fe_solve255*. Both commands differ only in the amount of input parameters; when using *fe_solve3* only 3 material parameters need to be defined. When using *fe_solve255* all 255 material parameters need to be defined, which implies a lot of extra input in case only 1 or a few materials exist in the model. Typing *fe_solve* will issue the *fe_solve3* command.

(UNIX:) Assuming that the environmental variable and the pathname described in the installation section have been set, the FE-solver can be started by typing *fe_solve* in a terminal window. Before invoking the FE-solver, however, a segmented AIM file of version 3.0 (use the *IPL write_new* command) and character type (1-byte per voxel) must be copied from the VMS system to the UNIX system. The AIM file must be uncompressed (set compression to *none* when writing the file in IPL). This can be done using, for example, the *ftp* command (files must be transferred in binary mode). This AIM file represents the geometry of the FE-model. The value of each voxel is interpreted as a component number in the model where a non-zero component value represents material, and a zero-value an empty hole. Using the procedures described in the previous section it is possible to create AIM files with more than one component to represent more than one material (e.g. bone, marrow, cartilage). Due to the 1-byte type of the object, component values can range only from -128 to +127. Each component value can be converted to a specific material such that a maximum of 255 different materials is possible.

(SGI_UNIX:) In addition to the general UNIX section: the number of processors to be used during solving can be defined by setting the *OMP_NUM_THREADS* environment variable. For example, to use 8 processors type *setenv OMP_NUM_THREADS 8*.

B.1.2 Batch/background mode

(VMS:) Since the FE-analysis can take a lot of time, it is generally better to schedule these for batch processing. On the VMS system, the following commands are available for batch processing:

- To submit a FE-analysis to the queue type: *FEM aim_file_name*

where *aim_file_name* should be the name of an appropriate (character type) aim file. With this command, the FE-analysis will be submitted to the queue *SYS\$FE* with input parameters as defined in file *DISK1:[MICROCT.MAIN]IPL_FE_VM.COM*. In general, these parameters will need to be edited.

- To edit the parameters used for the FE-analysis typing: *EDIT UM:IPL_FE_VM.COM*.
This command file holds commands for pre-processing, the FE-analysis and for post-processing. Depending on the goal of the analysis this file may need editing. An overview of this command file and the parameters defined in this file is given in appendix B
- To show the status and jobnumber of a batch job type: *SHOW QUE SYS\$FE*
- To delete a job type: *DELETE/ENTRY=jobnumber*

where *jobnummer* must be replaced by its actual value

- To show interim job output type: TYPE SYS\$SCRATCH:AIM_FILE_NAME_FE.LOG
This file contains the information that is written to the screen during interactive processing. Note, however, that this file is updated only when the write buffer is flushed which is only when the buffer holds a few lines.

Note that when using the standard IPL_FE_VM.COM command file, extra files may be generated (see appendix B).

(UNIX:) When running in the background or batch mode, the input parameters must be read from a file. This file should list the input parameters in the same order as they are read when running the program interactively. There are two possible formats for this input file. In the first format, it simply lists the values of each parameter on a separate line, as they would be entered interactively. To improve readability and avoid errors, however, it is recommended to use the second format where each line starts with the name of the parameter, followed by a '>' symbol (without the quotes) which again is followed by the value of the parameter. If no value is entered after the '>' symbol, the default value is chosen. With this second format, the program will check if the parameters are specified in the right order. A sample input file of this format is given in Appendix C.

Assuming that the input parameters are stored in the file *default_file_name.stack*, the program can be started to run in the background by typing:

```
fe_solve < default_file_name.stack > default_file_name.out &
```

The output that will appear on the screen when running the program interactively thus is redirected to the file *default_file_name.out*.

To run the program in a batch queue, it will generally be necessary to write a small script file containing the line as written above (but without the &) that can be submitted to a batch queue using e.g. the *qsub* command.

B.2 Input parameters

The following input parameters are requested by the program when using the *fe_solve255* command:

Table B.1 input parameters

parameter name	type	default value	relevant range	short description
(VMS:) <i>in</i>	string	<i>in</i>		name of AIM object
(UNIX:) <i>aim_file_name</i>		<i>default_file_name</i>		name of the AIM file. Must be version 3.0 AIM file
<i>fea_file_name</i>	string	<i>default_file_name</i>		filename (without extension) for in/output files
<i>problem_nr</i>	integer	9	1 – 999	problem number: 1 to 99, >99 for custom
<i>scale_factor</i>	float	1.0		scaling factor for prescribed values
<i>output_option</i>	integer	1	1 – 4	output option flag, 1=min, 4=max. amount of output
<i>tolerance_force</i>	float	1E-4	> 0	force tolerance criterion
<i>tolerance_displ</i>	float	1E-4	> 0	displacement tolerance criterion
<i>max_nr_iter</i>	integer	10,000	> 1	max number of iterations
<i>restart_option</i>	integer	1	0, 1	1=use restart file
(UNIX:) <i>nr_mat</i>	integer	1	1 – 255	number of different materials
<i>comp_val_mat_1</i>	integer	127	-128 – 127	component value of mat. 1
<i>Ymodulus_mat_1</i>	float	10,000	> 0	Young's modulus of mat. 1
<i>Poissonr_mat_1</i>	float	0.3	-1 – 0.49	Poisson's ratio of mat. 1
<i>comp_val_mat_2</i>	integer	0	-128 – 127	component value of mat. 2
<i>Ymodulus_mat_2</i>	float	10,000	> 0	Young's modulus of mat. 2
<i>Poissonr_mat_2</i>	float	0.3	-1 – 0.49	Poisson's ratio of mat. 2
.				
.				
.				
<i>comp_val_mat_255</i>	integer	0	-128 – 127	component value of mat. 255
<i>Ymodulus_mat_255</i>	float	10,000	> 0	Young's modulus of mat. 255
<i>Poissonr_mat_255</i>	float	0.3	-1 – 0.49	Poisson's ratio of mat. 255

When using the *fe_solve3* command, the same input is required, but the number of materials is limited to 3.

A detailed description of each parameter is provided in the following table:

Table B.2: Description of input parameters

Parameter	Description
(VMS:) <i>in</i>	The name of the object loaded in memory. The object should be segmented data of character type (1-byte per voxel). The voxel value is interpreted as a component label.
(UNIX:) <i>aim_file_name</i>	The name of the AIM file. The AIM file should represent a segmented AIM file of character type (1-byte per voxel) and must be of version 3.0. The voxel value is interpreted as a component label.
<i>fea_file_name</i>	The root of associated file names. During the solving stage files can be needed for input and several output files are generated. These files will have a name <i>fea_file_name</i> .extension where 'extension' can be either 'bcd', 'listing', 'restart' or 'post'. For a description of these files: see sections B.3 and B.4.

Parameter	Description
<i>problem_nr</i>	A number defining the boundary conditions for the test that is simulated. Numbers in the range 1-99 represent standard problems that are stored in the 'standard.bcd' library. A listing of valid numbers and their meaning is given in Table B.3 below. Numbers ≥ 100 are interpreted as user-defined problems. In this case, a file with the name <i>fea_file_name.bcd</i> must be present in the default directory that contains a description of the boundary condition definitions for this problem. The format of this file is described in section 3.1.
<i>scale_factor</i>	A scaling factor used to redefine prescribed values (displacements, strains, forces, stresses) in standard or user-defined problems. In the standard problems, the prescribed value (stress or strain) is always set to one. With the default value for the <i>scale_factor</i> of 1.0 thus a unit stress or strain is prescribed with standard problems. Setting the <i>scale_factor</i> to -1.0 will transform tensile tests to compression tests. Note that all prescribed values are scaled e.g. if a user defined problem contains prescribed displacements, stresses and forces, all of these will be scaled. For certain standard problems (numbers 9 and 19) a unit prescribed strain or stress value is essential and the <i>scale_factor</i> should be set to its default value of 1.0. If this is not the case, the <i>scale_factor</i> will be reset to a unit value by the program.
<i>output_option</i>	Controls the amount of output that is written to the <i>fea_file_name.listing</i> file. The minimum amount of output is generated when <i>output_option</i> =0 and the maximum amount if <i>output_option</i> =4. The latter value should be used only for small test problems, since it will result in a very large listing file (see section B.4.1).
<i>tolerance_force</i>	Force convergence criterion for the iterative solver to indicate when the results are considered sufficiently accurate. During solving the ratio of the norm of the residual force vector F_{res} over the norm of the reaction force vector F_{reac} is calculated. If this ratio is small, the residual forces are much smaller than the reaction forces. This ratio thus determines the accuracy of the solution. A value of E-4 will ensure approximately four relevant digits for the calculated results, a value of E-6 approximate 6 significant digits etc. A converged solution requires that the condition $F_{res} / F_{reac} < tolerance_force$ is true. Note that, in the case that only displacements are prescribed, the reaction forces are not known on beforehand. In this situation, the reaction forces are calculated as well during the solution process. Since the calculation of reaction forces requires an extra matrix-vector multiplication, this is not done every iteration but only at iteration number 1, 2, 4, 8, 16, 32, 64, 100, 200, 300,....
<i>tolerance_displ</i>	Displacement convergence criterion for the iterative solver. During the solution process, the ratio of the change in the solution (displacement) vector at increment i: $x_i - x_{i-1}$ over the norm of the solution vector x_i is calculated. If this ratio is small, the changes in the calculated solution vector are small relative to the vector itself. A converged solution requires that the condition $x_i - x_{i-1} / x_i < tolerance_displ$ is met. Using this criterion alone (i.e. in combination with a large value for the force convergence criterion) is not recommended since it provides no guarantee with regard to the accuracy of the results.
<i>max_nr_iter</i>	The maximum number of iterations allowed for the iterative solver. The iterative solver will be halted, displaying an error message, if this number of iterations is reached. If the restart option is selected (<i>restart_option</i> =1, see below), the solution process can be continued from the last increment that was written to the restart file. If no restart option is selected (<i>restart_option</i> ≠1) the run can not be continued and must be started from scratch with an increased value for <i>max_nr_iter</i>

<i>Parameter</i>	Description
<i>restart_option</i>	The restart option is selected if <i>restart_option</i> is set to 1 and deselected otherwise. If selected, a file with the name <i>fea_file_name-i.restart</i>, with <i>i</i> the loadcase number, is written to disk every 100 iterations. If the calculation is discontinued before reaching a solution (e.g. due to a computer crash, because a cputime limit or the <i>max_nr_iter</i> limit was reached) it can be continued later from the last increment that is written to the restart file. The restart file can become very large (see section B.4.2) and care should be taken that sufficient disk place is available.
<i>comp_val_mat_1</i>	Component value of voxels that are converted to material 1. The default value for the first material is set to 127 (the default value for non-zero voxels in segmented AIM files).
<i>Ymodulus_mat_1</i>	Isotropic Young's modulus of material 1. Note that, since all sizes are specified in [mm] and all forces in [N], the Young's moduli must be specified in [MPa].
<i>Poissonr_mat_1</i>	Isotropic Poisson's ratio of material 1 [-].
<i>comp_val_mat_2</i>	Component value of voxels that are converted to material 2. The default value for the second and other material numbers is zero. Voxels with a value of zero, however, cannot be converted to elements. Thus, specifying a zero value implies that this material number is not present in the model. If a second material within the holes of the first material is to be modeled (e.g. marrow within the trabecular structure) the IPL <i>setvalue</i> option should be used to re-label the zero voxels to a value different from zero and different from the value used for material 1. Note that the specification of a component value that was assigned to another material earlier, the previous assignment will be overwritten.
<i>Ymodulus_mat_2</i>	Young's modulus of material 2 [MPa].
<i>Poissonr_mat_2</i>	Poisson's ratio of material 2 [-].
.	.
.	.
.	.
<i>comp_val_mat_255</i>	Component value of voxels that are converted to material 255.
<i>Ymodulus_mat_255</i>	Young's modulus of material 255 [MPa].
<i>Poissonr_mat_255</i>	Poisson's ratio of material 255 [-].

