ELMER Front’s User Guide

ELMER Front's User Guide

CSC – Scientific Computing Ltd.

1st edition

http://www.csc.fi/elmer

About this Manual

The intended audience for the User Guides of ELMER are engineers and scientists in universities, research institutes and industry. The documents aim at offering a detailed approach to the use of ELMER software. For the first-time user, having basic knowledge on mathematical modeling, numerical methods and computer science, it acts as a start-up. Experienced users can utilize the User Guides in problem solving, teaching and training.

ELMER Front's User Guide provides information on ELMER graphical user interface. It leads the user to create mathematical models, and control existing numerical and visualization tools.

The present ELMER User's Guide corresponds to ELMER software version 1.4 for Windows NT and Unix platforms.

ELMER Front's User Guide will be delivered via the World Wide Web. Latest documentations and program versions of ELMER are available at http://www.csc.fi/elmer.

ELMER Installation

ELMER software is delivered on CD-ROM with the installation instructions. After inserting the CD-ROM into the computer, the ELMER software will be automatically installed by the auto install program.

As user receives a new ELMER version update, all user applications must be carefully checked against any discrepancies between old and new versions. The software provider does not guarantee congruent results between different versions.

The operating system can be either Windows NT 4.0 or Unix. It is recommended that the user has a computer system containing 64 Mb memory at minimum. Installed ELMER version requires approximately 40Mb disk space.

Contents

1 Overview **1**
1.1 ELMER Capabilities 1
1.2 ELMER Structure 2
1.3 ELMER Work flow 6

2 ELMER Front **7**
2.1 Input to ELMER Front 7
2.2 ELMER Front main window 11
2.3 General user interface features 13
2.4 Starting ELMER Front 17
2.5 ELMER Front settings file 18
2.6 ELMER Front definition file 18
2.7 File menu 19
2.8 Edit menu 22
2.9 Display menu 31
2.10 Problem menu 33
2.11 Model menu 48
2.12 Mesh menu 59
2.13 Solver menu 69
2.14 Run menu 72
2.15 Window menu 77
2.16 Help menu 78
2.17 Model window toolbar 80
2.18 Model Window 81
2.19 Control toolbar 82
2.20 Message area 83
2.21 Status area 83

3 ELMER Geometry File format **85**
3.1 Introduction 85
3.2 The format 87
3.3 An example file 90

4 ELMER Front settings file format **93**

5 ELMER Definitions File format 96
5.1 Introduction 96
5.2 ELMER Front model file and user definitions . . . 97
5.3 Loading definition files 98
5.4 The format 98
5.5 An example file 104

6 ELMER Mesh Generator input file format 110
6.1 Model description 110
6.2 Mesh density control 110
6.3 File structure 111
6.4 The header 112
6.5 The vertices 112
6.6 The edges 113
6.7 The bodies 114

List of Figures

1.1 ELMER Front 3
1.2 ELMER Solve 4
1.3 ELMER Post 5
1.4 ELMER workflow 6

2.1 Model files 11
2.2 ELMER Front main window 12
2.3 ELMER startup command under Microsoft Windows NT 17
2.4 File menu 19
2.5 Edit menu 22
2.6 Bodies window 23
2.7 Boundaries window 24
2.8 Settings window 28
2.9 Problem menu 34
2.10 Model name and directories window 34
2.11 Datafiles window 36
2.12 Coordinate settings window 37
2.13 Timestep settings window 39
2.14 Equations window 43
2.15 Advection-diffusion equation 46
2.16 Model menu 48
2.17 Body forces window 50
2.18 Materials window 51
2.19 Initial conditions window 52
2.20 Parameter selection example 53
2.21 Argument variable list 55
2.22 Boundary conditions window 57
2.23 Mesh element types 60
2.24 Mesh example 61
2.25 Define mesh window 62
2.26 Mesh structure window 64
2.27 Mesh density window 67

2.28 Mesh logfile example 68
2.29 Solver settings window 69
2.30 Advection-diffusion solver example 70
2.31 Processor settings window 72
2.32 Run menu 73
2.33 Process table 76
2.34 Window list example 77
2.35 Help menu 78
2.36 Graphics info window 78
2.37 System info window 79
2.38 Model window toolbar 80
2.39 Model window example 82
2.40 Status area 84

Chapter 1

Overview

ELMER software development is a joint project between Finnish universities, research laboratories and industrial companies. The main objective is to create a finite element method based program frame, which can be applied and further developed by academic and industrial communities in research and development, and in education.

ELMER development was initialized in the Finnish national computational fluid dynamics (CFD) technology programme in 1995-1999, funded by Tekes, the Finnish technology agency. At the first stage, the most active collaborators were CSC (responsible organization in ELMER development), Helsinki University of Technology, Jyväskylä University, Okmetic Ltd., and Technical Research Centre of Finland, and the main application areas were closely related to semiconductor and paper machine industries.

Currently, CSC continues its coordinative role in ELMER development. Development is in active phase, and new application areas are continuously founded. These include for instance micro electro mechanical systems (MEMS), simulation of blood flow, casting problems and crystal growth in magnetic fields. Some parts of ELMER have even utilized in CAVE environment.

1.1 ELMER Capabilities

Since its creation, ELMER has progress from the ordinary CFD program towards a computational tool which can tackle multi-physics problems. ELMER solves problems like

- compressible and incompressible fluid flows
- non-Newtonian flows
- laminar and turbulent flows
- heat transfer by conduction, convection and radiation
- phase changes
- free surfaces
- stresses and displacements in elastic bodies
- vibration
- magnetohydrodynamics
- fluid-structure interactions

The open design of ELMER allows user to work on a simulation on a desktop on small scale before performing the full simulation on an appropriate machine. ELMER has

- open and flexible structure for user application development
- scalability to various computing platforms and operating system
- flexible and fast re-simulation with temporary values

1.2 ELMER Structure

ELMER is a collection of programs rather than a single monolithic application. It consists of

- ELMER Front (preprocessor)
- ELMER Solver (solver based on finite elements)
- ELMER Post (postprocessor)

ELMER modules can be used as an integrated process or independently. The structure of ELMER is divided into three components. The first part, ELMER Front, is a tool for initializing the computational process in ELMER (see Figure 1.1). ELMER Front generates finite element meshes, allows user to build mathematical models graphically, and finally produces input data for ELMER Solver. Also, it can communicate with some external softwares.

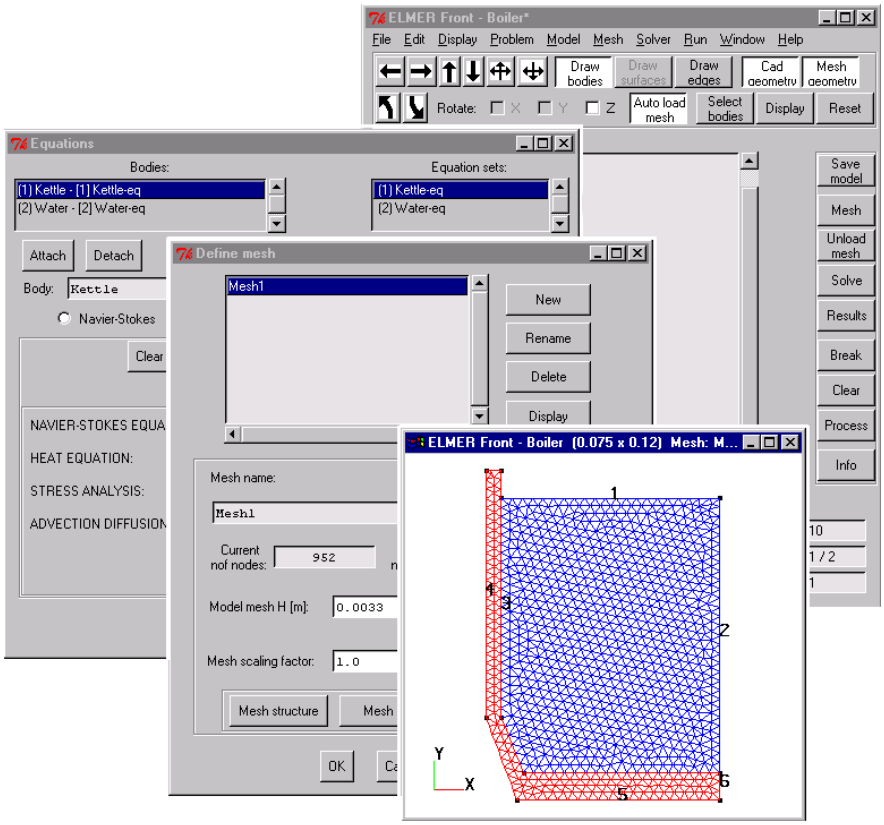


Figure 1.1: *ELMER Front*

ELMER Solver processes mathematical models into discrete form, handles coupled systems, nonlinearities and time-dependencies, and provides output data for ELMER Post.

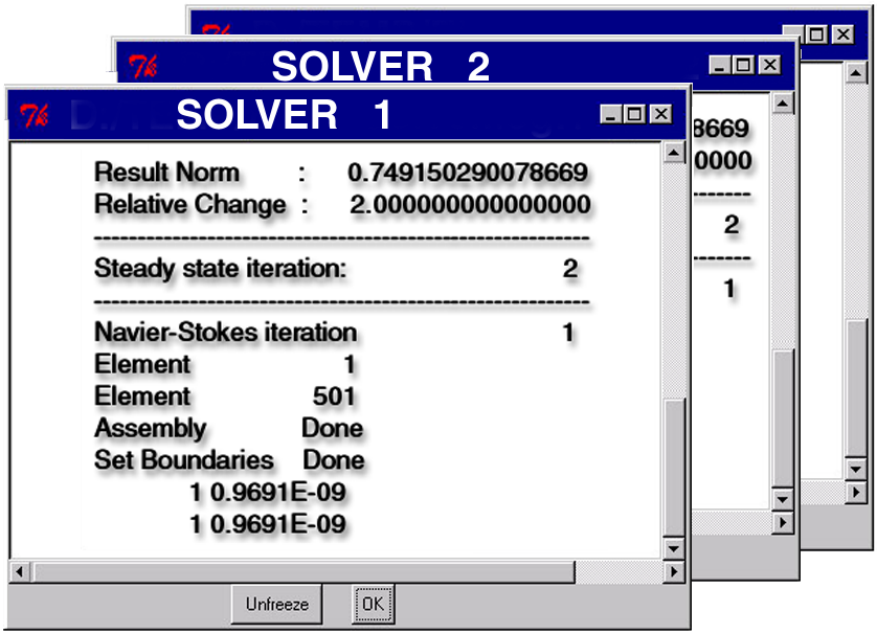


Figure 1.2: *ELMER Solve*

The function of ELMER Post is to visualize the numerical results produced by the solver. ELMER Post operates with the unknown variables and shows contours, vector fields, animations etc. on the screen. It can also manipulate computed data into another form (for instance heat fluxes from temperature distributions).

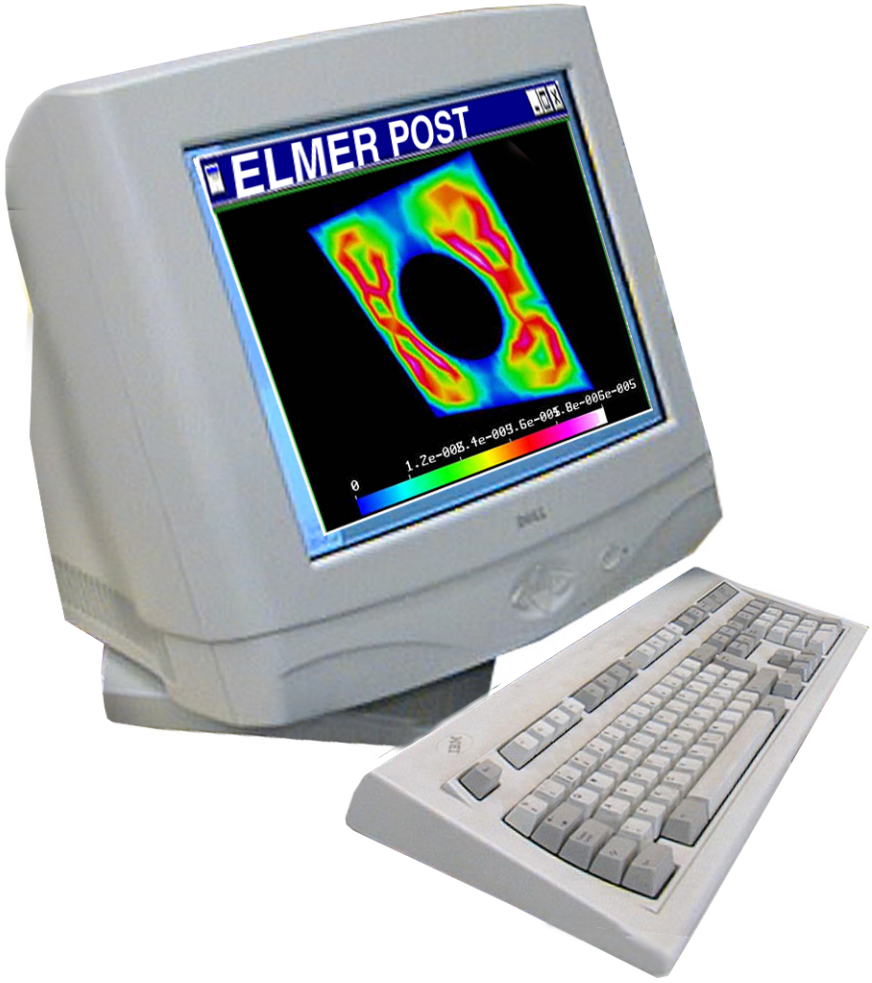


Figure 1.3: *ELMER Post*

Besides large variety of modeling, computing and visualization tools, ELMER also offers some special features, which may help in research work. These include, among others,

- interfaces to I-DEAS, ABAQUS and FLUENT
- MathC language for analyzing numerical results in ELMER Post
- computation of view factors and Gebhardt factors (modeling of radiation)

1.3 ELMER Work flow

The user can start ELMER software by clicking the ELMER icon on the computer's desktop, or writing the starting command on the command line. Then, ELMER work flow proceeds as shown in Figure 1.4.

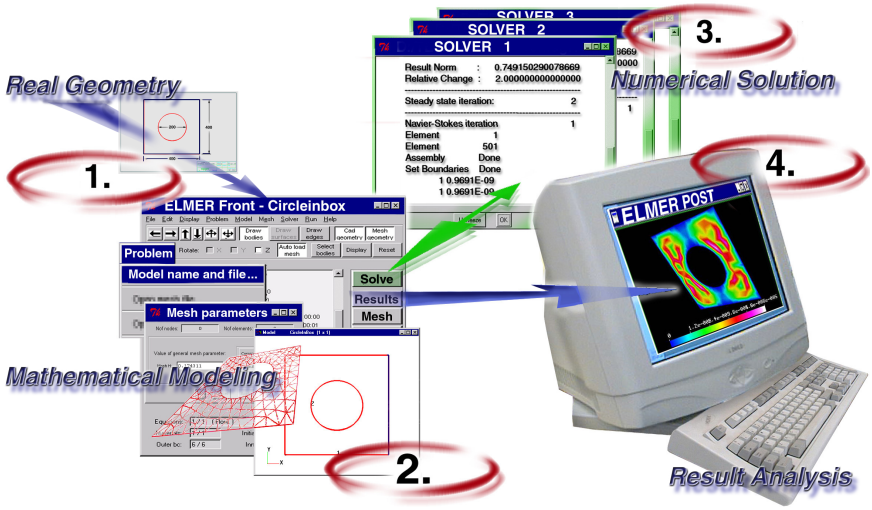


Figure 1.4: *ELMER workflow*

- As ELMER has been started, ELMER Front opens. User can now read in the file containing the model geometry **(1.)**, and start operating with the model shown on a separate Model window **(2.)**
- After all definitions (mathematical equations, mesh and material parameters,...) have been made, ELMER Solver can be started **(3.)**
- Solver output is used in ELMER Post visualization and results analysis **(4.)**

Workflow from **(1.)** to **(4.)** represents necessary/optional steps during a typical user session. Definitions in ELMER Front create a basis for numerical solution. The model can be solved in ELMER Solver after saving the model file, then by clicking the Solve button in ELMER Front. ELMER Post is started from the Results button.

NOTE! Notation from **(1.)** to **(4.)** in ELMER workflow is used as references later on in this manual.

Chapter 2

ELMER Front

ELMER Front is the Graphical User Interface module for ELMER. All essential problem solving definitions needed with other ELMER modules can be set in ELMER Front. First part of this chapter (sections 1.1 - 1.4) gives a general overview of ELMER Front. The rest of the chapter explains in detail all menus and commands needed in a typical ELMER Front session.

2.1 Input to ELMER Front

When a new model is to be built, in ELMER this starts by reading in the model geometry. This geometry must be first designed with some external program.

ELMER Front can read the geometry from three different sources:

- CAD geometry: a CAD file which has been created with an external CAD program or by other applications
- Mesh geometry: a mesh file which has been generated either by ELMER Mesh generator or by other applications
- Model file: a file produced and saved during a previous ELMER Front session

2.1.1 CAD file

A CAD file is an external file describing the model geometry in 2D wireframe format.

Supported CAD file formats are:

- *.unv (I-deas neutral CAD file format)
- *.igs (Iges file format)
- *.egf (Elmer Geometry File format). A text file format for creating basic 2D geometries. For more detailed description, see Appendix 3

Overview:

After a CAD file has been read in, ELMER Front creates internally a boundary representation from the input geometry.

A CAD model contains typically multiple parts, called bodies. The geometry input file should describe all the bodies (called also objects or materials) which are needed in the problem.

In 2D bodies are defined by edge loops. These loops should be closed and not intersecting with themselves or with any other edge loops. In 3D, bodies are defined by surfaces (surface patches) and restrictions are similar as in 2D. Surfaces are described by edge loops and again equivalent restrictions hold as for bodies in 2D. However, currently only 2D CAD geometries are supported. This is because current ELMER Mesh generator is restricted to 2D geometry.

Geometry definitions:

When defining a CAD geometry, the following restrictions should be taken into account:

- No change in the geometry is allowed after the external CAD geometry is read in. This means that bodies or boundaries cannot be split or combined
- Because boundaries cannot be split, a boundary segment defined by corner vertices is the lowest level where boundary conditions can be set
- Because boundaries cannot be recombined, a large number of boundary segments results in a long list of boundaries. This happens even though from the problem viewpoint the boundary segments could form a single physical boundary. To simplify the problem it is advisable to simplify the CAD geometry to contain as few boundary segments as possible

2.1.2 Mesh file

A mesh file (or a set of files) is an external file where bodies are described by volume elements and the surfaces by boundary elements. If no boundaries are given, they are formed at body surfaces. The mesh input file can be a 2D or 3D mesh file.

Supported input formats are:

- Abaqus mesh files (*.inp)
- Fidap mesh files (*.fdneut)
- I-deas mesh files (*.unv)
- ELMER mesh files (mesh.header).

Overview:

In ELMER external mesh files are saved in an ELMER specific format and the result files are named using the following scheme:

- *mesh directory*/mesh.header
- *mesh directory*/mesh.nodes
- *mesh directory*/mesh.elements
- *mesh directory*/mesh.boundaries

Here *mesh directory* refers to the directory where ELMER mesh files are stored. This directory is created automatically and it is formed from the model directory and the mesh name using the following scheme:

mesh directory = *model directory*/MESHDIR/*mesh name*

Geometry definitions:

Contrary to a CAD geometry, a mesh geometry can be edited. Mesh boundaries can be split and combined by regrouping mesh elements that define the boundaries. This procedure is explained later in the context of the Edit menu.

When using an external mesh file, a typical problem is that too few boundaries are available. Without any specific boundary information, the only way to define boundaries is to divide them into internal boundaries (boundaries between bodies) and outer boundaries based on body (material) information. With only one body this would mean that there is only one (outer) boundary for the whole model.