B.2.1 Pre-defined standard problems

The standard problems represent a set of boundary conditions for a number of standard tests as listed in the following table. A problem can consist of one or more loadcases that are solved sequentially. A detailed description of the boundary condition definitions of each standard problem is given in Appendix D.

Table B.3 Predefined standard problems

<i>problem_nr</i>	<i>nr_loadcases</i>	description
1	1	Uniaxial strain in the x-direction Displacements of nodes located at the $x=x_{min}$ and $x=x_{max}$ faces of the VOI are prescribed in the x-direction at a value of $u_x=x$ [mm] with x the x-coordinate. Displacements of nodes at the other faces are suppressed in the direction normal to the face.
2	1	Uniaxial strain in the y-direction Displacements of nodes located at the $y=y_{min}$ and $y=y_{max}$ faces of the VOI are prescribed in the y-direction at a value of $u_y=y$ [mm] with y the y-coordinate. Displacements of nodes at the other faces are suppressed in the direction normal to the face.
3	1	Uniaxial strain in the z-direction Displacements of nodes located at the $z=z_{min}$ and $z=z_{max}$ faces of the VOI are prescribed in the z-direction at a value of $u_z=z$ [mm] with z the z-coordinate. Displacements of nodes at the other faces are suppressed in the direction normal to the face.
4	1	Uniaxial shear strain in the yz-direction Displacements of nodes located at the $y=y_{min}$, $y=y_{max}$, $z=z_{min}$ and $z=z_{max}$ faces are prescribed in the y-direction at a value of $u_y=0.5z$ [mm], with z the z-coordinate, and in the z-direction at a value of $u_z=0.5y$ [mm], with y the y-coordinate. Displacements of nodes at the other faces are suppressed in the direction normal to the face.
5	1	Uniaxial shear strain in the xz-direction Displacements of nodes located at the $x=x_{min}$, $x=x_{max}$, $z=z_{min}$ and $z=z_{max}$ faces are prescribed in the x-direction at a value of $u_x=0.5z$ [mm], with z the z-coordinate, and in the z-direction at a value of $u_z=0.5x$ [mm], with x the x-coordinate. Displacements of nodes at the other faces are suppressed in the direction normal to the face.
6	1	Uniaxial shear strain in the xy-direction Displacements of nodes located at the $x=x_{min}$, $x=x_{max}$, $y=y_{min}$ and $y=y_{max}$ faces are prescribed in the x-direction at a value of $u_x=0.5y$ [mm], with y the y-coordinate, and in the y-direction at a value of $u_y=0.5x$ [mm], with x the x-coordinate. Displacements of nodes at the other faces are suppressed in the direction normal to the face.
9	6	Combination of standard problem 1 to 6 for the calculation of overall properties of cubic specimens with displacement boundary conditions
11	1	Uniaxial stress in the x-direction A distributed load is prescribed in the x-direction with a resultant force $F_x=\sigma^a l_y l_z$ at the face $x=x_{max}$, and with a resultant force $F'_x=-\sigma^a l_y l_z$ at the face $x=x_{min}$, with $\sigma^a=1$ [MPa] the apparent-level stress and l_y [mm] and l_z [mm] the size of the VOI in the y- and z-direction respectively. To suppress rigid body motions, displacements in the x-direction are suppressed for nodes located at the z-axis, displacements in the y-direction are suppressed for nodes located on the x-axis and displacements in the z-direction are suppressed for nodes located on the y-axis.

<i>problem_nr</i>	<i>nr_loadcases</i>	<i>description</i>
12	1	Uniaxial stress in the y-direction A distributed load is prescribed in the y-direction with a resultant force $F_y = \sigma^a l_x l_z$ at the face $y=y_{\max}$, and with a resultant force $F'_y = -\sigma^a l_x l_z$ at the face $y=y_{\min}$, with $\sigma^a=1$ [MPa] the apparent-level stress and l_x [mm] and l_z [mm] the size of the VOI in the x- and z-direction respectively. To suppress rigid body motions, displacements in the x-direction are suppressed for nodes located at the z-axis, displacements in the y-direction are suppressed for nodes located on the x-axis and displacements in the z-direction are suppressed for nodes located on the y-axis.
13	1	Uniaxial stress in the z-direction A distributed load is prescribed in the z-direction with a resultant force $F_z = \sigma_{\text{unit}} l_x l_y$ at the face $z=z_{\max}$, and with a resultant force $F'_z = -\sigma^a l_x l_y$ at the face $z=z_{\min}$, with $\sigma^a=1$ [MPa] the apparent level stress and l_x [mm] and l_y [mm] the size of the VOI in the x- and y-direction respectively. To suppress rigid body motions, displacements in the x-direction are suppressed for nodes located at the z-axis, displacements in the y-direction are suppressed for nodes located on the x-axis and displacements in the z-direction are suppressed for nodes located on the y-axis.
14	1	Uniaxial shear stress in the yz-direction A distributed load is prescribed in the y-direction with a resultant force $F_y = \sigma^a l_x l_y$ at the face $z=z_{\max}$, and with a resultant force $F'_y = -\sigma^a l_x l_y$ at the face $z=z_{\min}$, with $\sigma^a=1$ [MPa] the apparent-level stress and l_x [mm] and l_y [mm] the size of the VOI in the x- and y-direction respectively. A distributed load in the z-direction with a resultant force $F_z = \sigma^a l_x l_z$ is prescribed at the face $y=y_{\max}$, and with a resultant force $F'_z = -\sigma^a l_x l_z$ at the $y=y_{\min}$ face, with l_z [mm] the size of the VOI in the z-direction To suppress rigid body motions, displacements in the x-direction are suppressed for nodes at the z-axis, displacements in the y-direction are suppressed for nodes on the x-axis and displacements in the z-direction are suppressed for nodes on the y-axis
15	1	Uniaxial shear stress in the xz-direction A distributed load is prescribed in the z-direction with a resultant force $F_z = \sigma^a l_y l_z$ at the face $x=x_{\max}$, and with a resultant force $F'_z = -\sigma^a l_y l_z$ at the face $x=x_{\min}$, with $\sigma^a=1$ [MPa] the apparent-level stress and l_y [mm] and l_z [mm] the size of the VOI in the y- and z-direction respectively. A distributed load in the x-direction with a resultant force $F_x = \sigma^a l_x l_y$ is prescribed at the face $z=z_{\max}$, and with a resultant force $F'_x = -\sigma^a l_x l_y$ at the $z=z_{\min}$ face, with l_x [mm] the size of the VOI in the x-direction To suppress rigid body motions, displacements in the x-direction are suppressed for nodes at the y-axis, displacements in the y-direction are suppressed for nodes on the z-axis and displacements in the z-direction are suppressed for nodes on the x-axis

<i>problem_nr</i>	<i>nr_loadcases</i>	<i>description</i>
16	1	Uniaxial shear stress in the xy-direction A distributed load is prescribed in the y-direction with a resultant force $F_y = \sigma^a l_y l_z$ at the face $x = x_{\max}$, and with a resultant force $F'_y = -\sigma^a l_y l_z$ at the face $x = x_{\min}$, with $\sigma^a = 1$ [MPa] the apparent-level stress and l_y [mm] and l_z [mm] the size of the VOI in the y- and z-direction respectively. A distributed load in the x-direction with a resultant force $F_x = \sigma^a l_x l_z$ is prescribed at the face $y = y_{\max}$, and with a resultant force $F'_x = -\sigma^a l_x l_z$ at the $y = y_{\min}$ face, with l_x [mm] the size of the VOI in the z-direction To suppress rigid body motions, displacements in the x-direction are suppressed for nodes at the y-axis, displacements in the y-direction are suppressed for nodes on the z-axis and displacements in the z-direction are suppressed for nodes on the x-axis
19	6	Combination of standard problem 11 to 16 for the calculation of the complete stiffness matrix of cubic specimens with force boundary conditions. This is only possible if all voxels are converted to elements!
21	1	Uniaxial stress frictionless compression test in the x-direction Displacements of nodes located at the face $x = x_{\min}$ are suppressed in the x-direction, those located at the face $x = x_{\max}$ are prescribed in the x-direction at a value of $u_x = l_x$ [mm] with l_x the length of the specimen in x-direction. To suppress rigid body motions, displacements in the y-direction are suppressed for nodes located at the z-axis and displacements in the z-direction are suppressed for nodes located at the y-axis.
22	1	Uniaxial stress frictionless compression test in the y-direction Displacements of nodes located at the face $y = y_{\min}$ are suppressed in the y-direction, those located at the face $y = y_{\max}$ are prescribed in the y-direction at a value of $u_y = l_y$ [mm] with l_y the length of the specimen in y-direction. To suppress rigid body motions, displacements in the x-direction are suppressed for nodes located at the z-axis and displacements in the z-direction are suppressed for nodes located at the x-axis.
23	1	Uniaxial stress frictionless compression test in the z-direction Displacements of nodes located at the face $z = z_{\min}$ are suppressed in the z-direction, those located at the face $z = z_{\max}$ are prescribed in the z-direction at a value of $u_z = l_z$ [mm] with l_z the length of the specimen in z-direction. To suppress rigid body motions, displacements in the x-direction are suppressed for nodes located at the y-axis and displacements in the y-direction are suppressed for nodes located at the x-axis.
27	3	Combination of standard problem 21, 22 and 23 for the calculation of elastic properties of cubic specimens in the three spatial directions as in low-friction compression tests.
31	1	'High-friction' compression test in the x-direction Displacements of nodes located at the face $x = x_{\min}$ are suppressed in all directions, those at the face $x = x_{\max}$ are prescribed in the x-direction at a value of $u_x = l_x$ [mm] with l_x the length of the specimen in x-direction, and suppressed in the other directions.

<i>problem_nr</i>	<i>nr_loadcases</i>	<i>description</i>
32	1	'High-friction' compression test in the y-direction Displacements of nodes located at the face $y=y_{\min}$ are suppressed in all directions, those at the face $y=y_{\max}$ are prescribed in the y-direction at a value of $u_y=l_y$ [mm] with l_y the length of the specimen in y-direction, and suppressed in the other directions.
33	1	'High-friction' compression test in the z-direction Displacements of nodes located at the face $z=z_{\min}$ are suppressed in all directions, those at the face $z=z_{\max}$ are prescribed in the z-direction at a value of $u_z=l_z$ [mm] with l_z the length of the specimen in z-direction, and suppressed in the other directions.
37	3	Combination of standard problem 31, 32 and 33 for the calculation of anisotropic elastic properties of cubic specimens as in 'high-friction' compression tests

Notes:

- All standard problems are defined for a unit stress or strain. Other values can be obtained by setting the *scale_factor* input parameter. As an example, a low-friction compression test in the x-direction at 1% strain can be simulated by choosing *problem_nr* 21 and setting the *scale_factor* parameter to -0.01.
- The standard problems assume a cubic volume of interest (VOI). Only for cubic geometries will it be possible to calculate meaningful mechanical properties in all three spatial directions. Other geometries with two parallel sides, e.g. cylindrical specimens, can be used as well to calculate mechanical properties in the longitudinal directions (see section A.1.1).
- Care must be taken that the surfaces to which the load is applied are 'smooth'. If, for example, only one element is at the boundary, the loads are applied to this element alone, resulting in a very distorted load transfer. In cases with rough surfaces, the VOI must be cropped to obtain smooth surfaces.
- In order to suppress rigid body motion for problem number 11 to 16, and 21 to 23, displacements at the x, y and z -axes are prescribed in 1 direction. In those cases, however, where no nodal points are located at the x, y or z-axis, it is possible that rigid body motions are not fully suppressed and the program will halt. It is possible, however, that the program will not detect that rigid body motions are not fully suppressed. This will result in extremely slow convergence, no convergence or divergence of the iterative process. In these situations, it will be necessary to create a user-defined problem with prescribed displacements that ensure suppression of rigid body motion (see section B.3.1 to create user-defined boundary conditions).

B.2.2 Calculation of overall properties of cubic volumes of interest (VOI's)

Two standard problems are available for the calculation of the complete anisotropic elastic stiffness/compliance matrix of cubic VOI's: standard problem number 9 and number 19. With standard problem number 9 the overall properties are based on prescribed displacement boundary conditions. In general, the results thus obtained will provide an upper-bound on the stiffness of the specimen⁹. With the second standard problem (nr. 19) the overall properties are based on prescribed stress boundary conditions. The results thus obtained will provide a lower-bound on the stiffness of the specimen. Problem number 19, however, is possible only if all voxels are converted to elements, i.e. there can be no holes in the structure. In the case of porous structures the voxels representing the 'holes' must be assigned a different material number than voxels forming the structure. The elements representing the holes then can be assigned a very low Young's modulus and Poisson's ratio.

B.2.3 Boundary conditions and cputime

The number of iterations required by the iterative solver is largely dependent on the structure and on the number of prescribed displacements. It is not possible to predict the number of iterations required on beforehand. However, problems with many of their degrees of freedoms (i.e. displacements) prescribed will require (much) less iterations than problems with only a few of their degrees of freedom prescribed. This implies that problem number 1 to 9 can be solved (much) faster than problem number 11 to 19. Similarly, problem number 31 to 37 can be solved much faster than problem number 21 to 27.

B.3 Input files

B.3.1 The *fea_file_name.bcd* file

This file is needed only if *problem_nr* ≥ 100 , indicating a user-defined problem. For standard problems with a number less than 100, a set of boundary condition definitions is read from this file to represent the standard problems as defined in section B.2.1.

A problem can consist of one or more loadcases that are solved sequentially. The advantage of creating problems consisting of more than one loadcase is that all results will be written to the same post file. This ensures that results are kept together, whereas the post file will also be smaller than the post files for the same number of single-loadcase problems since the geometry information is written only once.

As an example, it can be convenient to have the results of compression tests in the 3 orthogonal directions written to the same post file (as is the case for standard problems 27 and 37). Another example is the case where overall properties are calculated based on the results of 6 loadcases (as for standard problem 9 and 19). In the latter case, a special parameter (*ihom*) must be flagged to indicate that this is a special 6-loadcases problem meant for the calculation of apparent properties.

Boundary conditions for each loadcase are specified as a set of boundary-condition-definitions (bcd's). Each boundary-condition-definition consists of three parts: a code parameter, a range description and a set of coefficients.

B. 3.1.1 The code parameter

The type of the boundary condition is defined by a code consisting of 5 digits: *ijklm* with the following meaning:

Table B.4 Code parameter description

digit	<range>	description
<i>i</i>	<1 – 5>	Prescribed quantity: 1: nodal displacement 2: nodal force 3: distributed load: prescribed element level stress 4: distributed load: prescribed apparent level stress 5: distributed load: prescribed total force
<i>j</i>	<1 – 3>	Prescribed degree of freedom: 1: x-direction 2: y-direction 3: z-direction

digit	<range>	description
k	<1 – 6>	Element face to which the boundary condition applies (relevant only if $i=4, 5$ or 6 , neglected otherwise): 1: $x=x_{\min}$ 2: $x=x_{\max}$ 3: $y=y_{\min}$ 4: $y=y_{\max}$ 5: $z=z_{\min}$ 6: $z=z_{\max}$
l	<0 – 9>	Number of the material to which the boundary definition applies: 0: all material 1: material 1 2: material 2 . . 9: material 9 In the case of boundary definitions that are defined at nodal points ($i=1$ or 2), the condition is applied if the node is connected to a voxel of the specified material type.
m	<0, 1>	Replace or add the boundary definition to the selected nodes: 0: Replace 1: Add It is allowed to define boundary conditions for the same degree of freedom more than once. This option determines whether a previously defined boundary condition should be overwritten or added to previously defined boundary conditions

B.3.1.2 The range parameters

The range to which the boundary condition definition applies is defined by the parameters $xfst$, $xlst$, $yfst$, $ylst$, $zfst$, $zlst$. These parameters refer to the dimensionless voxelgrid nodal coordinates in an integer-type coordinate system with the origin at (0, 0, 0) (Fig. B.1).

Note that the coordinate system of a VOI may change after issuing the IPL *bounding_box_cut* command, resulting in erroneous boundary conditions! This command is used in the standard version of the file IPL_FE_VM.COM which is used for batch-processing jobs. When using user defined boundary conditions, the *bounding_box_cut* command needs to be omitted from this file. See appendix B for more details.

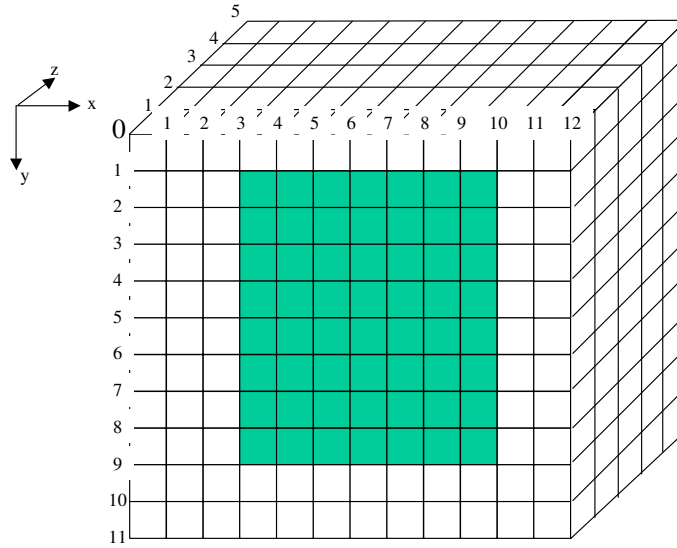


Fig. B.1 The grid-points coordinate system used to define the range for prescribed displacements or loads. The dark shaded area is defined by the coordinates (3,1,0) to (10, 9,0) and can be specified by setting the parameters $xfst=3$, $xlst=10$, $yfst=1$, $ylst=9$, $zfst=0$ and $zlst=0$. Nodes at the boundaries of the selected area are included.

B.3.1.3 The value coefficients

Six coefficients, a_x , b_x , a_y , b_y , a_z and b_z define the prescribed value for the boundary-condition-definition as a function of the coordinates in the following way:

$$f = s(a_x x^{b_x} + a_y y^{b_y} + a_z z^{b_z}) \quad (B.1)$$

with s the *scale_factor* [-] input parameter and x , y and z the spatial coordinates [mm] with their origin coinciding with the origin of the VOI. The interpretation of the coefficients a_x , a_y and a_z is dependent on the prescribed quantity as specified by the first digit i of the code as specified in Table B.4. Note that setting all b -coefficients to zero will result in a prescribed value that is independent of the position within the VOI.

Table B.5 value coefficients

i	interpretation of coefficients a_x , a_y and a_z
1	Nodal displacement Coefficients are interpreted as the coordinate-dependent nodal displacement [mm/mm ^b] that is applied to each nodal point in the specified range that is connected to an element of the selected material type. The power b represents the value chosen for b_x , b_y or b_z .
2	Nodal force: Coefficients are interpreted as the coordinate-dependent nodal force [N/mm ^b] that is applied to each node in the specified range that is connected to a voxel of the selected material type. The power b represents the value chosen for b_x , b_y or b_z .

<i>i</i>	interpretation of coefficients a_x , a_y and a_z
3	Distributed load: prescribed element-level stress Coefficients are interpreted as the coordinate-dependent (normal or shear) stress σ^e [MPa/mm^b] working on face number k of the elements in the specified range for material number l. The power b represents the value chosen for b_x, b_y or b_z. The total force equals: $F = n_l \sigma^e A_k^e$, with n_l the number of elements of material number l in the specified range and A_k^e [mm²] the surface of element face k.
4	Distributed load: prescribed apparent-level stress Coefficients are interpreted as the coordinate-dependent apparent (normal or shear) stress σ^a [MPa/mm^b] working on face number k of the elements in the specified range for selected material number l. The power b represents the value chosen for b_x, b_y or b_z. The total force applied to the elements in the selected range equals: $F = A \cdot \sigma^a$, with A [mm²] the surface spanned by the selected range and σ^a the prescribed stress. The equivalent element-level stress can be calculated from $\sigma^e = \left(\frac{A}{n_l A_k^e} \right) \sigma^a$, with n_l the number of elements of material number l in the specified range and A_k^e [mm²] the surface of element face k.
5	Distributed load: prescribed total force Coefficients are interpreted as the total force F [N/mm^b] of a distributed load working on the specified face number k of the elements in the specified range for selected material number l. The power b represents the value chosen for b_x, b_y or b_z. The equivalent element-level stress can be calculated from $\sigma^e = F / n_l A_k^e$, with n_l the number of elements of material number l in the specified range and A_k^e [mm²] the surface of element face k.