2.1.3 Model file

Model file is an internal ELMER file containing all the model information, including the model geometry, in ELMER Front specific format.

For ELMER Front the model file is the most essential file. If the model file is deleted, all model definitions are also lost.

At the very first stage, an external geometry is read into ELMER Front. When the user defines the model, these definitions are stored in a model file. Default name for this file is *modelname*.emf. (The extension .emf is an abbreviation for ELMER Model File).

When model file is saved, all the model information which is relevant for solving the problem is automatically stored also in a separate ELMER Solver control file. Default name for this file is *modelname*.sif (The extension .sif is an abbreviation for Solver Input File).

Both these file are stored in the model directory. This directory can be set using the Model name and directories command in the Problem menu.

2.1.4 Structural overview

Figure 2.1 gives an overview of all files created or used in ELMER during a model building and simulation session.

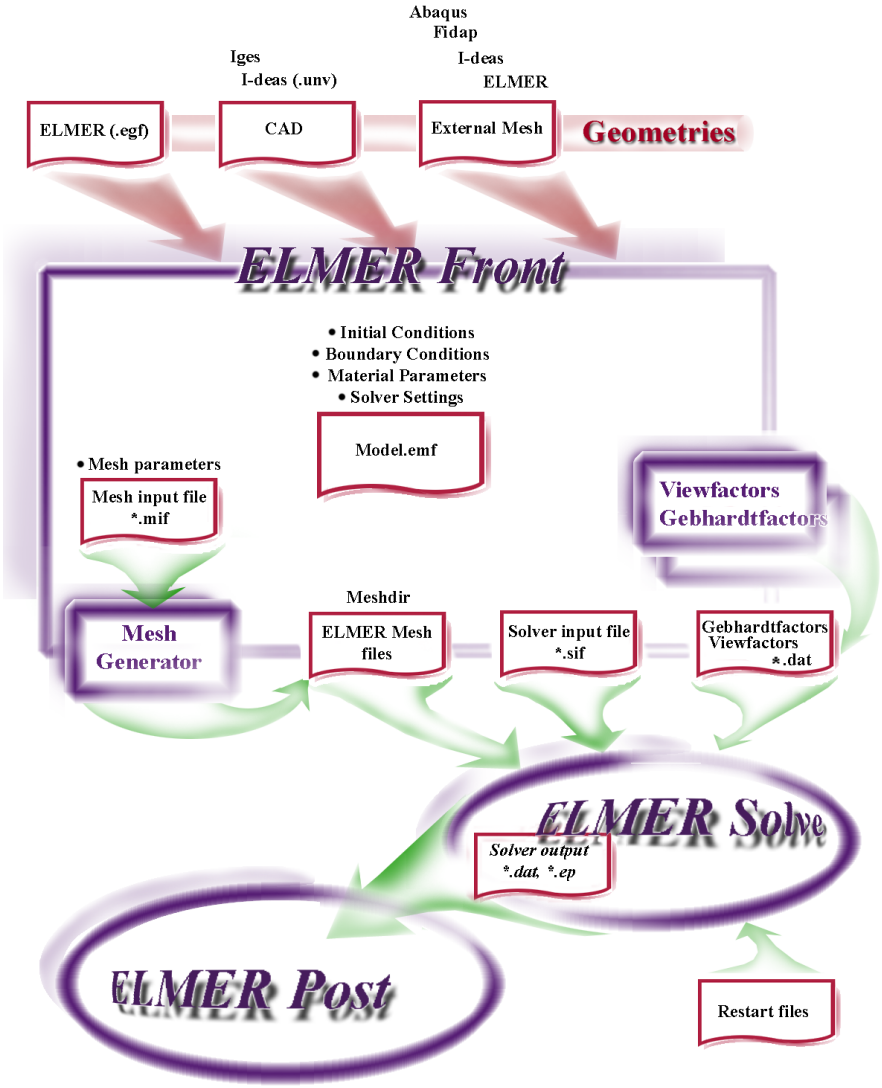


Figure 2.1: *Model files*

2.2 ELMER Front main window

ELMER Front main window is shown in Figure 2.2. The numbers shown in brackets refer to different areas in the main window. Window areas are the following:

- [1.] Titlebar
- [2.] Menubar

- [3.] Model window toolbar
- [4.] Prcoess control toolbar
- [5.] Message area
- [6.] Status area

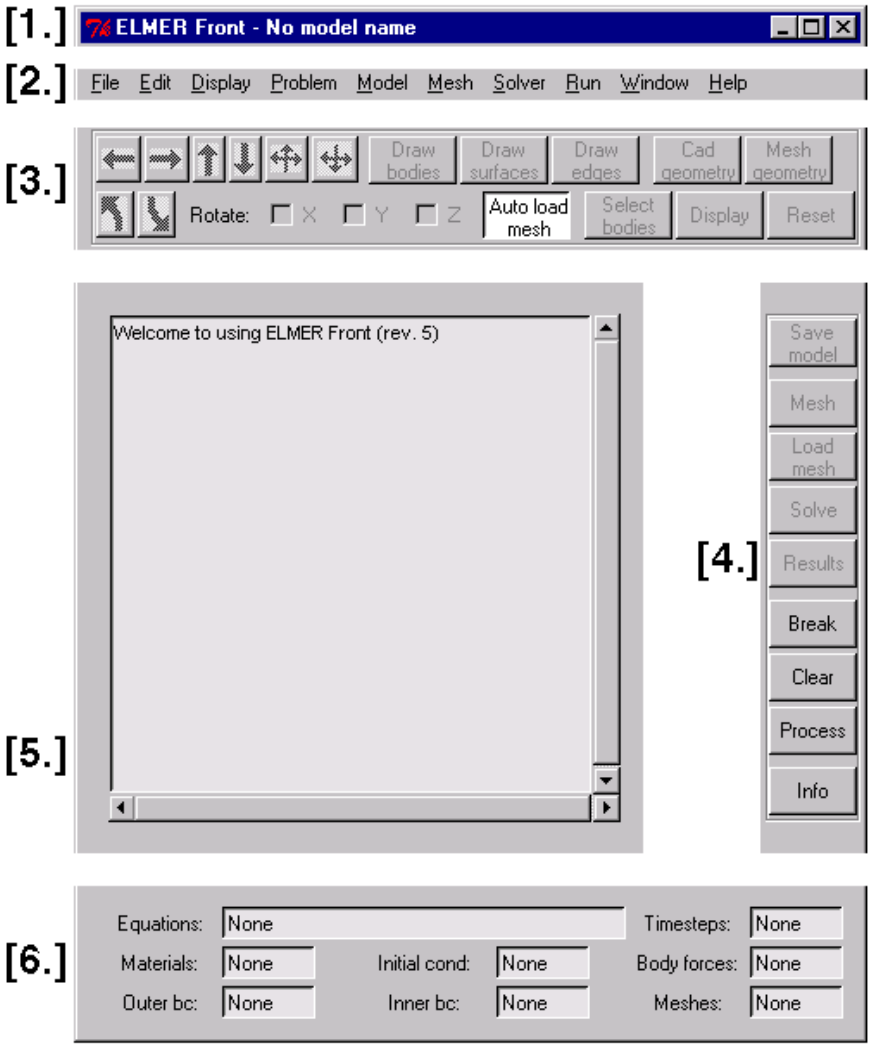


Figure 2.2: *ELMER Front main window*

The Model window toolbar [3.] contains command buttons which control how the model geometry is displayed in the Model window. This window is a separate graphics window for the model geometry and it is not shown in Figure 2.2. The Process control toolbar [4.] can be used to start solver, postprocessor and other processes. The

Message area [5.] in the middle of the window is used for displaying short messages related to commands and actions made during the session. The Status area [6.] at the bottom of the window tells the number of currently defined parameters and other status information for the model. In the example model parameters have not yet been defined as the 'None' values indicate.

2.3 General user interface features

This section gives an overview of the features which are common for all ELMER Front windows.

2.3.1 Standard window buttons

The window control buttons located at the bottom of each ELMER Front window have the following functions:

- Ok: accepts (if syntax is correct) and saves all possible changes made in the window. Note however that you still have to save the model file separately before changes are stored permanently. The window is closed
- Cancel: abandons all possible changes made in the window and closes the window
- Apply: same as the Ok button, but the window is left open

In general, windows will remain open until closed by the user with the Ok or Cancel buttons. Using the commands in the Window menu all open ELMER Front windows can be controlled in groups. For details, see the Window menu section in this chapter.

Modified data in entry fields is displayed using a red font until the modifications are accepted by pressing the Apply button.

In a window header, an asterisk displayed after the window title indicates that the data in the window has been modified. Pressing the

Apply button resets also this indicator.

2.3.2 Windows with body and boundary tables

A common factor to these windows is the Bodies listbox on the top left under the 'Bodies:' header. On the right side of the window is a listbox which contains the parameter sets which are entered in the panel. When these parameter sets are atteched directly to bodies, like the equations or material properties, only these two listboxes are visible. In some windows data is attached to the boundaries, and in these windows there is also a Boundaries listbox, which displays all the boundaries for the currently selected body.

In the following list the Boundaries, Boundary conditions and Mesh density window are boundary level windows, other windows are body level windows.

- *Edit menu:*

  Bodies

  Boundaries

- *Problem menu:*

  Equations

- *Model menu:*

  Initial conditions

  Body forces

  Materials

  Boundary conditions

- *Mesh menu / Define Mesh:*

  Mesh structure

  Mesh density

Note: These windows are designed to be used in conjunction with the Model window, because it helps to identify bodies and boundaries in the model. Refer to the Model window section in the end of this chapter.

When boundaries are displayed in the Model window they are numbered using the same id numbers that are used in boundary tables.

All these windows have common rules and features for the data entry, as explained below.

The first step in the data entry is to select the target body or boundary for the data. After that definitions can be entered using the data entry fields in the window. Some of the fields may however be disabled, because they are not applicable for the target body or boundary. For example, if the user is entering a material definition for a body, and the body does not have a flow equation defined, it would not be possible to enter any flow related material data for the body.

In many windows data fields are devided into subsets by equation type. These subsets can be activated and displayed using the equation type radio buttons in the area above the data fields.

Data in the currently active entry field can be reset to the previous value by pressing the Esc key on the keyboard. If a field supports entering multiple values, a space is used as a separator.

2.3.3 Attaching parameter sets to bodies and boundaries

When all necessary data for the new parameter set has been entered, it is added to the list of parameter sets using the Add button under the parameter listbox. If an existing parameter set is being modified, the update is activated only after the Update button is pressed. The Delete button removes the currently selected entry from the listbox.

Each paramter must have a unique name. In general it is better to use descriptive names for parameter sets. However, it is also possible to use the deafult automatic naming scheme where parameters are named like: 'BodyForce1', 'BodyForce2', ... If no parameter name is given, or the name is given is this basic form, the next available name in similar form is created automatically when the Add button is pressed. This practice is not recommended, but it be useful for quick experiments where the names are not essential.

If a target body or boundary does not currently have any parameter set attached to it, a new parameter set is automatically attached to the target body or boundary when the parameter is added to the listbox. Otherwise the user has to use the Attach button to attach the currently selected parameter to the currently selected body or boundary.

Only one parameter set can be selected at a time, but in some cases it is possible to select multiple target bodies or boundaries for the parameter. Multiple simultaneous selections are done by pressing the

control key when selecting the targets with the left mouse button. Pressing the shift key and the left mouse button while dragging with the mouse selects a continuous group of rows from the listbox.

When a parameter set is added to the listbox, it inherits certain properties from the active target body or boundary. In the Model menu windows these properties are related to the equation type of the active body. In the Mesh menu windows this property is the mesh structure of the body. This property inheritance creates some restircitions on how parameters can be attached to target bodies or boundaries. An attachment is not possible if the parameter and the target have conflicting properties. For example it is not possible to attach a material property for a body which has only heat equation defined, if this material definition was originally created for a flow body.

To identify this matching, an asterisk ('*') is used to mark parameter sets which match the selected body or boundary. This asterisk is shown in the beginning of the parameter row.

When a body or an boundary is selected, all those existing parameter sets which can be attach to the selected target are marked with an asterisk.

It is possible to select a parameter which is not marked with an asterisk. This parameter can be edited and updated, but it cannot be attached to the current target (the Attach button is not active), because it does not match this target.

When a parameter set is double-clicked, the first target, to which this parameter is possibly attached, is activated. The next double-click will activate the next target etc.

The buttons which are used to the handling of the paramter sets have the following functions:

- Add: adds a new parameter defined with the values in the data entry fields to the list of parameter sets

- Update: updates the currently active parameter with the values in the data entry fields

- Delete: deletes the selected parameter from the list and detaches the parameter from all the bodies or boundaries it is possibly attached to
- Attach: attaches the currently selected parameter to the currently selected body or boundary
- Detach: removes the current attachment from the currently selected body

2.4 Starting ELMER Front

ELMER Front starts by the 'ELMER' command. This command can be given in the command line or by using desktop menus or icons if they are available in the platform where ELMER is used.

Figure 2.3: *ELMER startup command under Microsoft Windows NT*

The following optional parameters may be used when the program is started:

- --model-directory=*model directory name*
- --model-name=*model name*
- --problem-name=*problem name*
- --settings-file=*settings file name*
- --definitions-file=*definitions file name*
- --results-directory=*results directory name*
- --include-path=*include path definition*

- --log-directory=*log directory name*

For example if ELMER Front is started using the command:

```
ELMER --model-directory=/ELMER/MODELS/stepflow
```

then the directory /ELMER/MODELS/stepflow would be the default directory for model files. If a model file stepflow.emf were stored in that directory, this command would automatically load this file into ELMER Front.

2.5 ELMER Front settings file

When an ELMER Front session is started, values for various control variables can be initialized from an input file (settings file). This file is loaded automatically if a file named 'front.esf' is found in the 'ELMER_HOME/lib' directory or if the user defined environment variable named 'ELMER_FRONT_SETTINGS' is set and contains any file name refering to a settings file.

The name for a settings file can be specified also in the command line when ELMER Front is started as explained above.

Settings files are searched and loaded in the above order and any later definition overwrites previous definitions.

The structure and the contents of this file is described in Appendix 4.

2.6 ELMER Front definition file

When an ELMER Front session is started, values for user defined equations and panel fields can be loaded from an input file (definition file). A definition file is loaded automatically if a file named 'front.edf' is found in the 'ELMER_HOME/lib' directory or if a user defined environment variable named 'ELMER_FRONT_DEFINITIONS' is set and contains any file name refering to a definition file.

The name for a definition file can be specified also in the command line when ELMER Front is started as explained in the previous section.

Definition files are searched and loaded in the above order and any later definition overwrites previous definitions.

The structure and the contents of the definition file is described in Appendix 5.

2.7 File menu

File menu commands are used for opening, saving and browsing the model data and related files. These commands are shown in Figure 2.7.

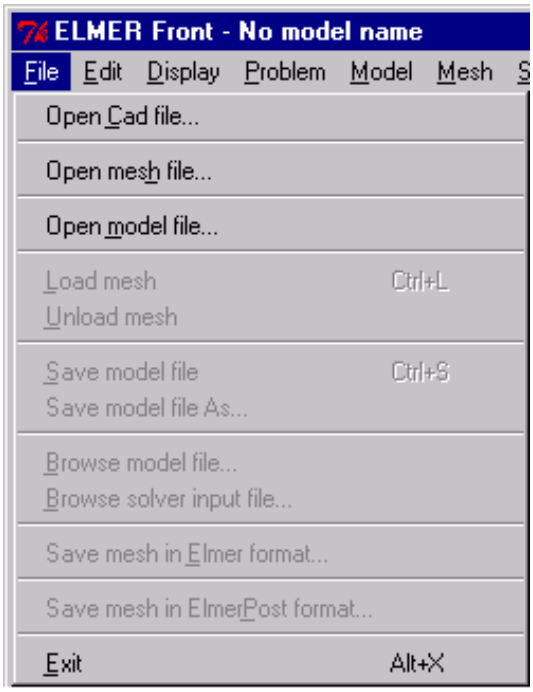


Figure 2.4: *File menu*

2.7.1 Opening commands

A new or an existing model is opened using the following commands:

- Open CAD file: read model geometry from an external CAD file
- Open mesh file: read model geometry and mesh from an external mesh file
- Open model file: read a previously saved model file (*.emf file)

If a model was previously opened and there is some unsaved data, an option is given to save the old model before loading a new one.

2.7.2 Load/Unload Mesh

Mesh can be loaded or unloaded during a ELMER Front session as follows:

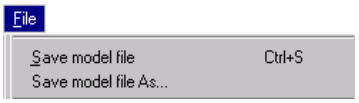
- Load mesh: if mesh file is available but currently not loaded, this commands reads the mesh into ELMER Front

- Unload mesh: this commands removes the mesh from the memory. Sometimes it may be useful to remove a large mesh from the memory before starting other ELMER modules

Note: If mesh is very large, it could be useful not to read in the mesh automatically. This autoload behaviour can be controlled using the Settings command in the Edit menu or through the Autoload mesh button in the main window.

2.7.3 Save model

The commands for storing model files and related data are the following:



- Save model file: saves model file (.emf file) in the model directory

- Save model file As: same as previous, but the name and location for the model file is asked instead of using the default name and location

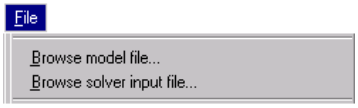
Note: When model definitions have been modified model data should be saved. Otherwise updated model data is not available for other ELMER modules. The model file is saved automatically when ELMER Solver is started (if Auto save model file setting is turned on, as it is by default). This concerns also all other processes started from the Run menu (excluding ELMER Post).

2.7.4 Browser windows

These windows are meant for browsing the files, file editing is not possible.

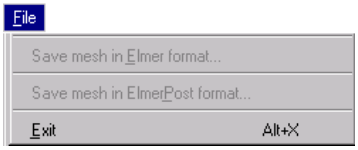
- Browse model file: displays the model file in a browser window

- Browse solver input file: displays the solver input file in a browser window



Note: Although model file is a normal text file, it is not meant to be edited 'off line'. The internal file structure is strictly predefined and any outside changes could make it unreadable for ELMER Front.

2.7.5 Save mesh



By default an external mesh is saved as an ELMER mesh when the model is saved.

Save mesh in ELMER format command can be used to save the external mesh explicitly if this default behavior is turned off.

Save mesh in ELMER Post format command saves the mesh in the ELMER postprocessor format

2.7.6 Exit

Exit command ends the current ELMER session. If there is any unsaved data, the user is given an option to save the model before closing the program.

2.8 Edit menu

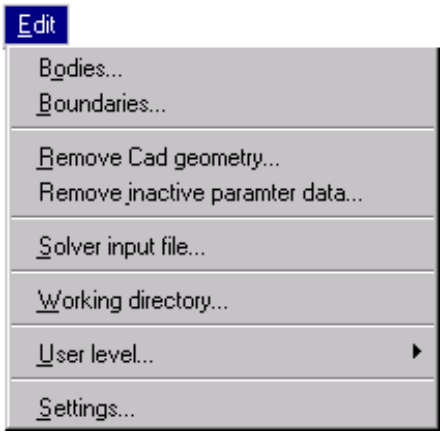


Figure 2.5: *Edit menu*

2.8.1 Bodies

Bodies command opens a window where body names and colors can be defined. This window is shown in Figure 2.6.

Default body names are in the form BodyN, where N is a sequence number. Because body names are used in the windows where data related to the body will be entered, they should be descriptive to make the identifying the bodies easier.

The name of the seleced body is displayed in the name entry field where it can be edited.