B.3.1.4 Format of the *fea_file_name.bcd* file

The *fea_file_name.bcd* file must have the following format:

Table B.6 Format of the *fea_file_name.bcd* file

problem number: (a15)			problem_nr (i5)		descriptive text (a80)							
nr_loads ases (i5)		ihom (i5)										
nbd (i10)												
code (i5)	xfst (i6)	xlst (i6)	yfst (i6)	ylst (i6)	zfst (i6)	zlst (i6)	ax (E12.4)	bx (E12.4)	ay (E12.4)	by (E12.4)	az (E12.4)	bz (E12.4)
.												
.												
.												
code (i5)	xfst (i6)	xlst (i6)	yfst (i6)	ylst (i6)	zfst (i6)	zlst (i6)	ax (E12.4)	bx (E12.4)	ay (E12.4)	by (E12.4)	az (E12.4)	bz (E12.4)
nbd (i5)												
code (i5)	xfst (i6)	xlst (i6)	yfst (i6)	ylst (i6)	zfst (i6)	zlst (i6)	ax (E12.4)	bx (E12.4)	ay (E12.4)	by (E12.4)	az (E12.4)	bz (E12.4)
.												
.												
.												
.												
code (i5)	xfst (i6)	xlst (i6)	yfst (i6)	ylst (i6)	zfst (i6)	zlst (i6)	ax (E12.4)	bx (E12.4)	ay (E12.4)	by (E12.4)	az (E12.4)	bz (E12.4)

with:

<i>problem number:</i>	a string that must appear literally as typed here
<i>problem_nr</i>	the problem number value
<i>descriptive text</i>	a description of this problem
<i>nr_loadcases</i>	the number of loadcases for this problem
<i>ihom</i>	set to 1 to indicate if this problem is used to calculate apparent properties of a VOI with assumed uniaxial apparent strain conditions set to 2 to indicate if this problem is used to calculate apparent properties of a VOI with assumed uniaxial apparent stress conditions set to 3 to indicate if this problem is used to calculate apparent properties of a VOI without assumption about the apparent stress/strain state otherwise: set to zero
<i>nbd</i>	the number of boundary condition definitions that follow
<i>code</i>	a boundary condition definition code
<i>xfst</i>	x gridcoordinate of the first node to which this definition applies
<i>xlst</i>	x gridcoordinate of the last node to which this definition applies
<i>yfst</i>	y gridcoordinate of the first node to which this definition applies
<i>ylst</i>	y gridcoordinate of the last node to which this definition applies
<i>zfst</i>	z gridcoordinate of the first node to which this definition applies
<i>zlst</i>	z gridcoordinate of the last node to which this definition applies

The format is defined by:

(a15)	a string of 15 characters
(i6)	an integer type number of length 6
(i5)	an integer type number of length 5
(E12.4)	a float type number in exponential notation of 12 digits in length with 4 digits right from the decimal point

B.3.1.5 Examples:

1) Prescribe a constant displacement of 1 [mm] in the x-direction for nodes defined in Fig. B.1 at the z=0 face for any material in the model:

code: 11000
range: *xfst*=3, *xlst*=10, *yfst*=1, *ylst*=9, *zfst*=0, *zlst*=0
coefficients: $a_x=1.0$, $b_x=a_y=b_y=a_z=b_z=0$

1) Prescribe a constant displacement corresponding to a strain of 1% in the x-direction for nodes defined in Fig. B.1 at the z=0 face for any material in the model:

code: 11000
range: *xfst*=3, *xlst*=10, *yfst*=1, *ylst*=9, *zfst*=0, *zlst*=10
coefficients: $a_x=0.01$, $b_x=1$, $a_y=b_y=a_z=b_z=0$

2) Prescribe a distributed load in the z-direction that is a linear function of the x-direction with a resultant force of 10 [N] at the z=z_{min} face of elements in the range defined in Fig. B.1 and for any material:

code: 53000
range: *xfst*=3, *xlst*=10, *yfst*=1, *ylst*=9, *zfst*=0, *zlst*=0
coefficients: $a_x=10.0$, $b_x=1.0$, $a_y=b_y=a_z=b_z=0$

3) Prescribe a distributed load with an element-level normal stress in the z-direction which is dependent on the x-coordinate according to: $f_z(x) = 1 - \frac{1}{9}(x-6)^2$ (see Fig. B.2 below) at the z=z_{min} face of elements in the range defined in Fig. B.1, but only for nodes connected to material number 2.

The equation $f_z(x) = 1 - \frac{1}{9}(x-6)^2$ cannot be cast directly in the form of eq. B.1. However, the function can

be re-written as: $f_z(x) = -3 + \frac{4}{3}x - \frac{1}{9}x^2$, which can be defined as the sum of the following three functions:

$f_1 = -3$; $f_2 = \frac{4}{3}x$; $f_3 = -\frac{1}{9}x^2$. The boundary condition then can be defined as a set of three boundary

condition definitions:

1st line:

code: 33520
range: $xfst=3, xlst=9, yfst=1, ylst=9, zfst=0, zlst=0$
coefficients: $a_x = -3, a_y = b_x = b_y = a_z = b_z = 0$

2nd line:

code: 33521
range: $xfst=3, xlst=9, yfst=1, ylst=9, zfst=0, zlst=0$
coefficients: $a_x = 4/3, b_x = 1, a_y = b_y = a_z = b_z = 0$

3rd line:

code= 33521
range: $xfst=3, xlst=9, yfst=1, ylst=9, zfst=0, zlst=0$
coefficients: $a_x = -1/9, b_x = 2, a_y = b_y = a_z = b_z = 0$

Note the use of the append option (digit m of the code set to 1) to add the functions.

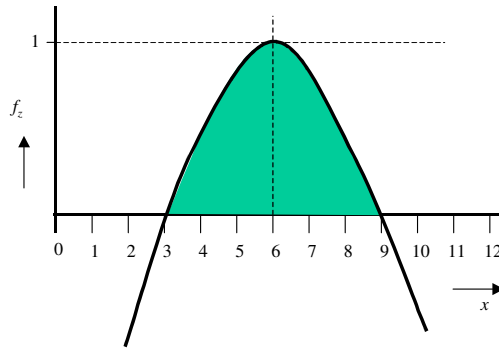


Fig. B.2 Sample stress distribution for example 3

B.4 Output files

B.4.1 The *fea_file_name.listing* file

This file contains information about the model and the analysis. Information is listed in a number of tables (see Appendix E for a sample listing file):

Table B.7 Description of the *fea_file_name.listing* file

Table	Contents
1	Parameters. This table echoes parameters passed from IPL as well as parameters read from the AIM file.
2	Conversion of voxels to elements The left-hand-side of this table provides an overview of the number of voxels with their component values found in the AIM file. The right-hand-side lists the material number that corresponds to each component value, the number of elements of this material and the volume fraction of this material relative to the VOI.

Table	Contents
3	FE-model Specifies the number of elements and nodes in the resulting FE-model. Note that the number of degrees of freedom is three times the number of nodes
4	Boundary condition definitions Lists the boundary condition definitions used for this loadcase. Specified are the code, <i>xfst</i> , <i>xlst</i> , <i>yfst</i> , <i>ylst</i> , <i>zfst</i> , <i>zlst</i> and coefficients a_x , b_x , a_y , b_y , a_z , b_z . A description of these parameters and coefficients is given in section B.3.1
5	Boundary conditions Lists the number of prescribed displacements and forces and the total applied force.
6	Iterative solving Lists parameters related to the iterative solving process: Iteration: iteration number Error_f: force convergence ratio that is compared to <i>tolerance_force</i> . Error_f is calculated as the ratio of the norm of the residual force vector $ F_{res} $ over the norm of the reaction force vector $ F_{reac} $ Fres : norm of the residual force vector Freac : norm of the reaction force vector Error_u: displacement convergence ratio that is compared to <i>tolerance_displ</i> . Error_u is calculated as the ratio of norm of the change in the solution (displacement) vector at increment i : $ x_i - x_{i-1} $ over the norm of the solution vector $ x_i $. delta x : norm of the change in the solution vector: $ delta x = x_{i-1} - x_i $ x norm of the solution vector
7	Nodal output Generated only if <i>output_option</i> > 3 Lists the calculated displacements and residual or reaction forces for each node in the model. The nodes are identified by their coordinates as indicated in Fig. B.1 Note that this option can produce a very large listing file and thus should be used only for small models.

If the number of loadcases is larger than 1, tables 4 through 7 are specified for each loadcase.

B.4.2 The *fea_file_name-i.restart* file

This file can be used to continue an analysis that was terminated due to, for example, a cputime limit or the *max_nr_iter* limit was reached, or because a higher accuracy of the results is required. It is a binary file that is written every 100 iterations. A separate restart file is written for each loadcase number i such that several restart files can exist if the number of loadcases is larger than 1. These files are generated only if *restart_option*=1. The size of this file is largely determined by 3 iterative vectors with a total size of $72 * nnode$ bytes, with *nnode* the number of nodes in the FE-model as listed in Table 3 of the *fea_file_name.listing* file. Thus, for a model with 1M nodes, the restart file will be approximately 72 MB in size.

If the run was successful, the restart file(s) can be deleted. This should be done manually.

B.4.3 The *fea_file_name.post* file

Contains parameters needed for the post-processing stage, the model geometry and two solution vectors (displacement and force vector). It is a binary file that is written at the end of each analysis. Total size of this file is approximately $nvox + 48 * nnode$ bytes, with *nvox* the total number of voxels (including voxels with a 0 value) and *nnode* the number of nodes in the FE-model as listed in Table 3 of the *fea_file_name.listing* file. Thus, for a model with 10M voxels and 1M nodes, the post file will be approximately 58 MB in size.

Section C: Post-processing

During this second stage, derived parameters are calculated and can be written to a new AIM file.

C1. Starting the program

The program is invoked by the *fe_post* command. The use of this command is equivalent to that of the *fe_solve* command described in section B1.

(SGI_UNIX:) Since the post-processing does not benefit much from parallelisation there is no need to request more than one processors.

C.1 Input parameters

The following parameters are requested by the program:

Table C.1 Input parameters

parameter name	type	default value	relevant range	short description
<i>post_file_name</i>	string	<i>default_file_name</i>		Name of postfile (without extension)
<i>output</i>	string	con		output AIM object
<i>variable_nr</i>	Integer	1	1 – 45	Number of the variable: 1 to 45
<i>loadcase_nr</i>	Integer	1	1 – 99	Loadcase number for contour plot
<i>output_option</i>	Integer	1	1 – 4	Output option
<i>interpol_option</i>	Integer	1	0 – 1	Interpolation option

A detailed description of each parameter is provided in the following table:

Table C.2: Description of input parameters

Parameter	Description
<i>post_file_name</i>	The root for the name of the in- and output files. The only input file needed is the 'postfile' with the name <i>post_file_name</i> .post, the only output file is a listing file which is given the name <i>post_file_name</i> .postlist. For a description of this file: see section C.3.1.
(VMS:) <i>output</i>	During the post-processing stage a new float-type AIM object is created if the selected <i>loadcase_nr</i> (see below) is available in the post file. The name of this object can be specified here. In this new AIM object, voxels are given the value of the calculated parameter if there is a corresponding element at that position, and are set to zero otherwise. This AIM object can be used to visualize the element-level loading patterns or to create histograms of the element-level parameter distribution.
<i>variable_nr</i>	Specifies the derived mechanical parameter to be calculated. Possible values and their meaning are specified in Table C.3 below.
<i>loadcase_nr</i>	The loadcase number for which the AIM object is calculated. Note that all loadcases available in the <i>post_file_name</i> .post file are processed and the results of all loadcases are written to the <i>post_file_name</i> .postlist file. However, the AIM object is generated for the selected <i>loadcase_nr</i> only. If AIM objects of more than one loadcase are required, the post-processing should be repeated with different values for <i>loadcase_nr</i> . Note that if <i>loadcase_nr</i> is set to 0, no AIM object is created.