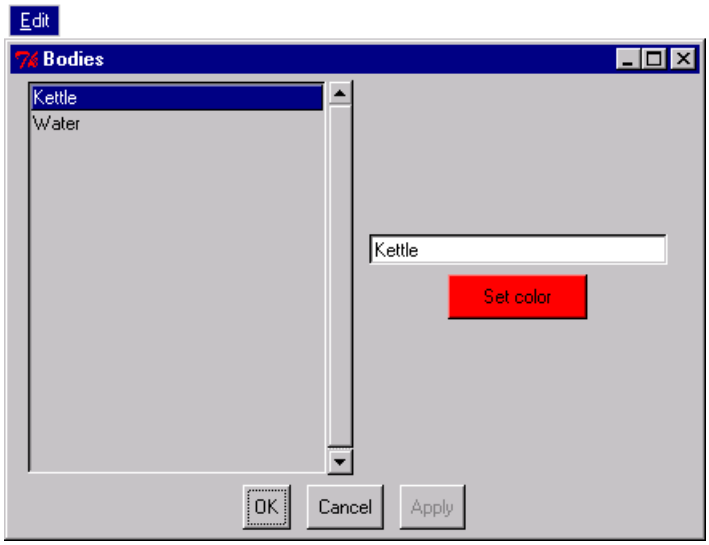


Figure 2.6: *Bodies window*

The color of the selected body color is shown on the Set color button. This color is used when the body is displayed in the Model window. (If the Model window is not currently visible, it can be opened by the Display model command in the Display menu or using the Display button in the main window). Pressing the Set color button opens a color palette where the color for the body can be defined.

2.8.2 Boundaries

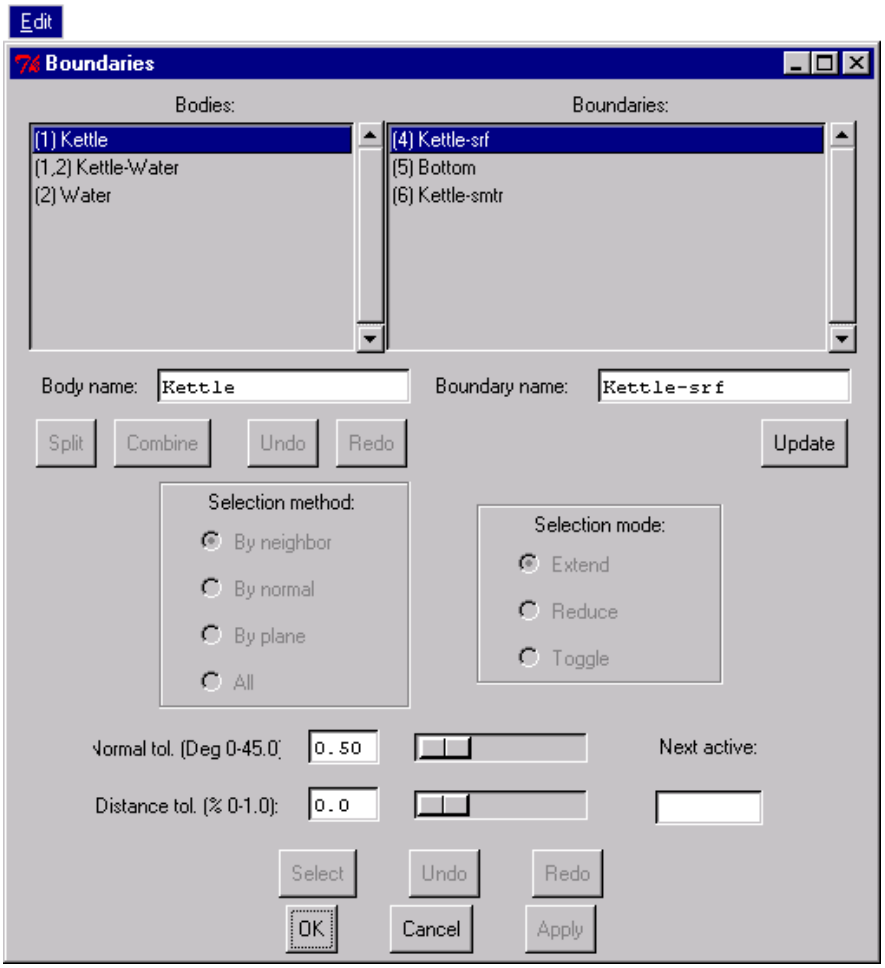


Figure 2.7: *Boundaries window*

Boundaries command opens a window where boundary names can are defined. If geometry is read from an external mesh file (i.e. model does not contain a CAD geometry), boundaries can also be edited by splitting and combining them.

Boundaries (edges or surfaces) of separate bodies can be adjacent. Such boundaries are called inner boundaries and the adjacent bodies are called body pairs. Other boundaries are called outer boundaries. When a body is selected, all its outer boundaries are displayed in the boudaries listbox. Inner boundaries can be displayed by selecting the corresponding body pair.

Inner and outer boundaries are sometimes called as boundary elements. They should not be confused with mesh boundary elements.

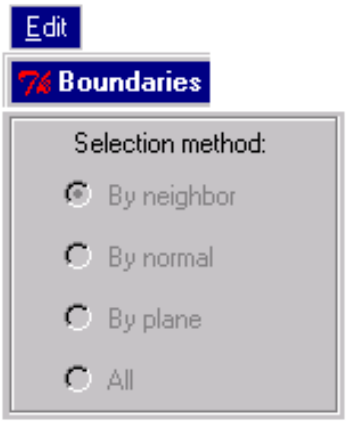
Boundary names are by default in the form BoundaryN, where N is a sequence number. A boundary can be activated by selecting it from the boundary listbox or from the Model window. The name of the selected boundary is displayed in the boundary name entry field, where it can be edited.

Editing boundaries can be done only for one body or bodypair at a time. Only one boundary can be split at a time. Only complete boundaries can be combined. Split and combine operations can be applied in any order and both operations can be cancelled using the undo command.

Multiple boundaries can be selected by pressing the Control key during a selection.

Note: See also the Model Window section at the end of this chapter.

2.8.3 Element Selection method



A split operation for a boundary is done by selecting a group of mesh elements in the boundary and by forming a new boundary with this group. The remaining mesh boundary elements stay with the original boundary.

Selecting the mesh elements can be done by two methods:

1. First method: elements can be selected from the graphics window (Model window) by 'painting' them directly. This is done by pressing the shift key, and painting with the mouse cursor while pressing the left mouse button. If control key is pressed instead of the shift key, mesh elements can be selected by clicking them with the left mouse button. Using the right mouse button cancels the current selections.

2. Second method: at first a reference mesh element is selected using the mouse. This base selection can then be extended by using the Select button in the window. The extended selection is controlled by the criteria set in the Selection radio button group. These criteria are the following:

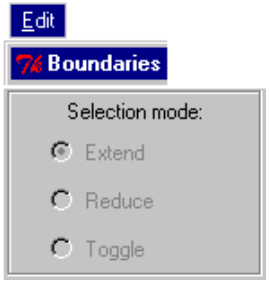
- By neighbor: selection is extended by adding neighbor elements for currently selected elements. Neighbor elements are added only if their normal vector changes less than what is set by the Normal tolerance slider.

- By normal: select all elements whose normal vectors differs less than the given tolerance from the normal vector of the reference element

- By plane: select all elements which are on the same plane as the reference element. Plane distance can differ within the tolerance which is set by the Distance tolerance slider.

- All: select all mesh elements in the boundary

2.8.4 Selection mode



Selection mode controls how the element selection actually behaves:

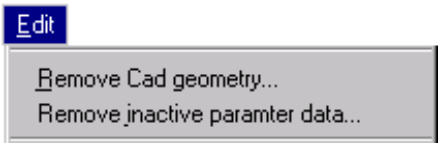
- Extend: always extends the current selection

- Reduce: always reduces (deselects) the current selection

- Toggle: toggle the selection state for the elements being effected

Value displayed in the Next active field tells the next tolerance value which would select new mesh elements.

2.8.5 Remove CAD geometry



Mesh boundaries cannot be modified if model geometry is defined by a CAD geometry. This limitation is implemented to prevent conflicts between the CAD and mesh geometries. However, if the original CAD geometry is removed from the model, the existing mesh can be modified as it were a normal external mesh.

The original CAD geometry is removed from the model using the Remove Cad geometry command. Note that remeshing is not possible if the CAD geometry is removed. It is also not possible to restore the CAD geometry, so it is recommended to save the original model and save the mesh only version in a different model file.

2.8.6 Remove inactive parameter data

An equation may be inactivated in the model. However, the parameters related to this equation are not erased by this. They just become idle and they will be reactivated if the equation is again turned on. This command removes these nonactive parameters permanently from the model file.

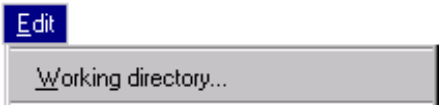
2.8.7 Solver input file

An existing solver input file is opened for editing with the Solver input file command. However, two requirements has first to be met:

1. User level has to be set to Power user. (See User level menu below)

2. The Auto save solver input file selection in the Settings window has to be checked off. By default all changes in the solver input file will be overwritten when the model is saved for the next time. This may be prevented by setting the Auto save solver input file checkbox off.

2.8.8 Working directory



The current working directory can be set using this command. This directory is used as the default model directory.

2.8.9 User level

The user level setting controls the amount of commands displayed in menus and the amount of warning messages displayed when working with the panels. There are three levels available. New users should first use the Novice level. This level simplifies menus by displaying only the most essential commands. It also protects againts unintentional loss of data by giving warning messages when modified panels are

closed without saving the data etc. The Power user level in contrast practically bypasses all warnings. The Advanced and Power user levels speed up the work for experienced users.

Note: The default level in the beginning of each session is the Advanced level.

2.8.10 Settings

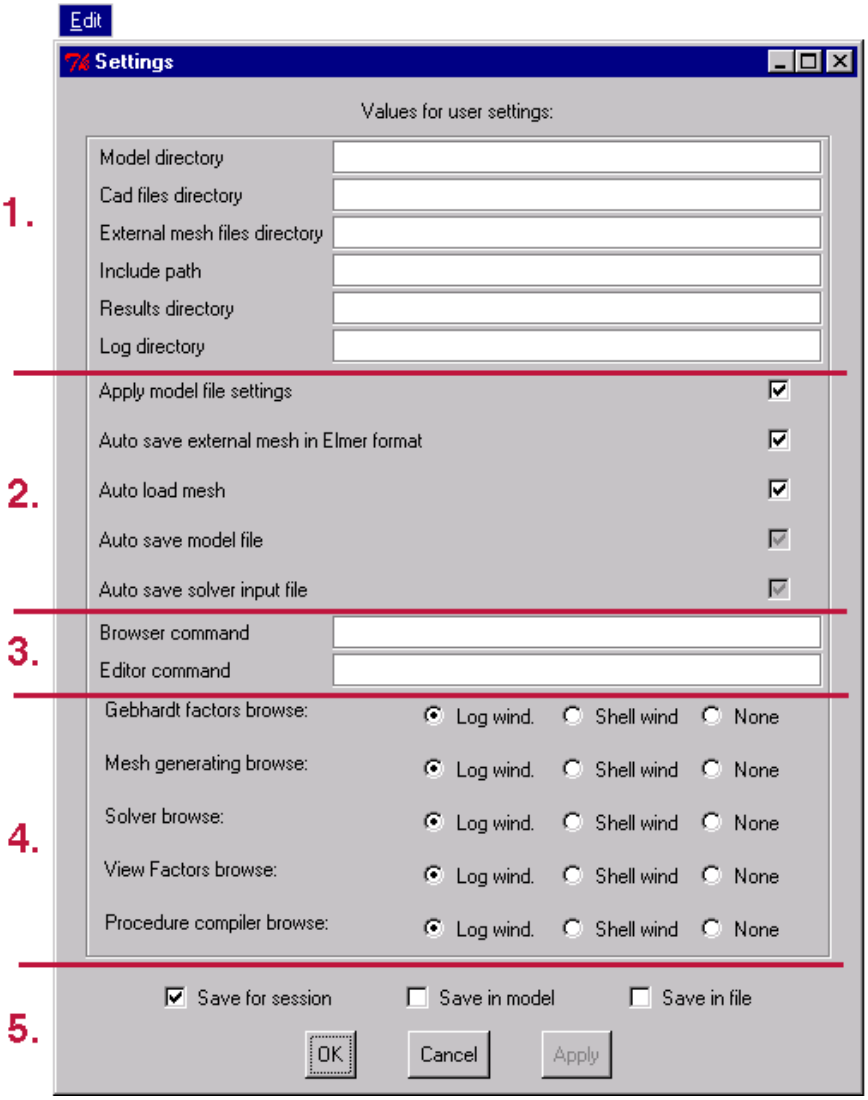
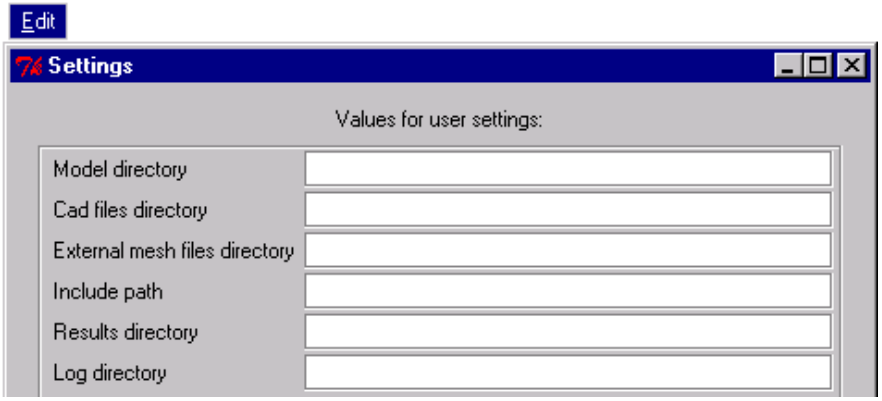


Figure 2.8: *Settings window*

Settings command opens a window where current problem settings can be modified.

The setting parameters can be devided into five groups. These groups are shown by numbers 1...5 in Figure 2.8 and they explained below in this order.

1. Directory definitions:



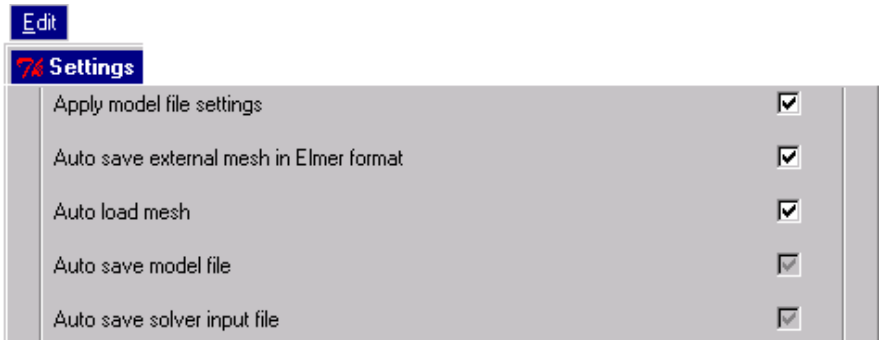
The default values for various model directories can be set and modified using these entry fields.

2. Save/Load control:

These parameters control various actions associated to the loading and saving of a model file.

- Apply model file settings forces the settings saved in a model file to be applied when the model file is loaded. Otherwise these settings are ignored

- Auto save external mesh in Elmer format forces the external mesh to be saved automatically in the Elmer mesh format when the corresponding model file is saved.



- Auto load mesh forces the possible mesh to be loaded always when a model file is loaded.

- Auto save model file forces the model file to be saved automatically when processes in the Run menu are started (excluding ELMER Post). This option is active only in the Power user level

- Auto save solver input file controls whether the solver input file will be saved when a model file is saved. This option is active only in the Power user level

Note: If the model contains a very large mesh and displaying the mesh is not necessary, it may be useful to turn off temporarily the mesh autoloading. This can be done by unchecking the first and third checkboxes seen in the figure above, and then selecting the Save for session command checkbox at bottom of this window and finally pressing the Ok button.

3. Browser and Editor:

- Browser command field can be used to define an external program for file browsing

- Editor command field can be used to define an external program for file editing.

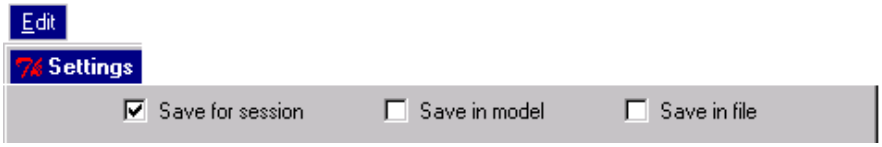
4. Data output methods:

These radio buttons are used to control where the output from other ELMER modules or the procedure compiler is directed or stored to:

- Log file: save output into a log file
- Shell window: direct output to the operating system shell window
- None: Nothing is displayed during the process

Note: Possible logfiles are stored in the Log directory This directory can be defined using the Model name and directories command in the Problem menu. Logfiles can be viewed in the process browser window (see the Process table command in the Run menu). Logfiles are automatically deleted when an Elmer Front session ends.

5. Save options:



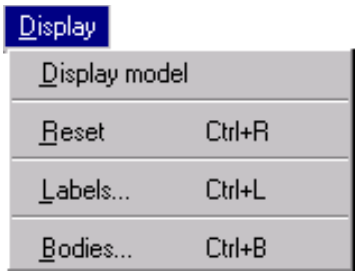
The following options are used to control how the parameter values set in this window are used:

- Save for session: current settings apply only in this session
- Save in model: settings are saved (when applicable) in the current model file
- Save in file: settings are saved in the current settings file

2.9 Display menu

Display menu is used to control the Model graphics window.

2.9.1 Display model



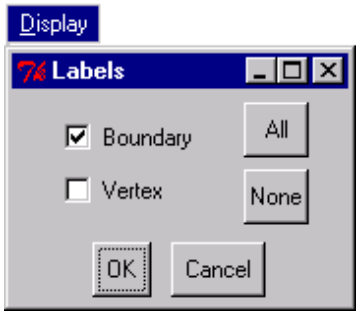
The Display model command can be used to open the Model window if it is not visible. This command is also attached to the Display button in the main window.

The model is displayed in a separate graphics window. For details see the Model window section in this chapter.

2.9.2 Reset

The Reset command sets the Model window into its initial state.

2.9.3 Labels



All body elements (inner and outer boundaries, edges and vertices) are uniquely numbered. These numbers can be displayed as element labels in the Model window. These same numbers are also used to identify the elements in the data entry windows.

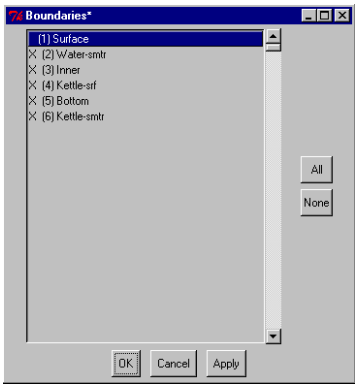
The Labels command opens a window where the lable display can be controlled.

In a complex model it is sometimes desirable to turn the labels off by unchecking the corresponding checkboxes. The All and None buttons control these labels in groups.

2.9.4 Bodies

Bodies to be displayed can be selected by using the Bodies menu command or the Select bodies button in the main window.

2.9.5 Boundaries



Boundaries to be displayed can be selected by using the Boundaries command in the Display menu. When a boundary is double-clicked in the listbox, its draw mode is turned off or on. The 'All' and 'None' settings can be activated by pressing the Apply or Ok buttons.

2.10 Problem menu

In this menu the actual problem definition is started. Most of the definitions here concern the whole model and are not specific to bodies or body elements.

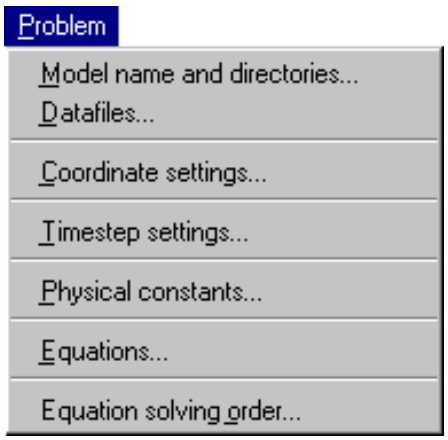


Figure 2.9: *Problem menu*

2.10.1 Model Name and directories

The Model name and directories command opens a window where the model name and the file directories can be entered. These directories are used when reading, saving or searching data files related to the model.

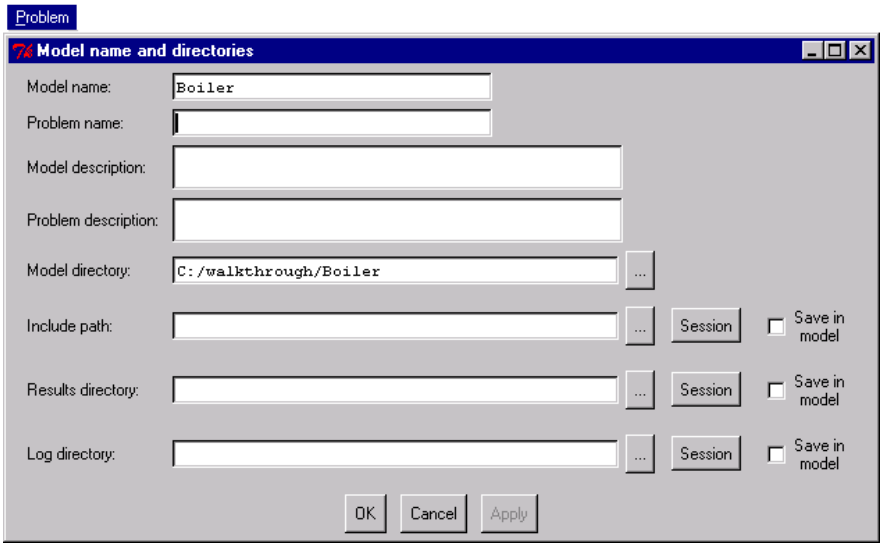


Figure 2.10: *Model name and directories window*

Note: it is possible to use spaces in the model and problem name, but because these names are used also in filenames, it may be safer to avoid spaces. Spaces can create problems especially when filenames

are used as arguments when calling external programs.

- Model name: this name must be given because it is used as a part of the model file name whose default value is: *modelname*.emf

- Problem name: this optional name can be used for example to identify different versions of the basic model. If given, it will be part of the model file name: *modelname.problemname*.emf

- Model description: this comment field is optional. It is displayed only in this window

- Problem description: this comment field is optional. It can be used to describe a specific version of the basic model. It is displayed only in this window

- Model directory: the root directory for the model files

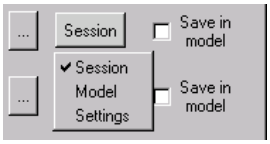
- Include path: A semicolon separated list of paths which are used as search directories when looking for parameter files. Parameter file can be used in the following windows: Initial conditions, Body forces, Material parameters and Boundary conditions

- Results directory: the root directory for the solver output files. These simulation result files are stored using the following directory structure: *result directory*/MESHDIR/*meshname*. The default value for this directory is the model directory.

- Log directory: The location for the log files created during the session, the defaul value is *model directory*/LOGDIR.

The directory browser button opens a browser window for selecting directories.

Note: Directory selection is done by single-clicking the directory name in the listbox and pressing ok button. Double-clicking the directory name opens a subdirectory list.



When this button is pressed a selection list opens. Currently active type is displayed as the button label. Available options are:

- Session: the value defined for the session is used

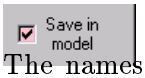
- Model: the value defined in the model file is used

- Settings: the value defined in the settings files is used

When the Save in model checkbox is checked, the active value is saved

in the model file.

2.10.2 Datafiles



The names for different input and output file can be defined in this window which is opened by the Datafiles command. This window is shown in Figure 2.11.

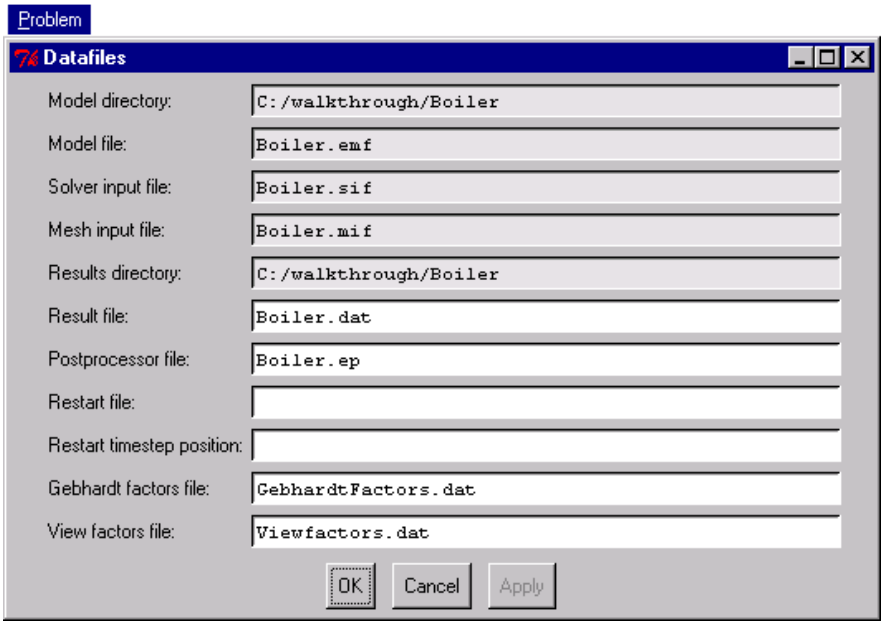


Figure 2.11: *Datafiles window*

- Model directory: (information field, not editable)
- Model file: (information field, not editable)
- Solver input file: (information field, not editable)
- Mesh input file: (information field, not editable)
- Results directory: (information field, not editable)
- Result file: solver output file, the default value is: *modelname*.dat
- Postprocessor file: this file stores solver results in the ELMER Post format, the default value is: *modelname*.ep.
- Restart file: if this file is given ELMER Solver uses the variable values in this file as initial values. This file should be in the output file format

- Restart timestep position: the timestep in the restart file which is used for the restart values. A zero value in this field means that the last time step is used.

- Gebhardt factors file: this file is used to store Gebhardt factors when they are calculated using the Gebhardt factors command in the Run menu. Default value is: GebhardtFactors.dat

- View factors file: this file is used to store view factors when they are calculated using the View factors command in the Run menu. Default value is: ViewFactors.dat

Note: Only simple file names can be entered in the file name fields, because directory structure is already predefined by the result directory.

2.10.3 Coordinate settings

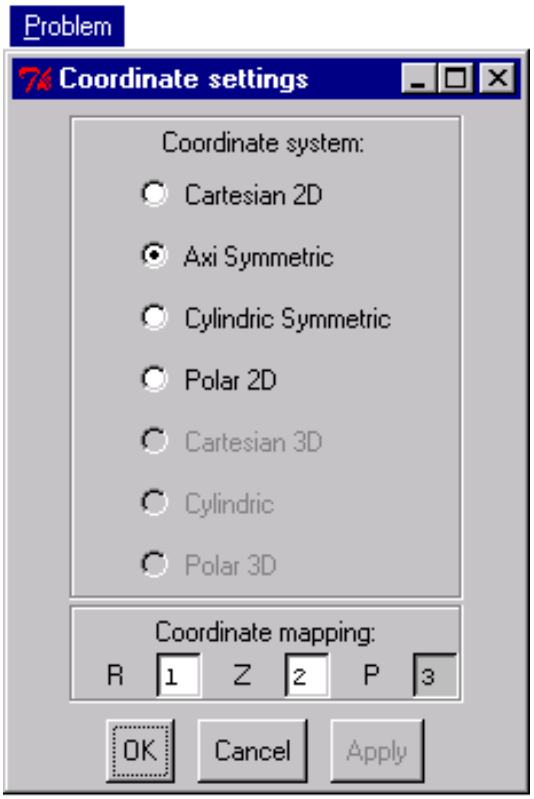


Figure 2.12: *Coordinate settings window*

Model's coordinate system is specified in this window which is opened with the Coordinate settings command.

Available coordinate systems depend on the dimension of the model geometry.

In 2D possible options are:

- *Cartesian 2D*
- *Axisymmetric*
- *Cylindric symmetric*
- *Polar 2D*

In 3D options are:

- *Cartesian 3D*
- *Cylindric*
- *Polar 3D*

Coordinate Mapping: Coordinate dependent variables (like velocity) are entered in data windows by components (like velocity-X, velocity-Y, velocity-Z). The default mapping between the components and the coordinate axes is X = 1, Y = 2 and Z = 3. This mapping can be changed entering a new index set in the Coordinate mapping fields.

The field labels for the entries are also altered when coordinate system is changed:

- *Cartesian: X, Y, Z*
- *Cylindric: R(ho), Z, P(hi)*
- *Polar: R(ho), T(heta), P(hi)*

These labels are also used in data entry windows to illustrate the coordinate system and the mapping currently in use.

2.10.4 Timestep settings

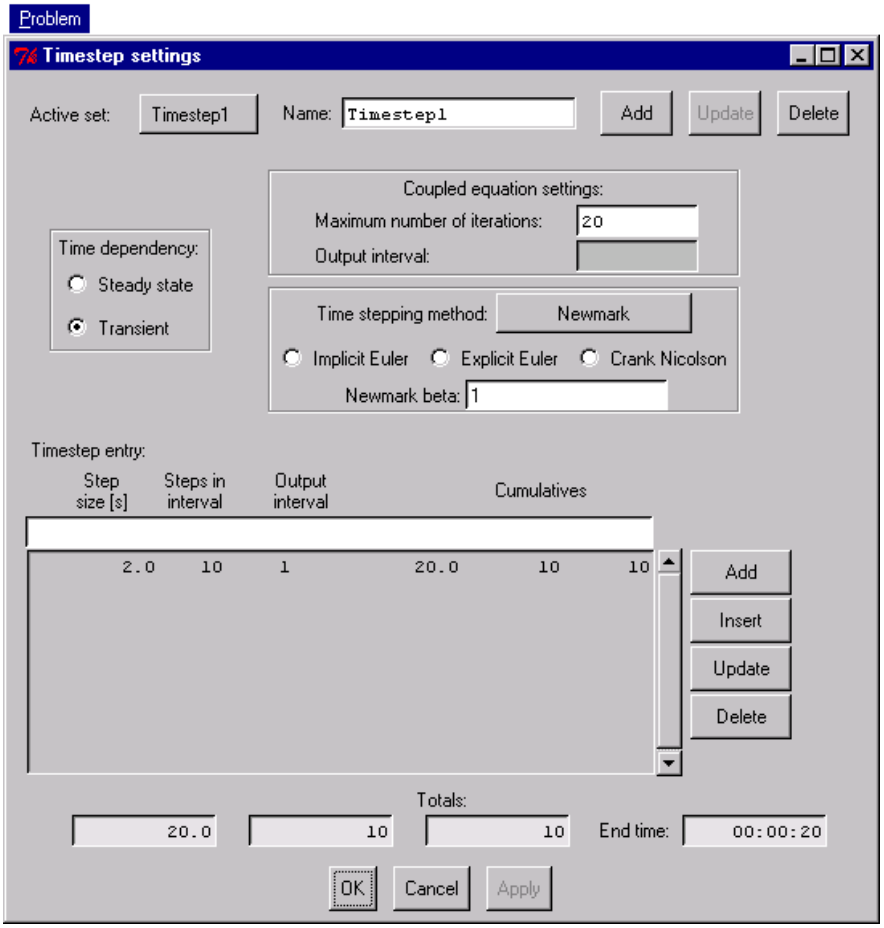


Figure 2.13: *Timestep settings window*

The Timestep settings window is used to set the time dependency and timestep settings for the simulation.

Active set button: selection list of the currently stored timestep sets. The name of the currently active set is displayed as the button label

Name: a user given name for the current timestep set

The Add, Update, and Delete buttons are used to add, update and delete timestep set entries.

The time dependency type of the current simulation problem is set in the Time dependency box. Options are Steady state and Transient.

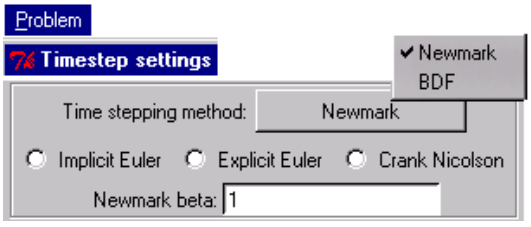
In the Steady state case, total iteration amount and the output fre-

quency are controlled by the following fields:

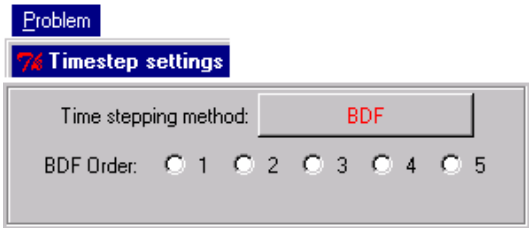
- Maximum number of iterations: Maximum limit for the steady state iterations (iteration is stopped even if the steady state convergence criterion is not yet met, if this limit is met first)
- Output interval: Output frequency for steady state iterations

For transient problems two basic timestepping schemes are available: Newmark and BDF.

In the Newmark method the beta parameter can be freely selected, but the three most common methods can be simply selected using the radio buttons for the Impilict Euler, Explicit Euler and the Crank Nicolson methods. These methods correspond the beta values 1.0, 0.0 and 0.5. Default value is 1.0 (Implicit Euler).



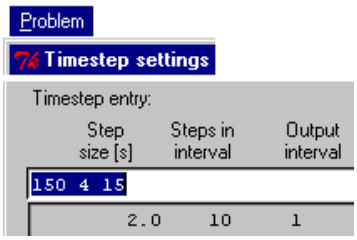
For the BDF method only one of the predefined values 1..5 can be selected.



Timesteps are defined by entering three numbers in the Timesteps field. They have the following meaning (values must be entered in this order):

- Step size: the step size in seconds in the interval
- Steps in interval: the number of steps in the interval
- Output interval: the output frequency in the interval

A new timestep interval can be added by entering three values separated with a space and then pressing the Add button. In the example step size is 150, steps in interval 4, and output interval 15 (because every fifteenth timestep should be output, this setting would not create any output from this interval!).

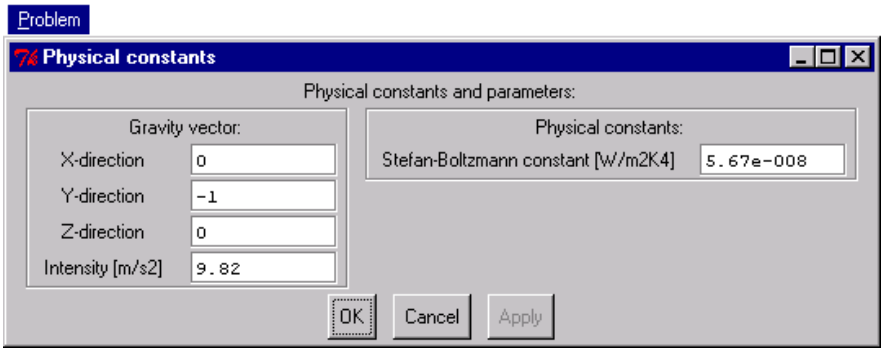


New entries are added at the end of the table when the Add button or the Enter key is pressed. The Insert button inserts a new entry above the currently selected row in the table. The currently active row may be updated by updating the values in the Timestep field and then pressing the Update button. The currently selected row may be deleted from the table by pressing the Delete button.

For each row in the list, running cumulatives are displayed as the last three numbers. Total cumulatives are shown in the Totals fields at the bottom of the window.

2.10.5 Physical constants

The Physical constants window is used to modify the default values for physical constants. Modifying these values may be necesary when for example input values are in nonstandard units. The constants are the gravity vector and the Stefan-Boltzmann constant.



The default gravity value is 9.82 m/s^2 in the direction of the negative y-axis.

2.10.6 Equations

Assigning equations for the bodies is perhaps the most important task in the problem setup in ELMER Front. Each body must have an equation attached to it before any equation related data (like material property sets, body force sets, initial and boundary condition sets) can be entered for the body or its boundaries.

A typical equation definition sequence could be the following:

- A body or bodies are selected

- An equation is defined;

  a descriptive equation name is entered

  an equation type is selected

  parameters for the equation are entered.

- When all data is entered, the new equation set is added to the list using the Add button

- The defined equation set is attached to the selected body (or bodies) using the Attach button

Equations window is shown in Figure 2.14.

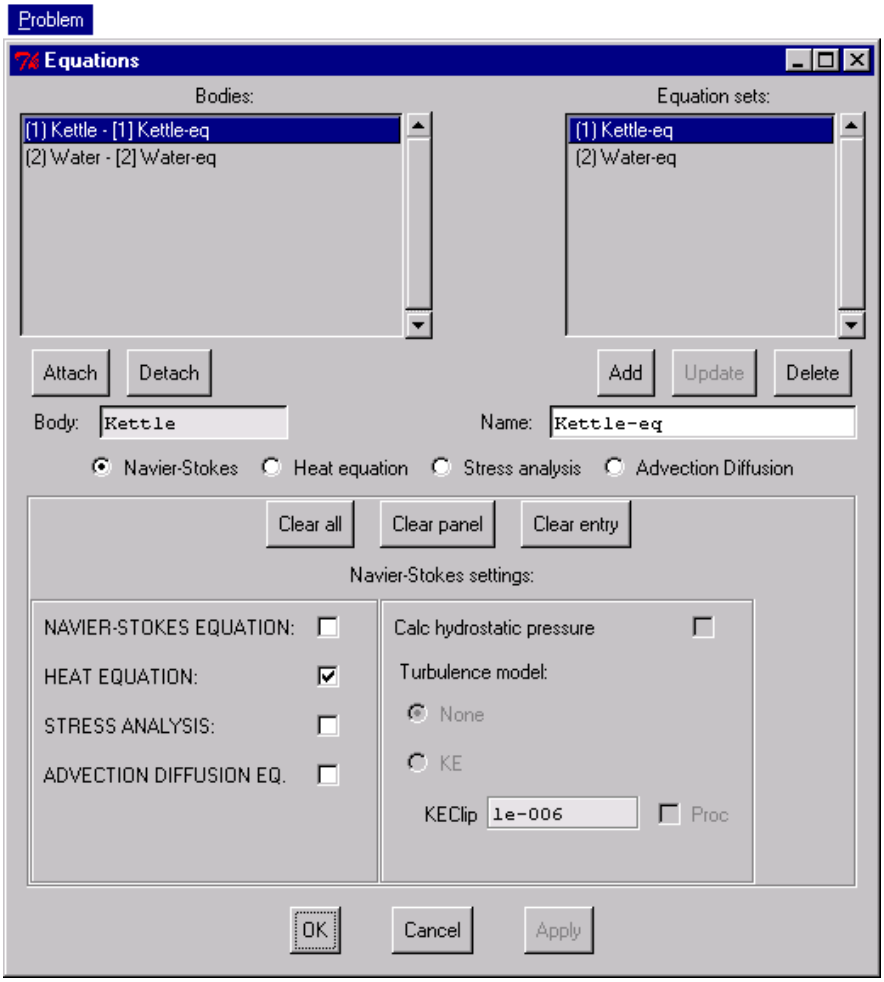


Figure 2.14: *Equations window*

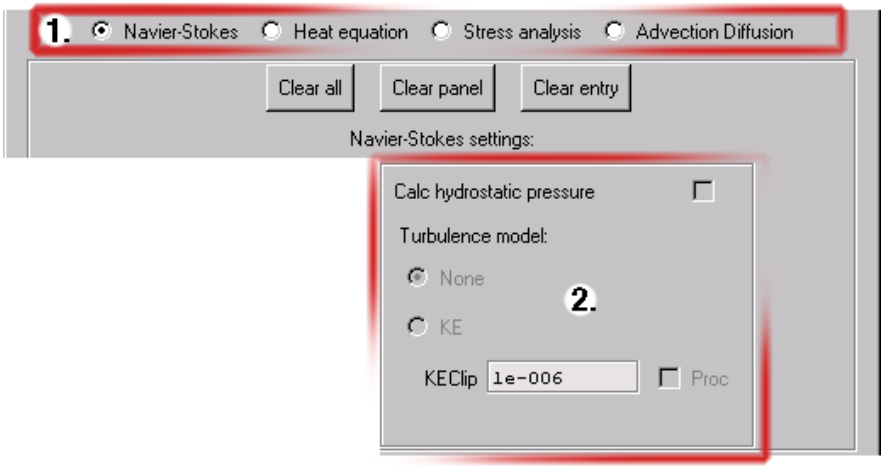
The name of the currently selected body is shown in the Body field This field is not editable.