Parameter	Description
<i>output_option</i>	Controls the amount of output that is written to the <i>post_file_name</i> .listing file. The minimum amount of output is generated when <i>output_option</i> =0 and the maximum amount if <i>output_option</i> =4. The latter value should be used only for small test problems, since it will result in a very large listing file (see section C.3.1).
<i>interpol_option</i>	If set to 0, the element values of the chosen mechanical parameter are written to the output object. If set to 1, the element values are first extrapolated to the element nodes, then averaged between neighboring elements, and finally interpolated from the nodes to the element center. Written to the output object are the interpolated values. Choosing this option will result in a somewhat smoother distribution of the selected parameter.
<i>averaging_option</i>	Controls how the average material properties are calculated when using problem number 9, 19 or a user-defined problem in which <i>ihom</i> is set to a non-zero value. If set to 0, the value defined in the bcd file is used. If set to 1, uniaxial apparent strain is assumed. If set to 2, uniaxial apparent stress is assumed. If set to 3, no assumptions are made. Note that values 2 and 3 are possible only for models without holes (i.e. for models in which all voxels were converted to elements). See section C.3.1 for further details.

1.1 Choice of mechanical parameters

Mechanical parameters are calculated at the center of each element of the FE-model. The following table lists the available parameters:

Table C.3 Mechanical parameters

Parameter number	Description
1	ϵ_{xx} : first component of strain
2	ϵ_{yy} : second component of strain
3	ϵ_{zz} : third component of strain
4	ϵ_{yz} : fourth component of strain
5	ϵ_{xz} : fifth component of strain
6	ϵ_{xy} : sixth component of strain
11	σ_{xx} : first component of stress
12	σ_{yy} : second component of stress
13	σ_{zz} : third component of stress
14	σ_{yz} : fourth component of stress
15	σ_{xz} : fifth component of stress
16	σ_{xy} : sixth component of stress

Parameter number	Description
21	ϵ_{p1} : signed largest component of principal strain when the principal components are sorted as: $\epsilon_{p1} > \epsilon_{p2} > \epsilon_{p3}$
22	ϵ_{p2} : signed intermediate component of principal strain when the principal components are sorted as: $\epsilon_{p1} > \epsilon_{p2} > \epsilon_{p3}$
23	ϵ_{p3} : signed smallest component of principal strain when the principal components are sorted as: $\epsilon_{p1} > \epsilon_{p2} > \epsilon_{p3}$
24	ϵ'_{p1} : signed largest component of principal strain in an absolute sense, i.e. when the principal components are sorted as: $ \epsilon_{p1} > \epsilon_{p2} > \epsilon_{p3} $
25	ϵ'_{p2} : signed intermediate component of principal strain in an absolute sense, i.e. when the principal components are sorted as: $ \epsilon_{p1} > \epsilon_{p2} > \epsilon_{p3} $
26	ϵ'_{p3} : signed smallest component of principal strain in an absolute sense, i.e. when the principal components are sorted as: $ \epsilon_{p1} > \epsilon_{p2} > \epsilon_{p3} $
31	σ_{p1} : signed largest component of principal stress when the principal components are sorted as: $\epsilon_{p1} > \epsilon_{p2} > \epsilon_{p3}$
32	σ_{p2} : signed intermediate component of principal stress when the principal components are sorted as: $\sigma_{p1} > \sigma_{p2} > \sigma_{p3}$
33	σ_{p3} : signed smallest component of principal stress when the principal components are sorted as: $\sigma_{p1} > \sigma_{p2} > \sigma_{p3}$
34	σ'_{p1} : signed largest component of principal stress in an absolute sense, i.e. when the principal components are sorted as: $ \sigma_{p1} > \sigma_{p2} > \sigma_{p3} $
35	σ'_{p2} : signed intermediate component of principal stress in an absolute sense, i.e. when the principal components are sorted as: $ \sigma_{p1} > \sigma_{p2} > \sigma_{p3} $
36	σ'_{p3} : signed smallest component of principal stress in an absolute sense, i.e. when the principal components are sorted as: $ \sigma_{p1} > \sigma_{p2} > \sigma_{p3} $
41	ϵ_{eq} : equivalent strain, calculated as: $\epsilon_{eq} = \sqrt{\frac{1}{2} \left((\epsilon_{xx} - \epsilon_{yy})^2 + (\epsilon_{yy} - \epsilon_{zz})^2 + (\epsilon_{zz} - \epsilon_{xx})^2 + 6(\epsilon_{yz}^2 + \epsilon_{xz}^2 + \epsilon_{xy}^2) \right)}$
42	σ_{eq} : Von Mises equivalent stress, calculated as: $\sigma_{eq} = \sqrt{\frac{1}{2} \left((\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2 + 6(\sigma_{yz}^2 + \sigma_{xz}^2 + \sigma_{xy}^2) \right)}$
43	ϵ_{energy} : energy equivalent strain, calculated as: $\epsilon_{energy} = \sqrt{\frac{2U}{E}}$, with U the strain energy density: $U = \frac{1}{2} (\epsilon_{xx} \sigma_{xx} + \epsilon_{yy} \sigma_{yy} + \epsilon_{zz} \sigma_{zz} + \epsilon_{yz} \sigma_{yz} + \epsilon_{xz} \sigma_{xz} + \epsilon_{xy} \sigma_{xy})$ and E the Young's modulus.

Parameter number	Description
44	σ_{energy}: energy equivalent stress, calculated as: $\sigma_{\text{energy}} = \sqrt{2UE}, \text{ with } U \text{ the strain energy density:}$ $U = \frac{1}{2} (\varepsilon_{xx} \sigma_{xx} + \varepsilon_{yy} \sigma_{yy} + \varepsilon_{zz} \sigma_{zz} + \varepsilon_{yz} \sigma_{yz} + \varepsilon_{xz} \sigma_{xz} + \varepsilon_{xy} \sigma_{xy})$ and E the Young's modulus.
45	Strain energy density U, defined as: $U = \frac{1}{2} (\varepsilon_{xx} \sigma_{xx} + \varepsilon_{yy} \sigma_{yy} + \varepsilon_{zz} \sigma_{zz} + \varepsilon_{yz} \sigma_{yz} + \varepsilon_{xz} \sigma_{xz} + \varepsilon_{xy} \sigma_{xy})$

C.2 Input files

C.2.1 The *post_file_name.post* file

The only input file needed for post-processing is the *post_file_name.post* file that is written at the end of the solving stage. This file contains all necessary parameters, the geometry of the model and the results vectors (displacement and force vector) that are calculated during the solving stage.

C.3 Output files

C.3.1 The *post_file_name.postlist* file

The only output file generated is the *post_file_name.postlist* file that lists the parameters read from the post file and the calculated results in several tables. An example of this file is given in Appendix F.

Table C.4 Description of the *post_file_name.postlist* file

Table	Contents
1	Parameters read from post file This table echoes the parameters read from the post file.
2	Conversion of voxels to elements The left-hand-side of this table provides an overview of the number of voxels with their component values found in the post file. The right-hand-side lists the material number that corresponds to each component value, the number of elements of this material and the volume fraction of this material.
3	FE-model Specifies the number of elements and nodes in the resulting FE-model. Note that the number of degrees of freedom is three times the number of nodes
4	Average tissue strains and stresses Lists the average values for the 6 strain- and stress-components for each of the materials in the model. Strains are calculated as: $\bar{\varepsilon}_{ij} = \frac{1}{N} \sum_{k=1}^N \varepsilon_{ij}^k$, with ε_{ij}^k strain component ij of element k and N the number of elements of the material and similar for the stresses.

Table	Contents
5	Apparent strains and stresses Lists the strains and stresses of the VOI based on the deformation of and reaction forces at its surfaces as in real experimental tests. Values are calculated as: $\varepsilon_{xx} = \frac{(\bar{u}_{x,x=x_{\max}} - \bar{u}_{x,x=x_{\min}})}{l_x}$ $\varepsilon_{yy} = \frac{(\bar{u}_{y,y=y_{\max}} - \bar{u}_{y,y=y_{\min}})}{l_y}$ $\varepsilon_{zz} = \frac{(\bar{u}_{z,z=z_{\max}} - \bar{u}_{z,z=z_{\min}})}{l_z}$ $\varepsilon_{yz} = \frac{(\bar{u}_{y,z=z_{\max}} - \bar{u}_{y,z=z_{\min}})}{l_z} + \frac{(\bar{u}_{z,y=y_{\max}} - \bar{u}_{z,y=y_{\min}})}{l_y}$ $\varepsilon_{xz} = \frac{(\bar{u}_{x,z=z_{\max}} - \bar{u}_{x,z=z_{\min}})}{l_z} + \frac{(\bar{u}_{z,x=x_{\max}} - \bar{u}_{z,x=x_{\min}})}{l_x}$ $\varepsilon_{xy} = \frac{(\bar{u}_{x,y=y_{\max}} - \bar{u}_{x,y=y_{\min}})}{l_y} + \frac{(\bar{u}_{y,x=x_{\max}} - \bar{u}_{y,x=x_{\min}})}{l_x}$ $\sigma_{xx} = \frac{(F_{x,x=x_{\max}} - F_{x,x=x_{\min}})}{2l_y l_z}$ $\sigma_{yy} = \frac{(F_{y,y=y_{\max}} - F_{y,y=y_{\min}})}{2l_x l_z}$ $\sigma_{zz} = \frac{(F_{z,z=z_{\max}} - F_{z,z=z_{\min}})}{2l_x l_y}$ $\sigma_{yz} = \frac{(F_{y,z=z_{\max}} - F_{y,z=z_{\min}})}{4l_x l_y} + \frac{(F_{z,y=y_{\max}} - F_{z,y=y_{\min}})}{4l_x l_z}$ $\sigma_{xz} = \frac{(F_{x,z=z_{\max}} - F_{x,z=z_{\min}})}{4l_x l_y} + \frac{(F_{z,x=x_{\max}} - F_{z,x=x_{\min}})}{4l_y l_z}$ $\sigma_{xy} = \frac{(F_{x,y=y_{\max}} - F_{x,y=y_{\min}})}{4l_x l_z} + \frac{(F_{y,x=x_{\max}} - F_{y,x=x_{\min}})}{4l_y l_z}$ with: $\bar{u}_{x,x=x_{\max}}$ the average displacement in the x-direction for nodes located at the face with the coordinate $x=x_{\max}$ and similar for the other directions/faces. $F_{x,x=x_{\max}}$ the sum of the nodal forces in the x-direction for nodes located at the face $x=x_{\max}$ and similar for the other directions/faces. l_x, l_y, l_z the dimensions of the VOI in x-, y- and z-direction respectively.
6	Nodal values Lists the average displacements and the total force in the three spatial directions for the nodes located at the 6 faces of the VOI.
7	Distribution selected parameter Lists the average value, standard deviation, minimum value, maximum value and other parameters describing the distribution of the selected parameter.