The text for the selected body in the figure ((1)Kettle - [1]Kettle-eq), shows that the equation number 1 is attached to the body number 1. It also shows the body name (Kettle), and the equation name (Kettle-eq).

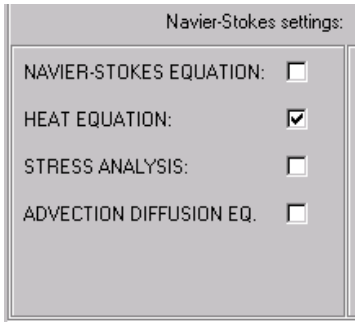
Name: in this field the user can enter or modify the name of an equation set. Pressing the Update button stores the name.

The currently defined equations are displayed in the Equation sets listbox. An automatically generated equation sequence number is shown at the beginning of each line. When an equation is attached to a body, this number is also displayed after the body name.

When parameters for a specific equation are entered, this equation must be first selected using the equation type radio buttons. Pressing a radio button on area 1 opens the appropriate input panel on area 2 where the equation is defined in detail. The Navier-Stokes equation has been checked in the example below.



Separate equation types in the currently active equation set can be turned on and off by using the checkboxes in the lower left part of the window. In the example, only the heat equation is activated, others are turned off.

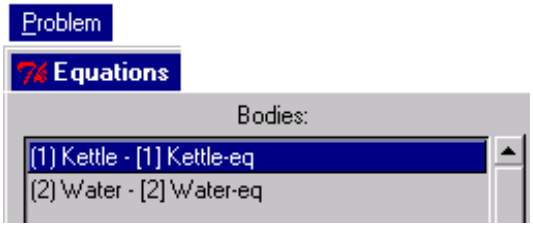


When solving an equation attached to a body, only those equation types are taken into account whose checkboxes are turned on. When an inactive equation type is made again active, all previously defined parameter values (in the input area 2) are automatically taken into use. Making an equation inactive may also affect other equations. For example, if the flow equation is turned off, all flow related fields (like

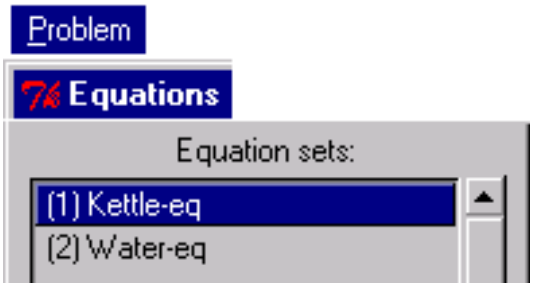
convection) are made inactive in other equations. So, it is a good idea to check all active equation types by using the equation type radio buttons, whenever an equation is made active/inactive by the checkboxes.

When the new equation set has been completely defined it can be added to the list of equation sets by pressing the Add button. Pressing the Update button updates the currently active equation set definition. The Delete button removes the currently active equation set from the listbox and also removes any attachments to bodies for this equation set.

2.10.7 Attachment



1. An equation is attached to a body by first selecting the body in the bodies list.



2. Next the equation is chosen.

3. Finally the attachment is done using the Attach button under the bodies listbox.

An equation may be attached to a single body or multiple bodies. There are no restrictions how bodies and equations are attached to each other, but when the attachement is done, the body inherits a

kind of equation mask from the equation. This mask is transparent to the user, but it controls how other properties can be attached to this body or its boundaries. For example, if no flow equation is defined for a body, it is not possible to attach any flow related material properties for this body in the material properties panel.

2.10.8 Advection-diffusion equation

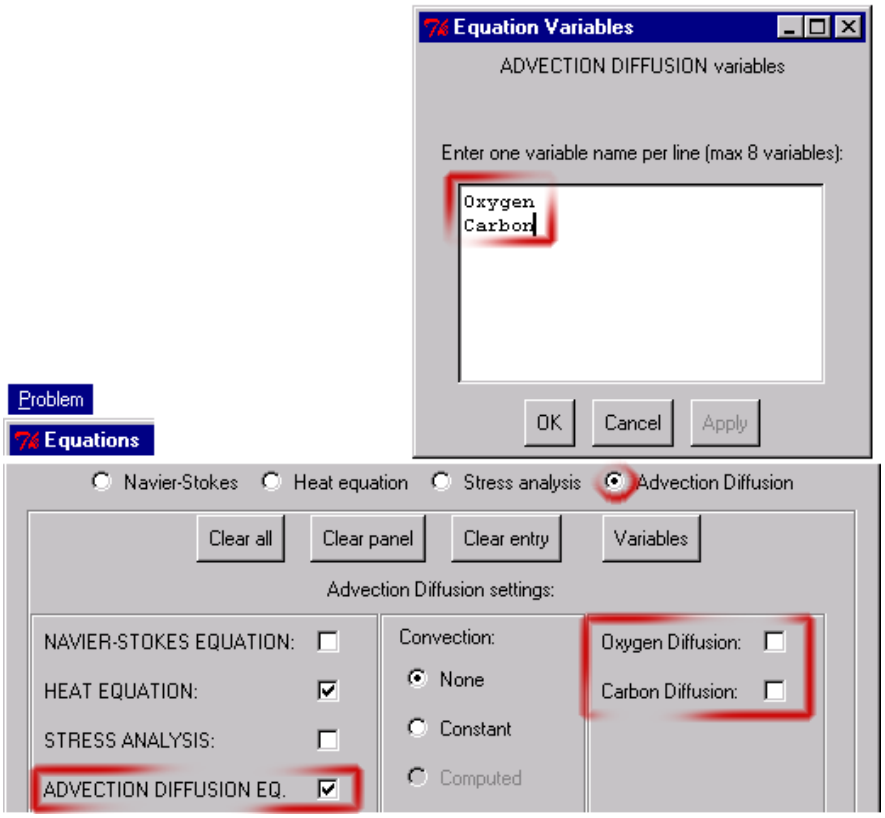


Figure 2.15: *Advection-diffusion equation*

When the advection diffusion equation is activated by checking the corresponding checkbox, a separate window for advection-diffusion variables is opened. In this window the user may enter or update the variable names for which the advection-diffusion equation should be solved. These variable names are displayed in the input panel area 2. for the adevetion-diffusion equation and the checkboxes in this area control if these variables are active or not in the current equation set. Each active variable activates also new entry fields in other panels

where advection-diffusion equation related data can be entered. Because each variable is solved by a separate solver, each active variable also creates its own entry in the Solver settings window.

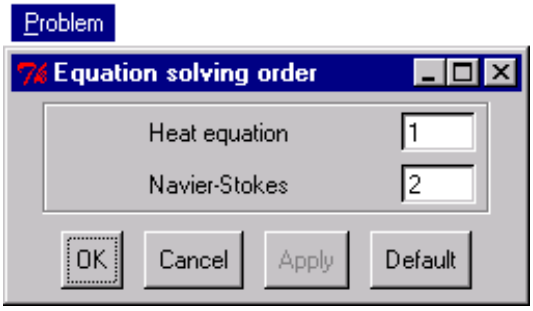
This window can be reopened using the Variables button which is visible only in the advetion-diffusion equation entry panel.

2.10.9 Data checking

As the user activates an equation, certain data integrity checks are automatically done. They are the following:

- Heat equation: heat equation includes automatically heat conduction and constant convection can be added (note: constant convection values are given in the material parameters window). The latent heat release can be selected only if some of the phase change models is activated.
- Navier-Stokes equation: when this equation is active, it is possible to select computed convection for the heat and the advection-diffusion equations.

2.10.10 Equation solving order



The Equation solving order is defined in this window. The order can be changed by entering new order numbers, These values can be any numbers, only their relative order is important. Only the currently active equation types are displayed in this window.

The default solving order is the following:

1. Heat equation

2. Navier-Stokes

3. Stress analysis

4. KE Turbulence

5. Advection-diffusion equations (a separate equation for each advection-diffusion variable)

2.11 Model menu

After all body equations and other general problem data have been entered, the model parameters are entered. This is done using the model menu commands.

Figure 2.16 shows the Model menu where also the body list submenu has been opened. Clicking the dotted line on the top of this submenu opens a separate body list window on the desktop.

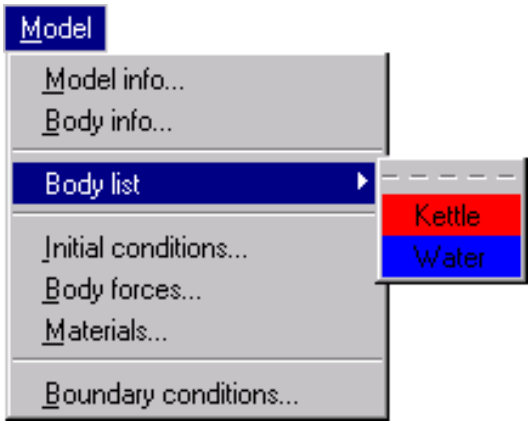


Figure 2.16: *Model menu*

2.11.1 Model info

The Model info command opens a window displaying general model information. Most of the fields are non-editable information fields but the following fields are editable:

- CAD input file: the name of the original CAD source file
- Mesh input file: the name of the external mesh input file
- Model created: model creation info

Values for these fields are set automatically when a model is created. They can be updated to reflect possible changes in the external file names, but it is not necessary because these values are not used for any actual processing of the model.

Other fields are either collected from other panels for reference or are simple model object counters, but Minimum edge size field needs some explanation. It is the smallest edge size which is used when a (linearized) CAD geometry is transmitted for the ELEMR mesh generator. This information may be useful when setting the mesh density parameters for the model.

Note: Nof is an abbreviation for 'Number of.'

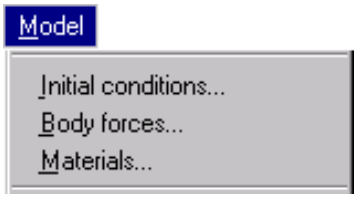
2.11.2 Body info

Body info command opens an information table where the physical dimensions and the number of mesh elements for each body can be checked. Data in this window cannot be edited.

2.11.3 Body list

Body list command opens an information table displaying a list of all model bodies. Button colors are the same as the body colors and body names are used as button labels. This list may be used to match the bodies in the data entry lists with the bodies in the Model window.

2.11.4 Initial conditions/Body forces/Materials



In the Initial conditions window the user can set initial values for model variables.

In the Body forces window the user can set the external forces affecting to model bodies.

In the Materials window the user can set material parameters for the bodies.

Next figures display the layout for these windows. The workflow in these windows is essentially the same and in the following the Initial conditions window is used as an example.

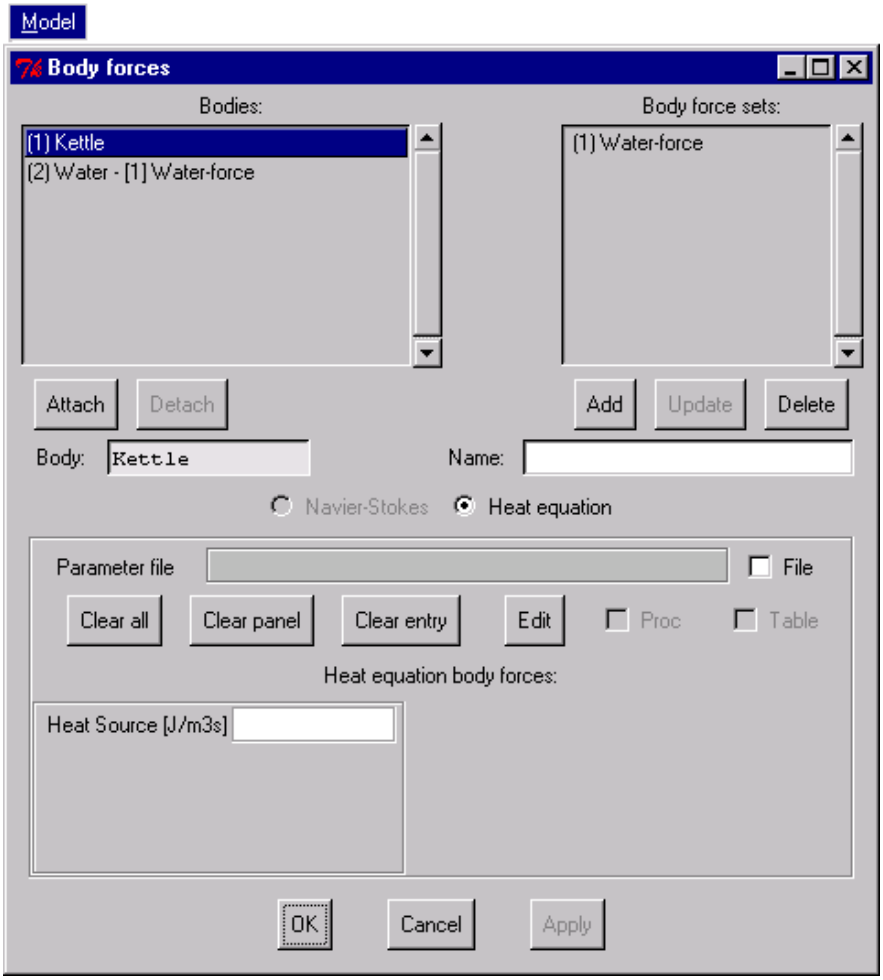


Figure 2.17: *Body forces window*

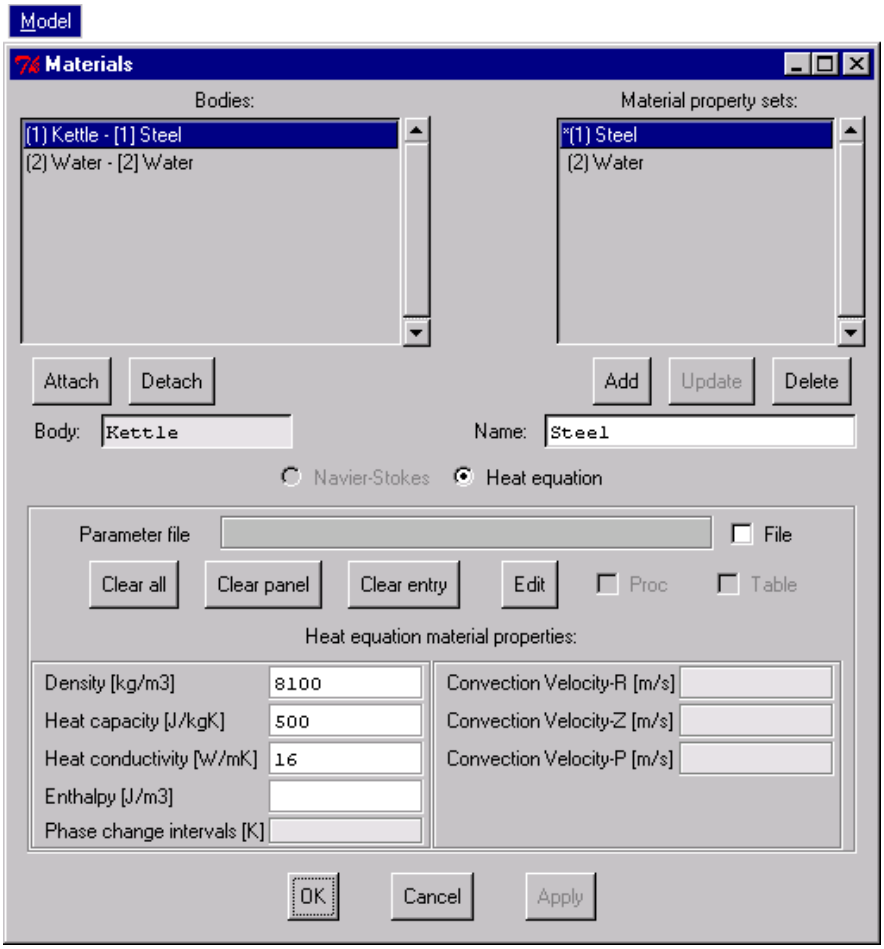


Figure 2.18: *Materials window*

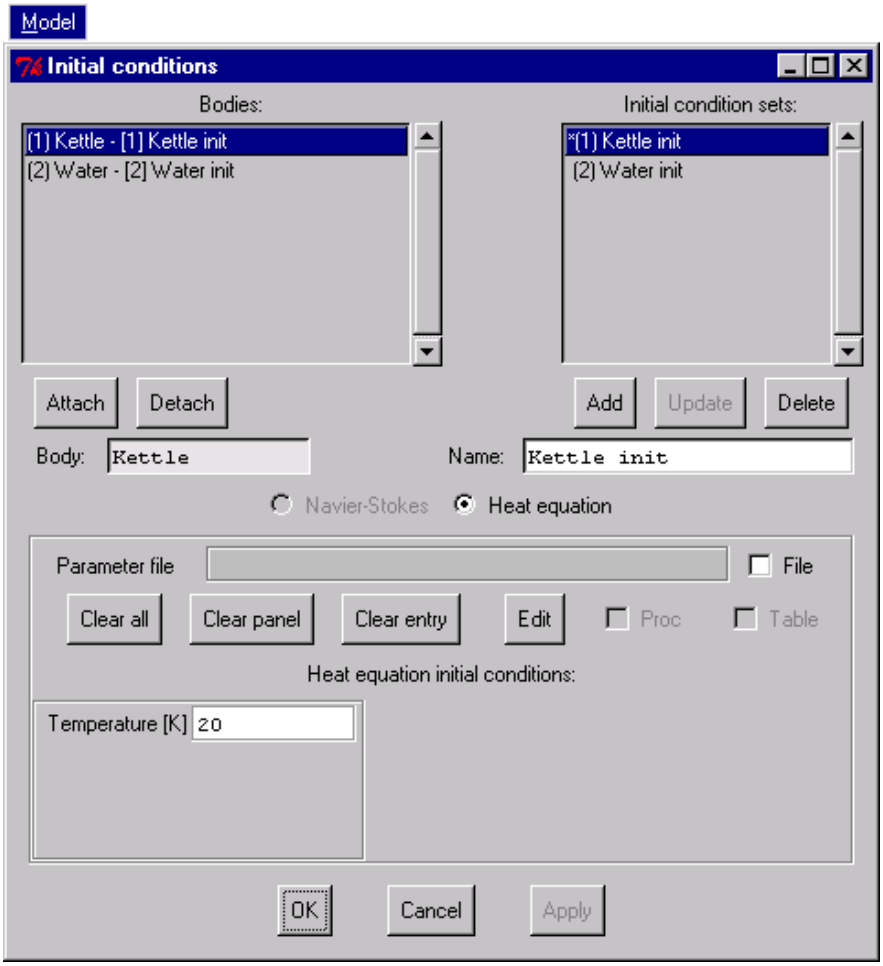


Figure 2.19: *Initial conditions window*

In all these windows parameter sets are attached to the bodies. The general workflow and the button properties were already described in the Equations window section.