Table	Contents
8	Element output Generated only if <i>output_option</i> > 3 Lists the value of the selected parameter calculated at the center of each element in the model. The elements are identified by the coordinates of their first corner node. In the example of Fig. B.1 the first element would be given coordinates (3,2) etc. Elements are sorted by their material number. Note that this option can produce a very large listing file and thus should be used only for small models.
9	Apparent stiffness matrix in specimen coordinate system Generated only if <i>ihom</i> ≠ 0. The apparent stiffness matrix E and the apparent compliance matrix C are 6×6 matrices that relates the apparent stresses in column $\bar{\sigma}$ and the apparent strains in column $\bar{\epsilon}$ according to: $\bar{\sigma} = \mathbf{E}\bar{\epsilon} \quad \text{and} \quad \bar{\epsilon} = \mathbf{C}\bar{\sigma}$ The apparent stresses and strains can be calculated from¹⁰: $\bar{\sigma} = \sum_{k=1}^{nmat} \rho_k \bar{\sigma}_k \quad \text{and} \quad \bar{\epsilon} = \sum_{k=1}^{nmat} \rho_k \bar{\epsilon}_k,$ with $\bar{\sigma}_k = \frac{1}{V_k} \int_{V_k} \sigma_k dV$ and $\bar{\epsilon}_k = \frac{1}{V_k} \int_{V_k} \epsilon_k dV$ the average stress and strain, respectively, in material <i>k</i> and ρ_k the volume fraction of material <i>k</i>. The complete stiffness matrix E or compliance matrix C can be calculated if six independent measurements of the apparent stress and strain are available. For the following, it is assumed that 6 analyses are done and that the 6 independent apparent strains are stored as the 6 columns of a 6×6 matrix $\bar{\epsilon}_{\sim}$ whereas the corresponding 6 independent average stresses are stored as columns in a 6×6 matrix $\bar{\sigma}_{\sim}$. There are 3 different options to calculate the apparent properties based on these 6 analyses: Option 1: assumed uniaxial apparent strain In the case that <i>ihom</i>=1 or <i>averaging_option</i>=1 it is assumed that $\bar{\epsilon}_{\sim} = \mathbf{I}$, i.e. the strains represent 6 cases of uniaxial strain, such that: $\bar{\sigma}_{\sim} = \sum_{k=1}^{nmat} \rho_k \bar{\sigma}_k = \mathbf{E}\mathbf{I} = \mathbf{E}$ with $\bar{\sigma}_k = \frac{1}{V_k} \int_{V_k} \sigma_k dV$ holding the average stresses in material <i>k</i> for each of the 6 load cases stored as columns in a 6×6 matrix. E thus can be determined by calculating the average stresses in each component for each loading case. Since the average stress in the holes of a structure are zero, it suffices to calculate only the average stresses for the non-zero voxels. Note that this option is reasonable only if boundary conditions are chosen to represent 6 cases of uniaxial strain. Option 2: assumed uniaxial apparent stress In the case that <i>ihom</i>=2 or <i>averaging_option</i>=2 it is assumed that $\bar{\sigma}_{\sim} = \mathbf{I}$, with I the 6×6 unit matrix, such that:

Table	Contents
	$\bar{\underline{\underline{\epsilon}}} = \sum_{k=1}^{nmat} \rho_k \bar{\underline{\underline{\epsilon}}}_k = \underline{\underline{C}} \underline{\underline{I}} = \underline{\underline{C}}$ with $\bar{\underline{\underline{\epsilon}}}_k$ holding the average strains in material k for each of the 6 load cases stored as columns in a 6x6 matrix. $\underline{\underline{C}}$ thus can be determined by calculating the average strains in each component for each loading case. Note that this assumption is reasonable only if boundary conditions are chosen to represent 6 cases of uniaxial stress. Also note that, the average strain in holes is not zero, such that the use of this option requires the modeling of the holes. Option 3: no assumptions about apparent stresses or strain In case that $ihom=3$ or $averaging_option=3$, the apparent stiffness matrix is calculated from: $\bar{\underline{\underline{\sigma}}} \bar{\underline{\underline{\epsilon}}}^{-1} = \underline{\underline{E}}$ $\underline{\underline{E}}$ thus can be determined by calculating the average stresses and strains in each component for each loading case. Note that this calculation requires $\bar{\underline{\underline{\epsilon}}}$ being non-singular, i.e. the six loading conditions that are applied should result in six independent strain vectors. Also note that, the average strain in holes is not zero, such that the use of this option requires the modeling of the holes.
10	Apparent compliance matrix in specimen coordinate system Generated only if $ihom \neq 0$. The apparent compliance matrix $\underline{\underline{C}}$ which is calculated as: $\underline{\underline{C}} = \underline{\underline{E}}^{-1}$
11	Material parameters in specimen coordinate system The engineering elastic constants calculated from the component c_{ij} of the compliance matrix $\underline{\underline{C}}$ in the following way: $E_{xx} = 1/c_{11}$ $E_{yy} = 1/c_{22}$ $E_{zz} = 1/c_{33}$ $G_{yz} = 1/c_{44}$ $G_{xz} = 1/c_{55}$ $G_{xy} = 1/c_{66}$ $\nu_{yx} = -c_{12}E_{yy}$ $\nu_{zx} = -c_{13}E_{zz}$ $\nu_{xy} = -c_{21}E_{xx}$ $\nu_{zy} = -c_{23}E_{zz}$ $\nu_{xz} = -c_{31}E_{xx}$ $\nu_{yz} = -c_{32}E_{yy}$
12	Apparent stiffness matrix in best orthotropic coordinate system Using an optimization algorithm¹¹ a transformation matrix is sought that can transform the stiffness matrix as listed in Table 9 in the orthotropic form. This table lists the best orthotropic representation $\underline{\underline{E}}^{ort}$ that could be found.
13	Apparent compliance matrix in best orthotropic coordinate system The best possible orthotropic form of the compliance matrix $\underline{\underline{C}}^{ort}$ is calculated as: $\underline{\underline{C}}^{ort} = (\underline{\underline{E}}^{ort})^{-1}$
14	Material parameters in specimen best orthotropic system are calculated in the same way as listed in table 11 but now using the best orthotropic representation of the compliance matrix $\underline{\underline{C}}^{ort}$.

C.4 (VMS:) Output IPL objects

C.4.1 The output AIM object

Created is a float-type (4byte per voxel) AIM object in which each voxel is given the value of the parameter calculated at the corresponding element position or the interpolated value if *interpol_option*=1. This object can be saved to disk or can be further analyzed. It can be used, for example, to create a VOI that is smaller than the analyzed model and to calculate histograms for the selected VOI using the appropriate IPL commands (in preparation). Information about the chosen parameters is written to the header of the AIM object. Using the IPL *examine* command, the information in the AIM header can be displayed.

Section D: Tools

D.1 Examine post and restart files

In order to examine the parameters stored in the header of existing post and restart files, a tool “fe_exa” is available. The program is invoked by the *fe_exa* command. The only input parameter required is the full name of the post or restart file:

Table D.1 input parameters

parameter name	type	default value	relevant range	short description
<i>full_file_name</i>	string	default_file_name.post		complete name of post or restart file

After executing the command, parameters stored in the header of the post or restart file are written to the screen.

Section E: advanced options

E.1 the Scanco_fe.ini file

This file contains parameters that define the operating system, the format and directories in which files are written, and the default values of input parameters (the latter applies only to the UNIX version). It is recommended to leave the values in this file to the default value for the operating system. Parameters can be of three different types: integer, float or string. Values can be assigned by typing the name of the parameter followed by the assignment character “=” (without the quotes) and the value. String values must be placed between single quotes. An example of the scanco_fe.ini file for the VMS OS is shown in appendix G. The parameters defined in the scanco_fe.ini file are described in Table E.1

Table E.1 Parameters defined in the scanco_fe.ini file

parameter name	type	recommended value	relevant range	short description
<i>os_name</i>	string	(VMS:) VMS (SGI_UNIX:) SGI_UNIX	VMS SGI_UNIX	name of the operating system
<i>os_endian</i>	string	(VMS:) LITTLE_ENDIAN (SGI_UNIX:) BIG_ENDIAN	LITTLE_ENDIAN BIG_ENDIAN	endian type of operating system
<i>os_float</i>	string	(VMS:) VAX_FLOAT (SGI_UNIX:) IEEE_FLOAT	VAX_FLOAT IEEE_FLOAT	float type of operating system
<i>os_defrecl</i>	integer	512	32 – 4096	default record length for binary files [bytes]
<i>postfile_recl</i>	integer	512	32 – 4096	record length of post file [bytes]
<i>postfile_endian</i>	string	(VMS:) LITTLE_ENDIAN (SGI_UNIX:) BIG_ENDIAN	LITTLE_ENDIAN BIG_ENDIAN	endian type for postfile
<i>postfile_float</i>	string	(VMS:) IEEE_FLOAT (SGI_UNIX:) IEEE_FLOAT	VAX_FLOAT IEEE_FLOAT	float type of postfile

<i>postfile_version</i>	integer	10		version number of postfile
<i>restfile_recl</i>	integer	512	>32	record length of restart file [bytes]
<i>restfile_endian</i>	string	(VMS:) LITTLE_ENDIAN (SGI_UNIX:) BIG_ENDIAN	LITTLE_ENDIAN BIG_ENDIAN	endian type for restart file
<i>restfile_float</i>	string	(VMS:) VAX_FLOAT (SGI_UNIX:) IEEE_FLOAT	VAX_FLOAT IEEE_FLOAT	float type of restart file
<i>restfile_version</i>	integer	10		version of restart file
<i>spfname</i>	string	(VMS:) SCANCO_FE_DIR:standar.bcd (UNIX:) /users/scanco/fe_64/standard.bcd		path + name of the file that holds the standard boundary condition definitions
<i>def_file_name</i>	string	default_file_name		default for <i>fea_file_name</i>
<i>def_problem_nr</i>	integer	9	1 – 999	default value for <i>problem_number</i>
<i>def_scale_factor</i>	float	1.0		default value for <i>scale factor</i>
<i>def_output_option</i>	integer	1	1 – 4	default value for <i>output option</i>
<i>def_tolerance_force</i>	float	1E-4		default value for <i>tolerance_force</i>
<i>def_tolerance_displ</i>	float	1E-4		default value for <i>tolerance_displ</i>
<i>def_max_nr_iter</i>	integer	10000		default value for <i>max_nr_iter</i>
<i>def_restart_option</i>	integer	1	0, 1	default value for <i>restart_option</i>
<i>def_nr_mat</i>	integer	(VMS:) N/A (UNIX:) 1	1 – 255	default value for <i>nr_mat</i>
<i>def_comp_val_mat</i>	integer	0	-128 – 127	default value for <i>comp_val_mat</i>
<i>def_comp_val_mat1</i>	integer	127	-128 – 127	default value for <i>comp_val_mat1</i>
<i>def_Ymodulus_mat</i>	float	10000	> 0	default value for <i>Ymodulus_mat</i>
<i>def_Ymodulus_mat1</i>	float	10000	> 0	default value for <i>Ymodulus_mat1</i>
<i>def_Poissonr_mat</i>	float	0.3	-1 – 0.49	default value for <i>Poissonr_mat</i>
<i>def_Poissonr_mat1</i>	float	0.3	-1 – 0.49	default value for <i>Poissonr_mat1</i>
<i>def_variable_nr</i>	integer	42	1 – 45	default value for <i>variable_nr</i>
<i>def_loadcase_nr</i>	integer	1	1 – 99	default value for <i>loadcase_nr</i>

E.2 Exchange of binary files between platforms

The *fea_file_name.post* and *fea_file_name.restart* file are binary files that are written in a specific endian type (byte order) and float type (the representation of floats in memory). In general, binary files are platform specific and cannot be used on other platforms than the one they were written on. The VMS version of the FE-code, however, has conversion routines that enable reading and writing both endian types (little and big endian) and float types (VAX floats and IEEE floats). In this way it is possible to read binary files from other platforms or write files for other platforms. All binary files written by the FE-code have a header that specifies the endian and float type in which they were written, and the conversion of these files is done without any user intervention.