Entry fields displayed on the window depend on the selected equation type. When a body is selected only those equation type radio buttons are active which are relevant for the body. And only those equation type buttons are visible which are relevant for the current problem. For example, if no flow equation on any of the bodies has been defined, the Navier-Stokes radio button would not be visible on these windows.

Initial conditions:

In Figure 2.19 Body1 with the initial condition set "Kettle init" has been selected. The Heat Equation radio button is shown selected.

Now the user clicks on the Body2:

1. The 'Water' named body is clicked on.

2. The 'Water' body row and the initial condition set 'Water init' become highlighted.

3. The Navier Stokes radio button becomes enabled, and when the user clicks it entry fields for the flow equation are displayed.

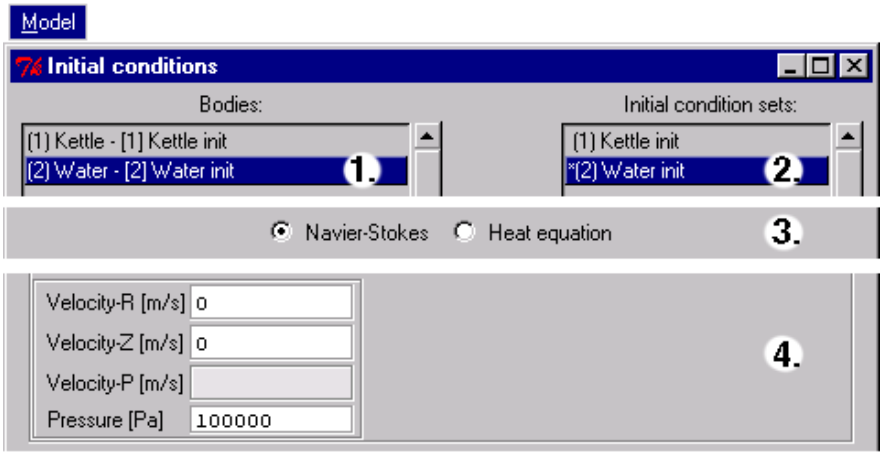


Figure 2.20: *Parameter selection example*

Bodies can be selected also in the Model window by double-clicking on the body area.

As a body is selected all those initial condition sets which can be attached to the body are marked with an asterisk (*). Any of the conditions marked with an asterisk matches the equation type of the currently selected body and consequently they can be attched to the body. In the example the 'Water init' condition set matches the 'Water' body.

Parameter file field:

An initial condition set can also contain a parameter file name. Name for this file is entered in the Parameter file field. This field can be activated by checking the File checkbox at the right side of the field. Unchecking the checkbox makes this field inactive and its value is not taken into acoount.

When ELMER Solver sees a parameter file name entry in the solver

input file, it reads input values for this parameter set from the given file. The structure of this file should be the same as in the standard solver input file.

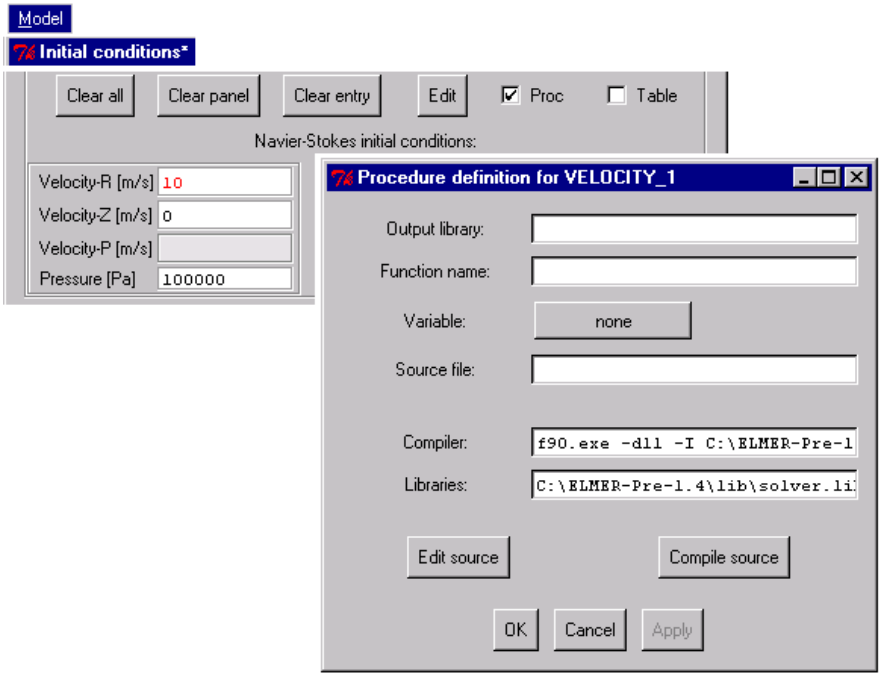
If the parameter file name does not contain any path information (i.e. when it is a simple file name), the file is searched using the Include path value given in the Model name and directories window. Otherwise the path given in the entry field is used when searching the file. (Note that the existence of the file is not checked when the data is entered in ELMER Front.)

If the initial condition set contains also other data fields, values in these fields will always overwrite corresponding values in the parameter file data.

Paramter files can be used also for body forces, material properties and boundary conditions.

Procedure entry:

- Proc checkbox: clicking this checkbox opens a separete window for procedure definition. This window is shown in Figure 2.11.4 and the fields in the window are the following:



- Output library: the library where the procedure being defined is

stored

- Function name: the call name for the procedure in the library

- Variable: the possible argument variable can be selected by pressing this button. This variable can be any of the variables solved in the model, any of the model coordinates and for a transient problem it can also be time.

Figure 2.21 shows the variables available for a transient 2D problem with heat and flow equations.

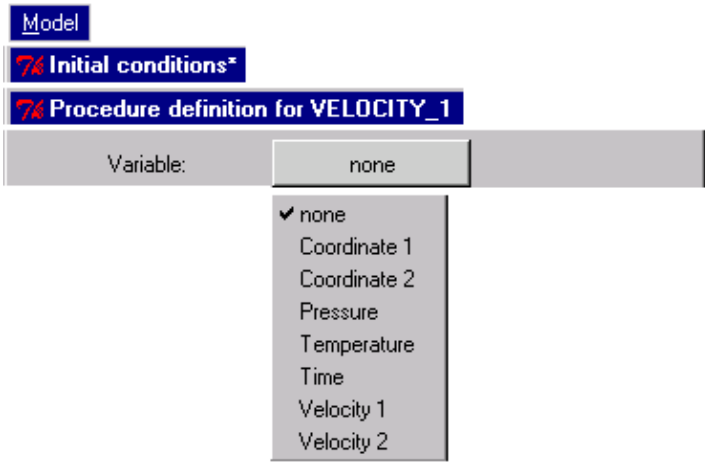


Figure 2.21: *Argument variable list*

- Source file: the source file for the procedure

- Libraries: possible external libraries needed by the compiler

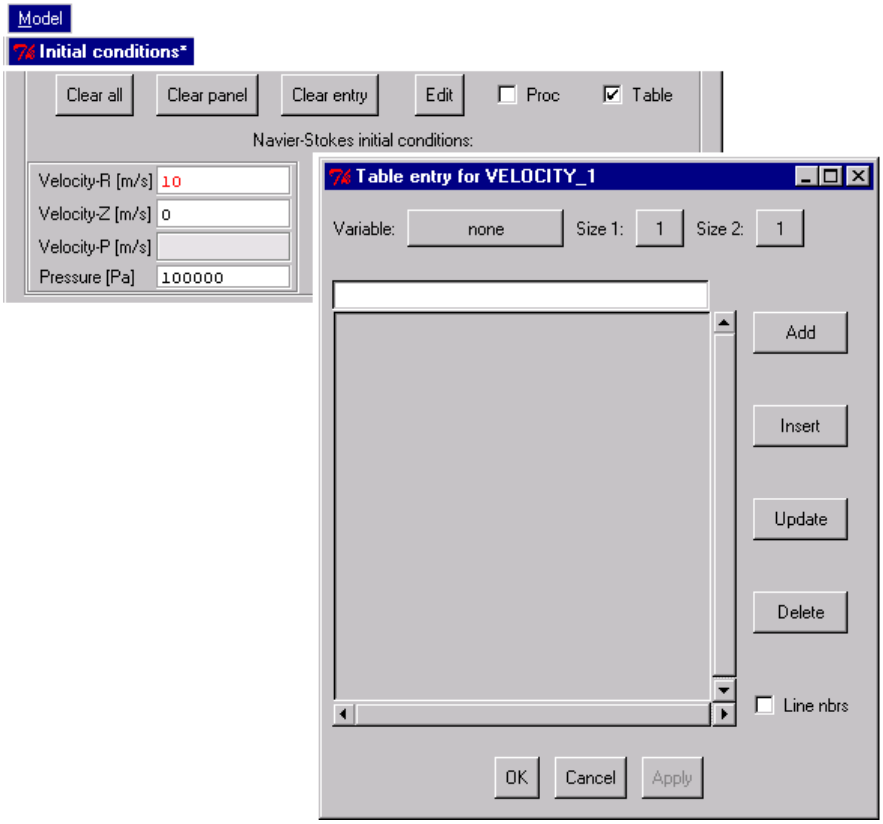
- Compiler: the compiler command script

- Edit source button: opens the procedure source file for editing.

- Compile source button: compiles the source file and append the procedure to the libary

Table entry:

- Table checkbox: clicking this checkbox opens a separete window for entering a data table. This window is shown in Figure 2.11.4 and the fields are the following:



- Variable: a possible argument variable for the table

- Size 1: if the parameter is not a scalar, this is the first dimension of the data

- Size 2: if the parameter is not a scalar, this is the second dimension of the data

For a scalar field (like density) these values are always one. Argument variables are always scalars.

- Entry field: the data in the table is entered using this field. First value should be always for the argument variable if it is defined. After that come the data values. The number of data values should be $Size1 \times Size2$ and these values should be separated by spaces

- Add: a new row is added at the end of the table

- Insert: a new row is added above the currently selected row

- Update: updates the currently selected row

- Delete: deletes the currently selected rows
- Line nbrs: when this checkbutton is checked a line number is displayed at left side of each row

2.11.5 Boundary conditions

The Boundary conditions window differs from the previous windows as boundary conditions are applied to boundaries and other elements defining the bodies, like edges and vertices. Thus, in addition to the bodies and boundary conditions listboxes, this window contains a third listbox, the boundaries listbox.

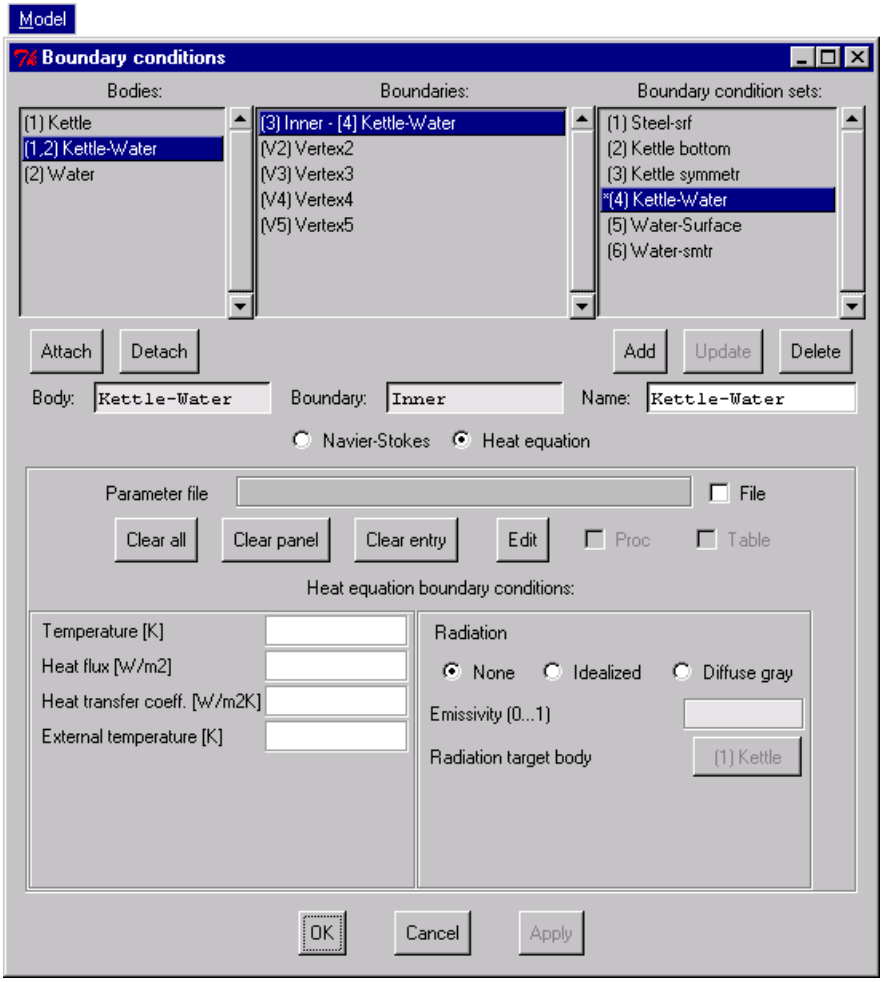


Figure 2.22: *Boundary conditions window*

Entry fields are grouped by the equation type also in this window. If the problem contains more than one equation type the radio buttons in the middle of the window are used to select the equation.

When a body is selected in the bodies listbox, all boundary elements (boundaries, edges and vertices) for the body are displayed in the boundaries listbox. An asterix is again used to mark the boundary conditions which can be attached to the boundaries of the selected body.

Selecting a body displays only outer boundaries for this body. These are boundaries which are not shared between bodies. Boundaries bewteen two bodies are called inner boundaries and they are grouped under body pairs. A body pair inherits equation types from both parent bodies and body pairs are listed just like normal bodies in the bodies listbox.

Boundary elements can be grouped into boundaries (faces in 3D and edges in 2D) and into subelements like edges (in 2D) and vertices. Normally boundary conditions are attached directly to boundaries, but this can create conflicts at common vertices (and at common edges in 3D) if Dirichlet-type conditions are applied. For example, if a temperature constraint has different values in two boundaries which have a common vertex, the final value at this vertex is not unique. In ELMER Solver boundary conditions are applied in the order they are created in ELMER Front. So, this offers a simple method to control these conflicting values. The last value will be used.

However, it is sometimes important to control constraints at edges and vertices more exactly, and for these cases the option to define boundary condtions at that specific level is needed. It should be also noted that it is not possible to attach a boundary level condition to edges or vertices and vice versa, because this conditions could contain values which are not applicable at these lower geometrical level. For example, it would not be meaningful to apply a flux constraint on a vertex.

Radiation boundaries

For a diffuse-gray radiation boundary, one has to define a radiation target body. For an outer boundary the target can be either the body itself or the 'Outside' environment. For an inner boundary, the target body must be one of the parent bodies. It is possible to apply a constraint defined at a boundary to boundaries in other bodies having similar equations, as long as the target body is the same. However, if a constraint is applied to multiple bodies, it is not any more possible

to change the target body, because this would create contradictions. The only way to change the target body in these cases is to detach attachements until the constraint is again attached to the boundaries of only one body or body pair.

2.12 Mesh menu

2.12.1 Background

ELMER Mesh supports triangular and quadrilateral 2D element meshes. Elements can be linear or parabolic and different bodies can have different mesh types. However, element types cannot be mixed within a body.

The structure of a triangular mesh is free. This means that element size can change freely over the body. On the other hand, quadrilateral meshes are always structured. This means that a quadrilateral mesh can be defined only for those bodies which are quadrilaterals, ie. for bodies which have four boundaries. These boundaries can be polylines which consit of multiple segments, but topologically they must define a quadrilateral geometry.

Triangular meshing is controlled by the mesh density values which are defined at body vertices. During the triangulation process these densities are interpolated over the body and these values control the mesh density in different parts of the body. The final structure of the mesh may depend on the actual meshing method, but the basic principle is the same.

When defining the mesh, it is possible to enter these density values for each vertex in the model. However, for convenience data entry is organized so that density values can be entered either at model, body, boundary or vertex level, whatever is suitable for the problem. Whenever a lower level value is not defined, it is inherited from the upper level. So, it is possible to enter just one model level density parameter and create a constant density mesh for the whole model.

In a structured mesh opposing boundaries are always devided into same number of mesh segments (mesh boundary elements). The structure is thus defined by two numbers which tell the number of elements per boundary pairs. It is possible to define a structured mesh for a body and a triangular mesh for a body which is adjacent to this body. However, in these cases the structured mesh defines automatically the

mesh density at the common boundary and the resulting triangular mesh may become quite artificial. So, mixing element types in one model is possible and in many cases quite useful, but it must be done carefully.

These concepts are highlighted in the example shown in Figure 2.24.

Labels

In the Model window normal boundaries are labeled by sequence numbers and vertices with the sequence number preceded by the letter V.

Mesh element types

ELMER Mesh2D supports the four mesh element types shown in Figure 2.23. `A` is a linear quadrilateral element, `B` is a parabolic quadrilateral element, `C` is a is linear triangular element and `D` is a parabolic triangular element.

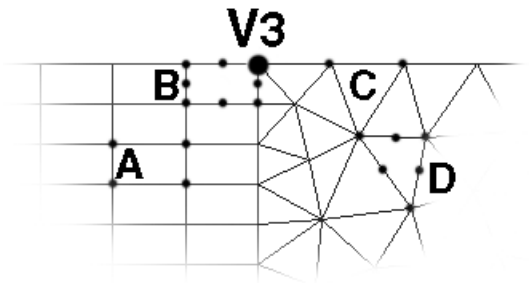


Figure 2.23: *Mesh element types*

2.12.2 The three mesh definition levels:

- The Model Level (L1)
- The Body Level (L2)
- The Boundary Level (L3) - including also edges and vertices

If no definitions has been made on (L2) and (L3) levels, the model level (L1) parameters are set as default to all meshes included in the model.

2.12.3 Mesh example

Figure 2.24 shows an example where each body of the model has a different mesh type. Body1 has a quadrialteral (structured) mesh and

Body2 has a triangular (free) mesh.

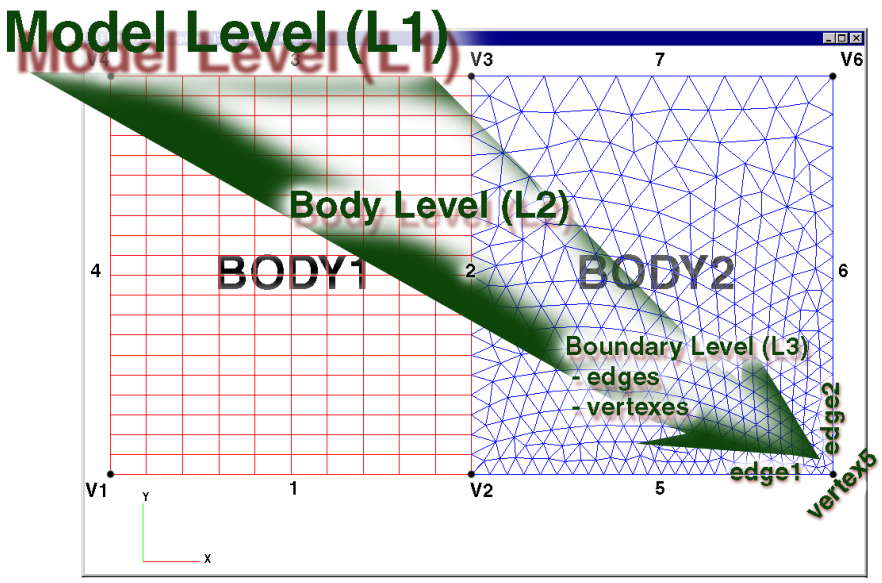


Figure 2.24: *Mesh example*

The arrow from the top left down to the right corner highlights the relation between different levels. The boundary level (L3) (edges and vertices) inherits the appropriate body level (L2) mesh density value if no definitions for them has been made. Similarily the body level inherits the model level (L1) density value.