The UNIX version of the code, however, can only do the endian conversion. This implies that reading files from the VMS platform on UNIX platforms requires that these files be written in the UNIX native float format (IEEE float). Table E.2 lists which float and endian types can be read on the different platforms.

Table E.2 Read and write options for binary files on different platforms

convert	for action	VMS	SGI UNIX
floats	read	VAX_FLOAT	IEEE_FLOAT
		IEEE_FLOAT	
	write	VAX_FLOAT	IEEE_FLOAT
		IEEE_FLOAT	
endian	read	BIG_ENDIAN	BIG_ENDIAN
		LITTLE_ENDIAN	LITTLE_ENDIAN
	write	BIG_ENDIAN	BIG_ENDIAN
		LITTLE_ENDIAN	LITTLE_ENDIAN

Note that with the default values as defined in the `scanco_fe.ini` file, it is possible to exchange the `fea_file_name.post` file between UNIX and VMS platforms in both directions. Exchange of the `fea_file_name-i.restart` files, however is limited to one direction: from UNIX to VMS, since no conversion of VAX_FLOATS to IEEE_FLOATS is available in the UNIX version. If the exchange of restart files from VMS to UNIX is desired, the `restfile_float` parameter in the `scanco_fe.ini` file of the VMS system must be set to IEEE_FLOAT. Since this implies the conversion of floats in the restart file each time the (usually large) restart file is written, this will require some extra cputime.

References

- 1 Van Rietbergen, B., Weinans, H., Huiskes, R., Odgaard, A., A new method to determine trabecular bone elastic properties and loading using micromechanical finite-element models, J. Biomech., 28, 69, 1995
- 2 Van Rietbergen B., Weinans H., Polman, B.J.W. and Huiskes, R. (1996) Computational strategies for iterative solutions of large FEM applications employing voxel data, Int. J. for Num. Meth. in Eng., 39: 2743-2767
- 3 Harrigan, T. P., Jasty, M., Mann, R. W. and Harris, W. H., Limitations of the continuum assumption in cancellous bone. J. Biomech. 21, 269-275., 1988
- 4 Van Rietbergen, B., Müller, R., Ulrich, D., Rügsegger, P. and Huiskes, R. (1999) Tissue stresses and strain in trabeculae of a canine proximal femur can be quantified from computer reconstructions, J. Biomech., 32:443-451
- 5 Guldberg, R.E., Hollister, S.J., Charras, G.T. (1998) The accuracy of digital image-based finite element models, J. Biomech Eng., 120:289-95
- 6 Ladd AJ, Kinney JH. Numerical errors and uncertainties in finite-element modeling of trabecular bone. J. Biomech. 1998 Oct;31(10):941-5
- 7 Ulrich, D., van Rietbergen, B., Weinans, H., and Rügsegger, P., (1998), Finite Element Analysis of trabecular bone structure: a comparison of image-based meshing techniques, J. Biomech., 31, 1187-1192
- 8 Jacobs, C.R., Mandell, J.A. and Beaupre, G.S. (1993) A comparative study of automatic finite element mesh generation techniques in orthopaedic biomechanics, ASME/BED 24:512-514
- 9 Hill, R. Elastic properties of reinforced solids: some theoretical principles, J. mech. Phys. Solids, 11, 357-37, 1963
- 10 Hill, R. Elastic properties of reinforced solids: some theoretical principles, J Mech Phys Solids 1963, 11:357-372
- 11 Van Rietbergen B, Odgaard A, Kabel J, Huiskes R. Direct mechanics assessment of elastic symmetries and properties of trabecular bone architecture. J Biomech. 1996 Dec;29(12):1653-7

Appendix A. (VMS:) Logicals and symbols defined for the FE-program

The following logicals and symbols are defined in the file DISK1:[MICROCT]MICROCT_FE.COM:

Logicals:

```
DEFINE SCANCO_FE_DIR UCTD:  
DEFINE UCT_FE_QUEUE SY$FE
```

Symbols:

```
IPLFE := $UM:IPL_SCANCO_FE.EXE /INTER  
IPL_FE_BATCH := $UM:IPL_SCANCO_FE.EXE  
FEM := @UM:UCT_FE_ANALYSIS.COM
```

- 1) preprocessing
 - read the aim file
 - perform component labeling
 - perform bounding box cut
- 2) FE-analysis
- 3) post-processing

In its default form this file can be used for segmented aim files in which all non-zero voxels have the same value. If, however, the aim file holds more than 1 component, the `cl_low_rank_extract` command must be omitted (e.g. by placing exclamation marks on the first position of line 39-45) and the `cl_rank_extract` in combination with the `gobj` command must be used instead. This can be achieved by removing the exclamation marks from line 49-66.

The *bounding_box_cut* command (line 69-73) will remove any empty space around the object in the VOI such that it is ensured that boundary conditions are applied to the object. This can pose problems, however, when using user-defined boundary definitions, since, after the removal of empty space the coordinate system for the object may have changed (see Fig. A1). To avoid such problems, it will be necessary to omit the *bounding_box_cut* command when using user defined boundary condition definitions by placing exclamation marks at the first position of line 69-73.

36

```

45.
46. !!!! for multi-value object (IPL /concat by user)
47. !-----
48.
49. !/cl_rank_extract
50. ! -input          in
51. ! -output         cl
52. ! -first_rank     1
53. ! -last_rank      1
54. ! -connect_boundary false
55. ! -value_in_range 127
56. !
57. !/write
58. ! -name           cl
59. ! -filename       "fe_mask
60. ! -compress_type  bin
61. !
62. !/gobj_maskaimpeel_ow
63. ! -input_output   in
64. ! -gobj_filename  "fe_mask
65. ! -peel_iter       0
66. !
67. !!!! end multi-value object
68.
69. /bounding_box_cut
70. -input            in
71. -output           out
72. -z_only           false
73. -border           0 0 0
74.
75. /write
76. -name             out
77. -filename         "fe_seg
78. -compress_type    bin
79.
80. ..
81. $!
82. $ FE_SOLVE:
83. $ IPL_FE_BATCH
84.
85. /read
86. -name             in
87. -filename         "fe_seg
88.
89. /fe_solve3
90. -in               in
91. -fea_file_name    "fe_files
92. -problem_nr       "fe_problem
93. -scale_factor     1.000000E+00
94. -output_option    1
95. -tolerance_force  1.000000E-03
96. -tolerance_displ  1.000000E-03
97. -max_nr_iter      10000
98. -restart_option    1
99. -comp_val_mat_001 127
100. -Ymodulus_mat_001 1.000000E+04
101. -Poissonr_mat_001 3.000000E-01
102. -comp_val_mat_002 0
103. -Ymodulus_mat_002 1.000000E+04
104. -Poissonr_mat_002 3.000000E-01
105. -comp_val_mat_003 0
106. -Ymodulus_mat_003 1.000000E+04
107. -Poissonr_mat_003 3.000000E-01
108.
109. /fe_post
110. -post_file_name   "fe_files
111. -output           con
112. -variable_nr      "fe_par_no
113. -loadcase_nr      1
114. -output_option    1
115. -interpol_option   1
116.
117. /exa con log
118.
119. /write
120. -name             con
121. -filename         fe_par
122.
123. ..
124. $ EXIT

```

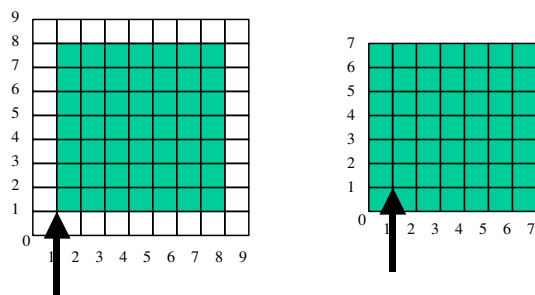


Fig. A1. When specifying absolute positions of coordinates for boundary conditions in a user-defined bcd file, using the *bounding_box_cut* option may results in errors. In the example above, the coordinate (1,1) is specified to define a force on the lower-left corner of the object. After removing the empty borders with the *bounding_box_cut* command, the coordinate system has changed and the force is applied to the wrong point.

Appendix C. (UNIX:) Sample input file for the definition of input parameters

```
aim_file_name      > default_file_name
fea_file_name      > default_file_name
problem_nr         > 9
scale_factor       >
output_option      >
tolerance_force    > 1E-3
tolerance_displ    >
max_nr_iter        > 50000
restart_option     >
nr_mat             > 1
comp_val_mat_1     >
Ymodulus_mat_1     >
Poissonr_mat_1     >
comp_val_mat_2     >
Ymodulus_mat_2     >
Poissonr_mat_2     >
```


40

41

problem number: 33 'High friction' compression test in the z-dir

```

1
0
6
11000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
12000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
13000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 1.0000d+00 1.0000d+00
11000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
12000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
13000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00

```

problem number: 37 Combination of standard problem 31, 32 and 33

```

3
0
6
11000 0 9999 0 9999 9999 9999 1.0000d+00 1.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
12000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
13000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
11000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
12000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
13000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
6
11000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
12000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 1.0000d+00 1.0000d+00 0.0000d+00 0.0000d+00
13000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
11000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
12000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
13000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
6
11000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
12000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
13000 0 9999 0 9999 9999 9999 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 1.0000d+00 1.0000d+00
11000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
12000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00
13000 0 9999 0 9999 0 0 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00 0.0000d+00

```

Appendix E. Sample listing file

```
=====
Table 1: Parameters:
-----
grid dimension in x-dir ..... 154
grid dimension in y-dir ..... 155
grid dimension in z-dir ..... 142

voxel size in x-dir ..... 5.5600E-02
voxel size in y-dir ..... 5.5600E-02
voxel size in z-dir ..... 6.0000E-02

number of specified materials ..... 1
component value of material 1 ..... 127
Youngs modulus of material 1 ..... 1.0000E+04
Poissons ratio of material 1 ..... 3.0000E-01

standard problem number ..... 23

scale factor ..... 1.0000E+00
maximum number of iteration ..... 20000
restart option ..... 0
force tolerance iterative solver ..... 1.0000E-04
displ. tolerance iterative solver ..... 1.0000E+00
output option ..... 1
=====
```

```
=====
Table 2: Conversion of voxels to elements
-----
Voxels | Elements
value | #voxels | mat. | #elements | Vv
-----+-----+-----+-----+-----
0 | 2868450 | | | |
127 | 521090 | 1 | 521090 | 0.15373
-----+-----+-----+-----+-----
Total: 3389540 521090
=====
```

```
=====
Table 3: FE-model
-----
number of elements ..... 521090
number of nodes ..... 834830
=====
```

```
*****
* Starting loadcase number 1 *
*****
```

```
=====
Table 4: Boundary condition definitions
-----
loadcase number: 1
code x-range y-range z-range a_x b_x a_y b_y a_z b_z
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
13000 0 154 0 155 142 142 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 1.0000E+00 1.0000E+00
11000 0 0 0 155 0 0 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
12000 0 154 0 0 0 0 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
13000 0 154 0 155 0 0 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
=====
```

```
=====
Table 5: Boundary conditions
-----
number of prescribed displacements
in x-direction ..... 0
in y-direction ..... 0
in z-direction ..... 11550
total ..... 11550

number of prescribed forces
in x-direction ..... 0
in y-direction ..... 0
in z-direction ..... 0
total ..... 0

total force
in x-direction ..... 0.0000E+00
in y-direction ..... 0.0000E+00
in z-direction ..... 0.0000E+00
total ..... 0.0000E+00
=====
```

```
=====
Table 6: Iterative solving
-----
iter Error_f = |Fres| / |Freact| Error_u = |delta x| / |x|
1 1.0402E+00 2.8378E+05 2.7280E+05 0.0000E+00 0.0000E+00 0.0000E+00
=====
```

```

2  9.4689E-01  1.2701E+05  1.3414E+05  5.9359E-01  3.0999E+02  5.2223E+02
3  6.2316E-01  8.3589E+04  1.3414E+05  3.3941E-01  2.0612E+02  6.0730E+02
4  9.3683E-01  6.3454E+04  6.7733E+04  2.5879E-01  1.7041E+02  6.5848E+02
5  7.3448E-01  4.9749E+04  6.7733E+04  1.9541E-01  1.3972E+02  7.1500E+02
6  6.1880E-01  4.1913E+04  6.7733E+04  1.6488E-01  1.2538E+02  7.6045E+02
7  5.2105E-01  3.5292E+04  6.7733E+04  1.3995E-01  1.1201E+02  8.0039E+02
8  8.8204E-01  3.0926E+04  3.5062E+04  1.1965E-01  1.0017E+02  8.3718E+02
9  7.8982E-01  2.7692E+04  3.5062E+04  1.0683E-01  9.3029E+01  8.7080E+02
10 7.0279E-01  2.4641E+04  3.5062E+04  9.3470E-02  8.4414E+01  9.0311E+02

```

```

.
.
. (listing file shortened)
.

```

```

7829 1.0043E-04  9.4498E-02  9.4089E+02  8.7597E-07  4.2473E-03  4.8487E+03
7830 1.0048E-04  9.4544E-02  9.4089E+02  8.8332E-07  4.2829E-03  4.8487E+03
7831 1.0053E-04  9.4590E-02  9.4089E+02  8.9526E-07  4.3408E-03  4.8487E+03
7832 1.0026E-04  9.4335E-02  9.4089E+02  8.9480E-07  4.3386E-03  4.8487E+03
7833 9.9620E-05  9.3731E-02  9.4089E+02  8.6845E-07  4.2108E-03  4.8487E+03
=====

```

Appendix F. Sample postlist file

Table 1: Parameters read from post file:

```

-----
grid dimension in x-dir..... 154
grid dimension in y-dir..... 155
grid dimension in z-dir..... 142

voxel size in x-dir ..... 5.5600E-02
voxel size in y-dir ..... 5.5600E-02
voxel size in z-dir ..... 6.0000E-02

number of materials ..... 1
component value of material 1..... 127
Youngs modulus of material 1 ..... 1.0000E+04
Poissons ratio of material 1 ..... 3.0000E-01

standard problem number ..... 23
standard problem name .....frictionless compression in z-dir
□
scaling factor ..... 1.0000E+00
number of loadcases for this problem .... 1
=====

```

Table 2: Conversion of voxels to elements

Voxels		Elements		
value	#voxels	mat.	#elements	Vv
0	2868450			
127	521090	1	521090	0.15373
-----+				
Total:	3389540		521090	

=====

Table 3: FE-model

```

-----
number of elements ..... 521090
number of nodes ..... 834830
=====

```

```

*****
*                               Processing loadcase number 1                               *
*****

```

Table 4: Average tissue strains and stresses

```

-----
Material number: 1
average strains      average stresses
eps_xx -5.4833E-02   sig_xx -7.4823E-04
eps_yy -5.4833E-02   sig_yy  8.7287E-05
eps_zz  1.8278E-01   sig_zz  1.8278E+03
eps_yz -2.7296E-06   sig_yz -1.0499E-02
eps_xz -4.6330E-07   sig_xz -1.7819E-03
eps_xy -1.3937E-06   sig_xy -5.3604E-03
=====

```

Table 5: Apparent strains and stresses

```

-----
apparent strains      apparent stresses
eps_xx -2.1481E-01   sig_xx -4.3911E-05
eps_yy -1.2253E-01   sig_yy -2.4612E-06
eps_zz  1.0000E+00   sig_zz  2.8099E+02
eps_yz -4.6090E-01   sig_yz -1.9894E-01
eps_xz -7.7492E-02   sig_xz -1.3642E-01
eps_xy -6.7951E-02   sig_xy  9.1292E-07
=====

```

Table 6: Nodal values

Face	average displacements			total force		
	x	y	z	x	y	z
x=min:	4.4620E-01	1.5849E-01	4.4636E+00	1.8633E-03	2.0426E-03	4.9805E-03
x=max:	-1.3931E+00	1.6124E-01	2.9391E+00	-4.5851E-03	4.2636E-03	-4.0055E+01
y=min:	3.0723E-01	4.0792E-01	5.2418E+00	4.2669E-03	-1.5985E-04	1.8226E+01
y=max:	-2.8115E-01	-6.4803E-01	3.7363E+00	2.3266E-03	-5.1895E-04	-3.9818E+01
z=min:	4.4242E-01	1.0551E+00	0.0000E+00	1.1270E-02	4.5830E-03	-2.0735E+04
z=max:	1.2992E+00	-1.3834E+00	8.5200E+00	4.0184E-03	-2.2085E-03	2.0735E+04

=====

```

=====
Table 7: Distribution selected parameter
=====
Parameter ..... 42
Average ..... 3.0488E+03
Standard Deviation ..... 2.6703E+03
Minimum value ..... 8.1243E-03
Maximum value ..... 7.1053E+04
Skewness ..... 2.4254E+00
Curtosis ..... 1.3127E+01
=====

```

Appendix G. Sample scanco_fe.ini file (with VMS settings)

```
# This files defines parameters that are read by the Scanco-FE program
# Do not edit these parameters!
```

```
#define operating system
```

```
os_name= 'VMS'
os_endian= 'LITTLE_ENDIAN'
os_float= 'VAX_FLOAT'
os_defrecl= 512
```

```
#define how post file is written
```

```
postfile_recl= 512
postfile_endian= 'LITTLE_ENDIAN'
postfile_float= 'IEEE_FLOAT'
postfile_version= 10
```

```
#define how restart file is written
```

```
restfile_recl= 512
restfile_endian= 'LITTLE_ENDIAN'
restfile_float= 'VAX_FLOAT'
restfile_version= 10
```

```
#define location of some files
```

```
spfname = 'disk1:[bert.fe_64]standard.bcd'
```

```
#define default values
```

```
def_file_name      = 'default_file_name'
def_problem_nr     = 9
def_scale_factor   = 1.0
def_output_option  = 1
def_tolerance_force= 1.0E-4
def_tolerance_displ= 1.0E-4
def_max_nr_iter    = 10000
def_restart_option = 1
def_nr_mat         = 1
def_comp_val_mat   = 0
def_comp_val_mat1  = 127
def_Ymodulus_mat   = 1.0E4
def_Ymodulus_mat1  = 1.0E4
def_Poissonr_mat   = 0.3E0
def_Poissonr_mat1  = 0.3E0
def_variable_nr    = 42
def_loadcase_nr    = 1
def_interpol_option= 1
```

Parameter index

A

[aim_file_name](#), 11

C

comp_val_mat, 11, 13

D

def_comp_val_mat, 33, 47
 def_comp_val_mat1, 33, 47
 def_file_name, 33, 47
 def_loadcase_nr, 33, 47
 def_max_nr_iter, 33, 47
 def_nr_mat, 33, 47
 def_output_option, 33, 47
 def_Poissonr_mat, 33, 47
 def_Poissonr_mat1, 33, 47
 def_problem_nr, 33, 47
 def_restart_option, 33, 47
 def_scale_factor, 33, 47
 def_tolerance_displ, 33, 47
 def_tolerance_force, 33, 47
 def_variable_nr, 33, 47
 def_Ymodulus_mat, 33, 47
 def_Ymodulus_mat1, 33, 47

F

fea_file_name, 11, 12, 13, 18, 21, 23, 24, 32, 33, 34

I

ihom, 18, 21, 22, 30, 31
[in](#), 11

M

max_nr_iter, 11, 12, 13, 24

N

nr_loadcases, 14, 22

O

os_defrecl, 32
 os_endian, 32
 os_float, 32
 os_name, 32
 output_option, 11, 12, 24, 25, 26, 30

P

Poissonr_mat, 11, 13
 post_file_name, 26
 postfile_endian, 32
 postfile_float, 32
 postfile_recl, 32
 problem_nr, 11, 12, 14, 17, 18, 21, 22

R

restart_option, 11, 12, 13, 24
 restfile_endian, 33
 restfile_recl, 33

S

scale_factor, 11, 12, 17, 20
 spfname, 33, 47

T

tolerance_displ, 11, 12, 24
 tolerance_force, 11, 12, 24

X

xfst, 19, 21, 22, 23, 24
 xlst, 19, 21, 22, 23, 24

Y

yfst, 19, 21, 22, 23, 24
 ylst, 19, 21, 22, 23, 24
 Ymodulus_mat, 11, 13

Z

zfst, 19, 21, 22, 23, 24
 zlst, 19, 21, 22, 23, 24