BODY 1:

For Body1 a quadrilateral mesh has been defined. Edges 1 and 3 are devided into 10 segment and edges 2 and 4 into 20 segments and the total number of elements is 200.

BODY 2:

For Body2, the user has defined a triangular mesh. Vertex5 has its own mesh density value and the mesh is denser around this vertex.

If the user would not have defined any value for Vertex5, it would have inherited an average value derived from the density values at the edges 5 and 6.

2.12.4 Define mesh

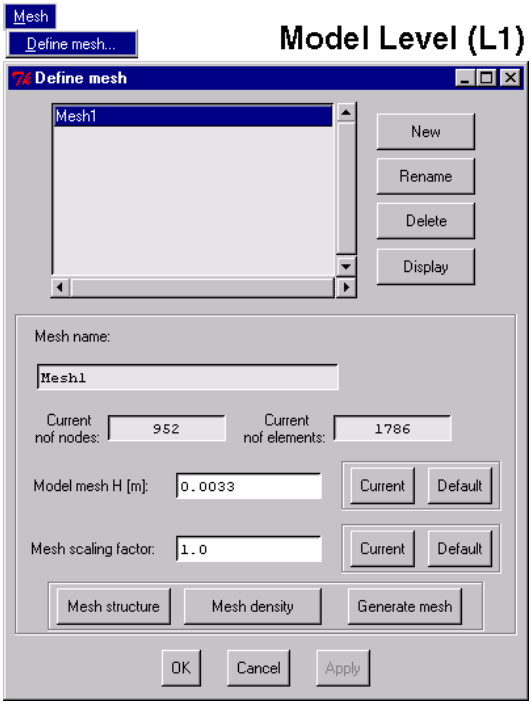


Figure 2.25: *Define mesh window*

ELMER Solver has a multimesh capability. This means that different equations can be solved using different meshes. If variable values from one equation are needed in other equations during the simulation, they are interpolated from one mesh to an other. This is completely transparent to the user, it is only necessary to define the mesh for each equation.

The Define mesh command in the Mesh menu opens a window where one can define all the meshes needed in the problem. These meshes must have unique names. These names are used also in the Solver settings window where the mesh for each equation is defined.

In the upper part of the window is a listbox where all currently defined meshes are listed. When a row in this listbox is selected, the corresponding mesh is made active and its parameter definition can be checked or updated and the mesh can be generated or regenerated using the Generate mesh button at the bottom of the window. When a new mesh is being defined, the user has to first press the New mesh button. This activates the mesh name field where one can now enter

a new unique mesh name. This new mesh is added to the mesh list when the mesh is generated.

The button and fields in the window are the following:

- New: a name for a new mesh can be entered
- Rename: a new name for the currently selected mesh can be entered
- Delete: the currently selected mesh is deleted
- Display: the currently selected mesh is displayed in the Model window
- Mesh name: the name for the selected mesh is displayed in this field and it is used for entering and renaming the mesh name as explained above
- Current nof nodes: the total number of nodes in the selected mesh
- Current nof elements: the total number of elements in the selected mesh
- MeshH value [m]: the default mesh element size for the whole model is defined with this value. Values defined at lower level will overwrite this value. Default value is derived from the average dimension of the model.
- Mesh scaling factor: all mesh density values given, derived or inherited in the model are finally multiplied with this factor. Default value is 1.0
- Current: this button sets the field value into the currently active value
- Default: this button sets the field value into the default value
- Mesh structure: this button opens a window for entering body level mesh definitions (L2)
- Mesh density: this button opens a window for entering boundary level mesh definitions (L3)
- Generate mesh: this button will generate the mesh using the current mesh definition values. Note that a new mesh is not added to the list of meshes until this button is pressed

2.12.5 Mesh structure

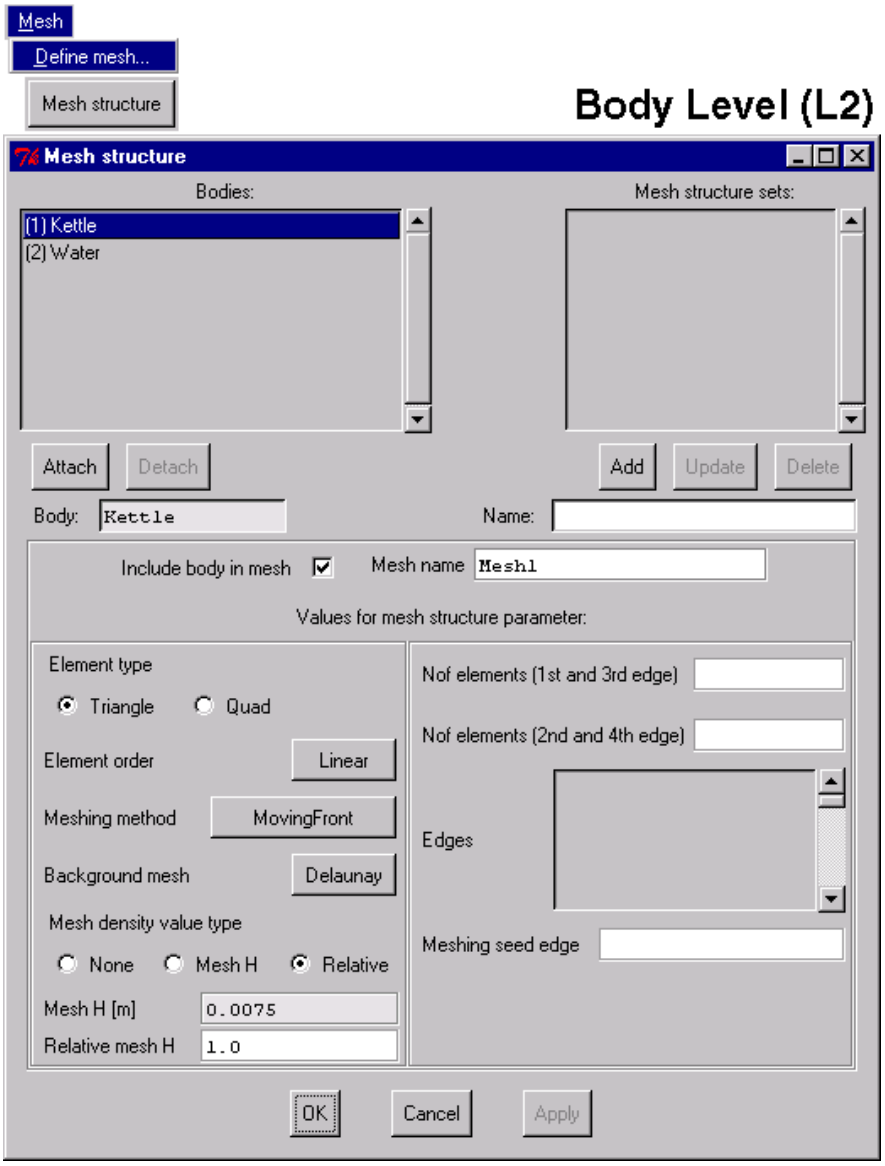
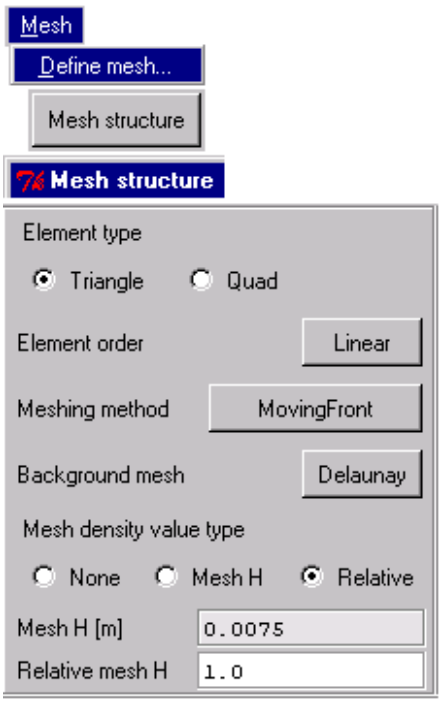


Figure 2.26: *Mesh structure window*

The Mesh structure window opens by clicking on the Mesh structure button in the Define mesh window. Parameters entered in this window define the mesh at body level (L2).

On top left is the list of all bodies and the mesh structure sets attached to them. The name of the currently active mesh is displayed in the

Mesh name field. This field is not editable. The selected body can be included or excluded from the mesh by using the Include body in mesh checkbox.



The mesh element type can be selected using the Triangle and Quad radio buttons. If the quadrilateral mesh type is selected the mesh size parameter entries on the right side are activated as shown in Figure 2.12.5.

- Element order: Linear setting creates three nodes triangular and four nodes quarilateral elements. Parabolic setting creates six nodes triangular and eight nodes quadrilateral elements

- Meshing method: Basic meshing strtegies for free (triangular) meshes are moving front methods and the Voronoi vertex method. Default value is the standard MovingFront method. Other moving front options are SSSFMovingFront (single seed, single front) and SSMFMovingFront (single seed, multi front) methods. These methods need a starting edge as seed. The boundary id for this seed can be entered in the Meshing seed edge field.

- Background mesh: Free (triangular) meshing starts from an automatically created background mesh. Delaunay is the default method. Other options are regular background grids: Grid, SparseGrid and DenseGrid. A dense background grid creates better quality meshes, but it is also a slower method. SparseGrid is faster, but the mesh quality may be lower.

Options for these buttons are displayed in Figure 2.12.5.

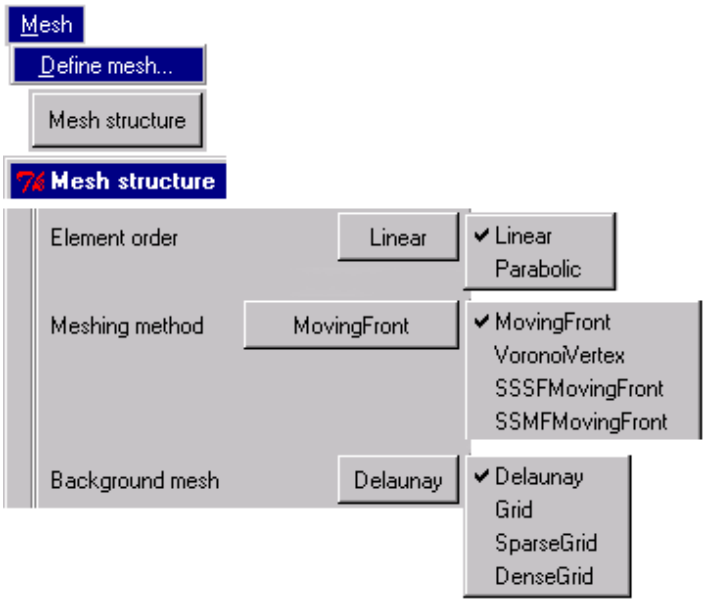
Mesh density value type radio buttons:

- None: deactivates values on this level

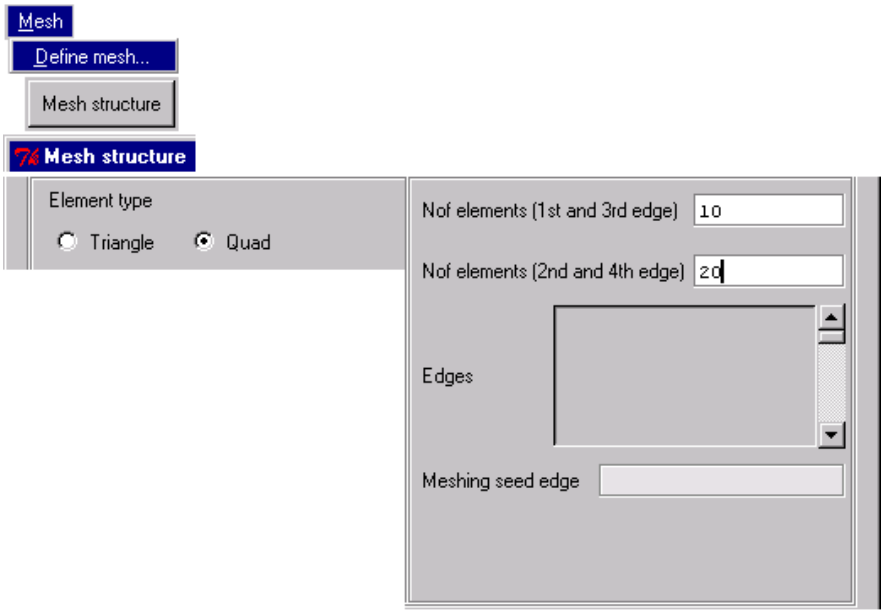
- Mesh H: activates the Mesh H field to be used for entering an absolute mesh density value in this level

- Relative: activates the Relative mesh H field to be used for entering

a scaling factor for mesh density in this level. The actual mesh density value for this level is the product of the scaling factor and the model level density value.



When the Quad mesh element type is selected, the fields at right are activated for input.



- Nof elements (1st and 3rd edge): the number of elements (segements) for the first and third edges in the Edges listbox

- Nof elements (2nd and 4th edge): the number of elements (segements) for the second and fourth edges in the Edges listbox

2.12.6 Mesh Density

The Mesh density window opens by pressing the Mesh density button in the Define Mesh window. These definitions affect the mesh at the boundary level (L3). Mesh density values can be set at edges or at vertices. However, a parameter set defined for an edge cannot be applied to a vertex and vice versa. They have different 'masks'. This is mainly because a 'nof elements'-type definition does not have any meaning at vertex level.

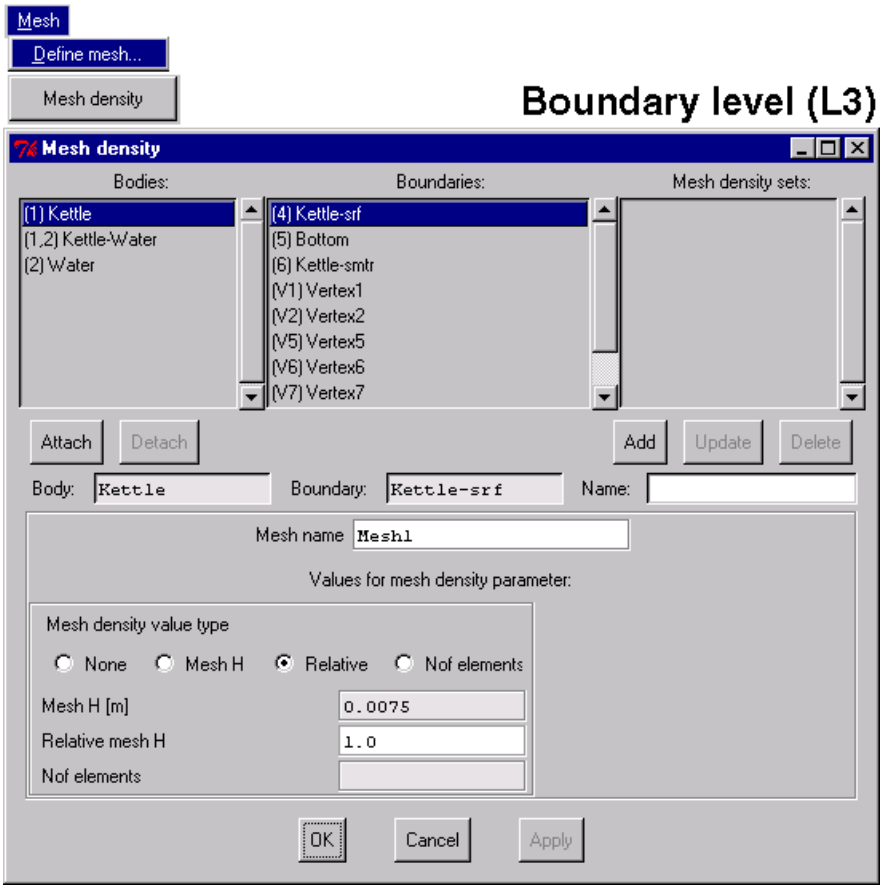


Figure 2.27: *Mesh density window*

Mesh density value type radio buttons are used to select between three different input options:

- None: deactivates values on this level
- Mesh H: activates the Mesh H field to be used for enetering an absolute mesh density value in this level
- Relative: activates the Relative mesh H field to be used for enetering a scaling factor for mesh density in this level. The actual mesh density value for this level is the product of the scaling factor and the body (or model) level density value.
- Nof elements: the fixed number of elements (segments) at the edge

Mesh log file

When the mesh is generated a log file is created (if this process output setting is activated). An example of this file is shown in Figure 2.28. This and other logfiles may be viewed using the Process table command in the Run menu.

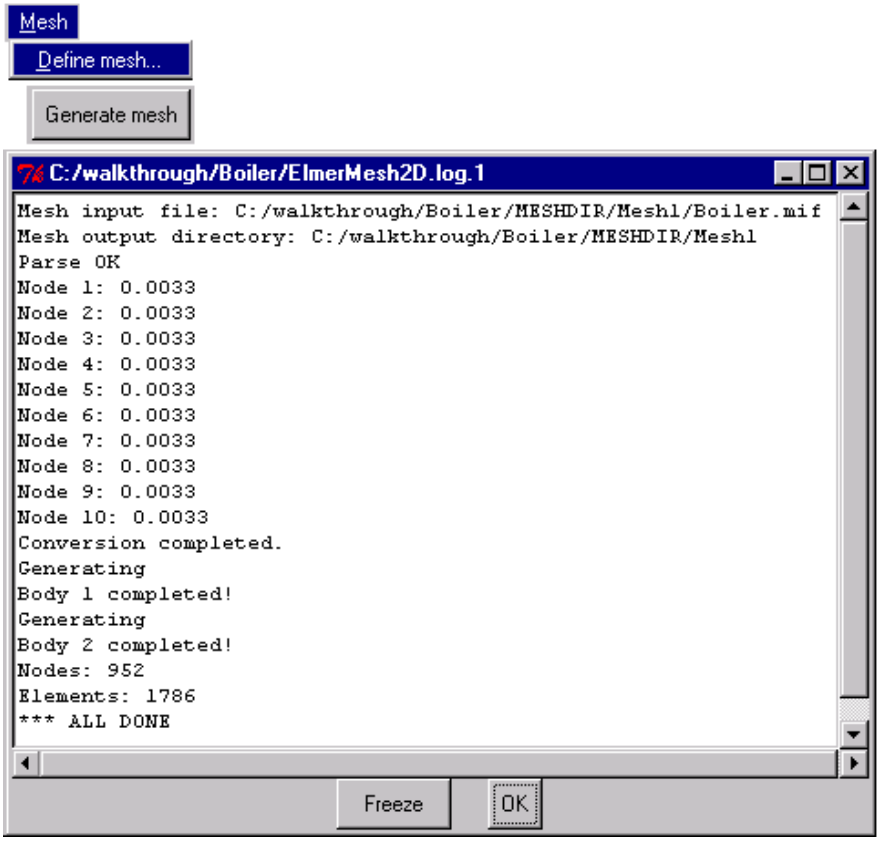


Figure 2.28: *Mesh logfile example*

2.13 Solver menu

2.13.1 Solver settings

The Solver settings window is shown in Figure 2.29. Settings for the Navier-Stokes equation are displayed. In this example all fields have their default values.

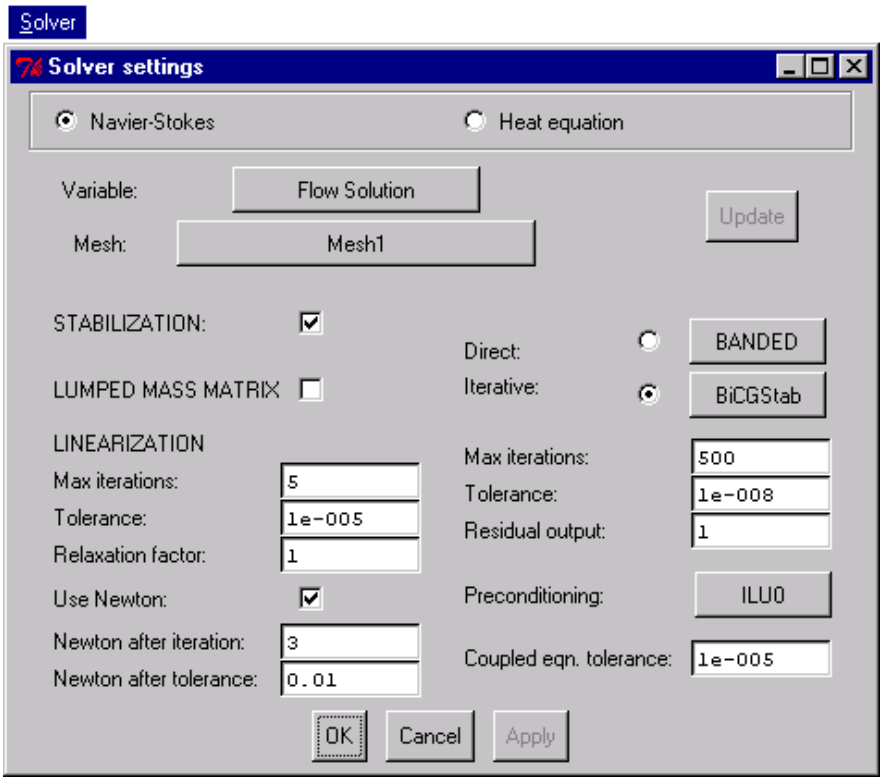


Figure 2.29: *Solver settings window*

Each equation system in the model is solved by a separate solver. These solvers can selected using the radio buttons on the top of the window. The advection-diffusion equation differs from other equations because it can solved for multiple variables. Each of these variables is solved by a separate solver. These solvers can be selected by pressing the Variable button and selecting the corresponding variable from the list.

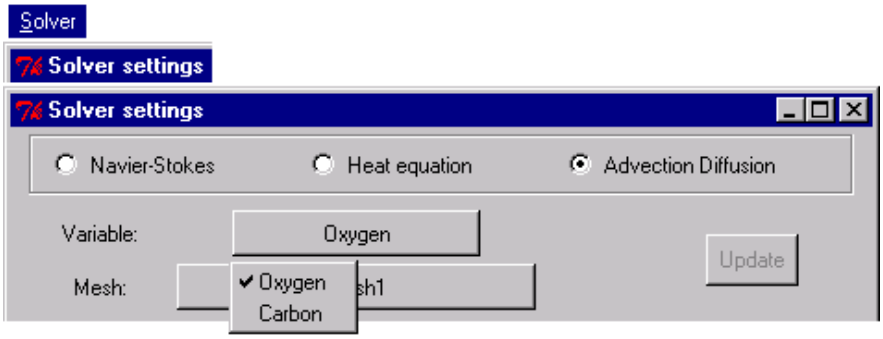


Figure 2.30: *Advection-diffusion solver example*

When all data fields for the solver are entered or modified, the settings for the solver are stored by the Update button. All solvers have initial values which are adequate for standard cases. However, it may be that the solution for an equation does not converge or converges slowly, and in these cases it is a good idea to test with different parameter values.

Fields in the window are grouped so that fields related to the general form and linearization methods of the equation are in the left side of the window. Fields related to the linear system and its solving methods are on the right side of the window.

STABILIZATION: when selected, solver uses stabilization. It is a recommended option for the Navier-Stokes equation

LUMPED MASS MATRIX: the matrix corresponding to the mass matrix is lumped in the equation

LINEARIZATION:

- Max iterations: linearization is based on subiterations and this number sets the maximum number for these iterations

- Tolerance: if the linearization error norm is smaller than this number, linearization iteration is stopped

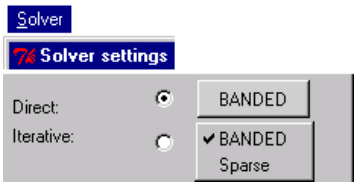
- Relaxation factor: default value is 1.0. Linearization could be speeded up in some cases using somewhat larger value. On the other hand, sometimes a value less than unity is needed to get any solution

Use Newton: when selected Newton's linearization method is used. It is faster than the default method (Picard), but is useful only when the actual solution is close enough to the exact one. The following fields control when Newton's method is started:

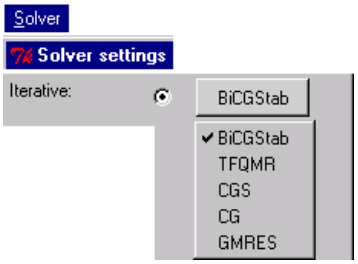
- Newton after iteration

- Newton after tolerance: Newton's method is not started until at least one of these criteria is fulfilled

- Direct: this radio button selects a direct solving method. For relatively small problems a direct solver is normally faster. For any larger problem it is usually better to use an iterative solver. The version for the direct solver can be selected pressing the button at the right side of the radio button.



- Iterative: this radio button selects an iterative solving method. Current options are the following:



- BiCGStab: BiConjugate Gradient Stabilized
- TFQMR: Transpose Free Quasi-Minimal Residual
- CGS: Conjugate Gradient Squared
- CG: Conjugate Gradient
- GMRES: Generalized Minimal Residual

- Max iterations: the maximum number of iterations for the iterative solver

- Tolerance: an error norm threshold. If the error norm is smaller than this number, iteration is stopped

- Residual output: an integer value; linear system residual is output for each Nth iteration (zero value disables the output)

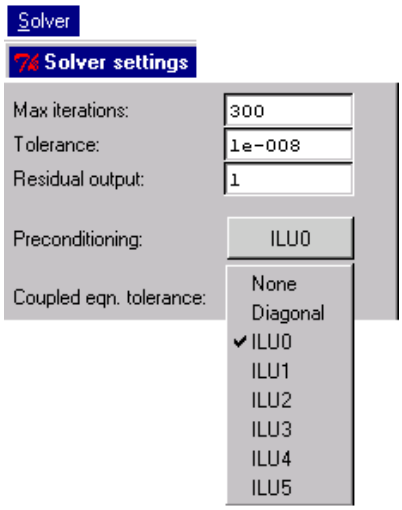
- Preconditioning: using preconditioning normally speeds up the iteration, however, resulting in some overhead. Options here are:

- None: no preconditioning

• Diagonal: Diagonal preconditioning matrix. Fast construction, but not very effective

• ILU: Different level of Incomplete LU decompositions. ILU0 is the recommended choice

• Coupled eq. tolerance (or Steady state tolerance): for transient problems this field sets the tolerance for stopping the iteration when multiple equations are coupled to each other. For steady state problems it sets the tolerance for stopping the steady state iteration



2.13.2 Processor settings

This command opens a window for setting the number of processors reserved for the problem. This parameter has meaning only in a parallel processing environment.

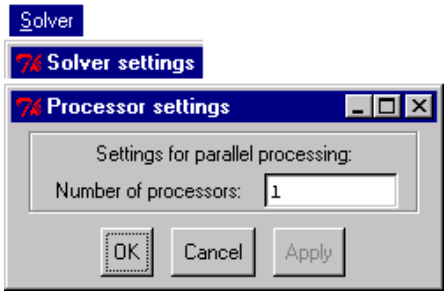


Figure 2.31: *Processor settings window*

2.14 Run menu

The Run menu commands are used to start other ELMER modules. Before a module can be started, all the necessary definitions and settings should be completed. If this not the case, an error message box is displayed. If the missing data is essential for the module, the process cannot continue.

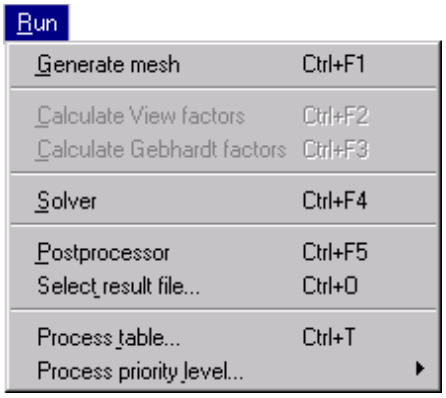


Figure 2.32: *Run menu*

2.14.1 Generate mesh:

Generate mesh command starts the ELMER mesh generator. it creates a browsable log file.

2.14.2 Calculate View factors:

Calculate View factors command starts ELMER View factors program. It stores the result data in the View factors file defined in the Problem / Datafiles window. **Note: for large meshes this can be a time consuming process.** It creates a browsable logfile.

2.14.3 Calculate Gebhardt factors:

Calculate Gebhart factors command starts ELMER Gebhardt factors program. It stores the data in the Gebhardt factors file defined in the Problem / Datafiles window. **Note: for large meshes this can be a time consuming process.** It creates a browsable logfile.

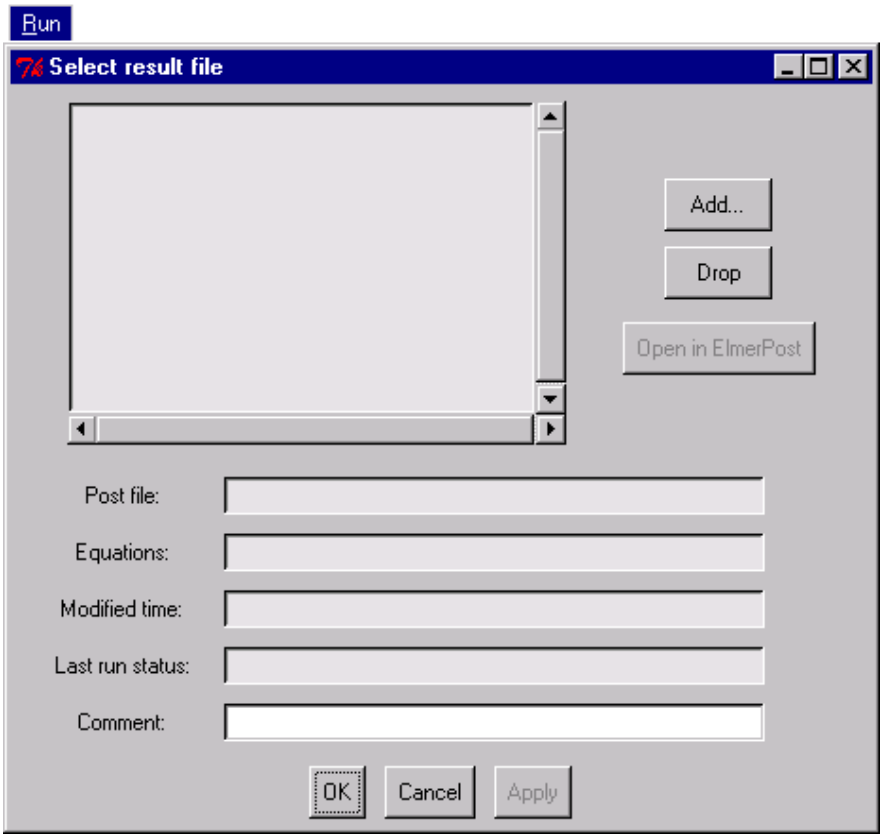
2.14.4 Solver:

This command starts ELMER Solver. It creates a browsable logfile.

2.14.5 Postprocessor:

Postprocessor command starts ELMER Post in a separate window.

2.14.6 Select result file



This window is needed when multiple meshes are in use. Because all result files are stored in the same directory where the mesh for an equation is stored, it is necessary to specify the result for postprocessing in a multimesh case.

All postprocessor output files produced during the session are listed in the listbox. Any of the files may be selected as the input for the postprocessor.

- Add opens a file browser for adding new files (created in other sessions) to the list

- Drop removes a file from this list. The file itself is not affected
- Open in ELMER Post opens the selected file in ELMER Post

The following fields are non-editable information fields:

- Post file: the full path for the postprocessor file
- Equations: the equations solved in the result file
- Modified time: the last update time for the result file
- Last run status: status of the process which updated last the output file
- Comment: this comment field is only for this session and it is not saved in the model file

2.14.7 Process table

When other ELEMR modules are started in ELMER Front, information on these processes is collected into a process table. This table can be opened using the Process table command or the Process button in the main window. This table displays information on the current status of a process and the user can also control the process using the command buttons in the process table window.

Process table columns:

- Process: the name of the process
- Nbr: an internal id number for the process
- PID: the operation system process id (if available)
- Prior: the process priority level
- State: the current process state. Value are:

  ALL DONE = process ended succesfully

  FAILED = process failed during the processing

  KILLED = process was ended by the user

  SUSPENDED = process is suspended and is waiting to be activated

  RUNNING = an active and currently running process
- Start / End: process start and end times (wall clock times). For a

running solver process the End field displays the number of currently processed timesteps

- Duration: total run time
- W: the rightmost (W) column is marked with an asterisk if the process created any error messages or warnings.

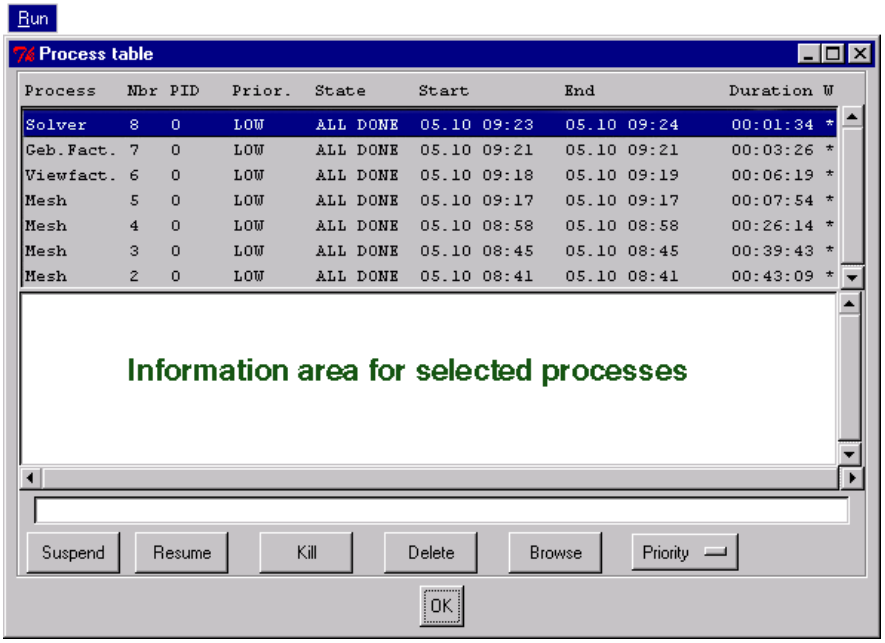


Figure 2.33: *Process table*

Processes are listed in the order they were started. The most recent process is on the top of the list. All fields are info fields. Below the process lists is an area where information on the currently selected process in displayed. The actual contents depends on the process type. Possible error and warning messages are also displayed in this area.

Process table buttons:

- Suspend: the selected process is suspended (paused)
- Resume: resumes (continues) the process
- Kill: stops the process (after confirmation). Note: the process name is not erased from the list
- Delete: kills the process and removes the process from the list (after confirmation)
- Browse: opens the log file (if available) for the selected process

- Priority: the process priority level can be set using this button. Options are Favor GUI or Favor process. These are descriptive values and the actual priority level transmitted for the operation system is platform specific. Favor GUI is the default and recommended value. Giving more priority for running processes can make the user interface almost unresponsive.

2.15 Window menu

Commands in the Window menu can be used to control currently active ELMER Front windows:

- Close all: closes all windows (except the main window and the Model window)

- Close unmodified: closes all windows where data has not been modified (except the main window and the Model window)

- Window list: displays a list of currently open windows. This list can be opened as a separate window by clicking on the dotted line as shown in Figure 2.34. Clicking a window name in this list activates the window.

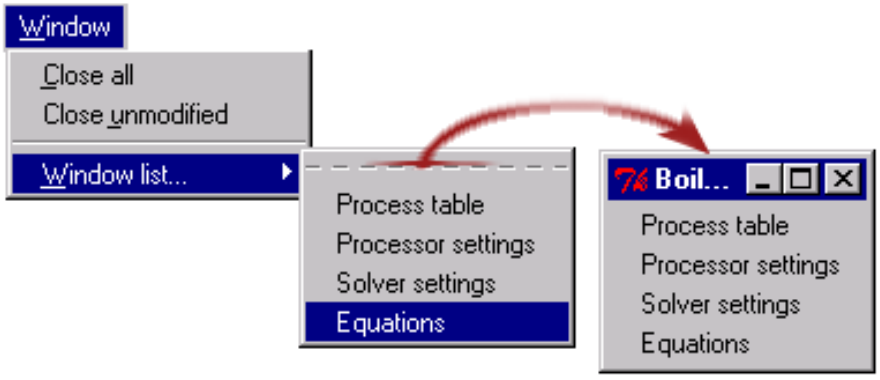


Figure 2.34: *Window list example*

2.16 Help menu

Figure 2.35: *Help menu*

2.16.1 Graphics info

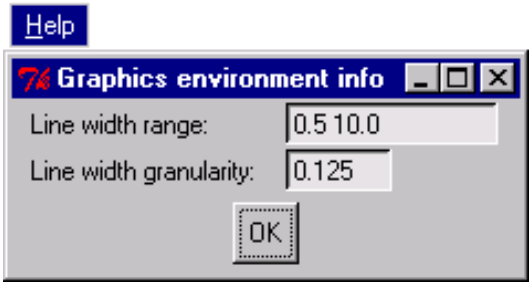


Figure 2.36: *Graphics info window*

The Graphics info window displays some OpenGL related graphics hardware information. These are non-editable information fields.

2.16.2 System info

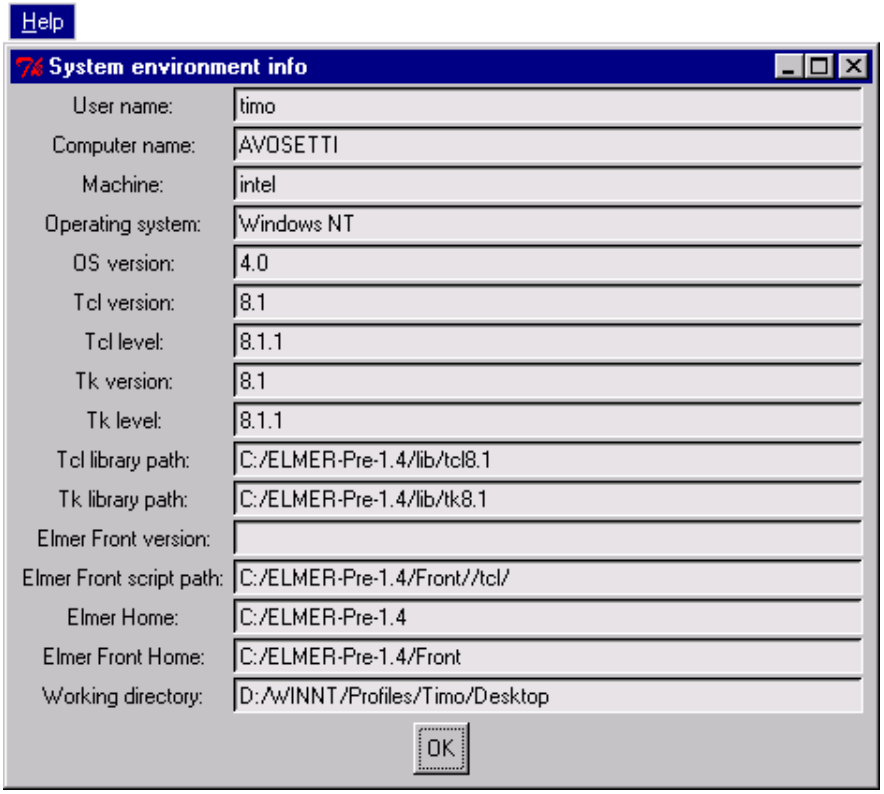
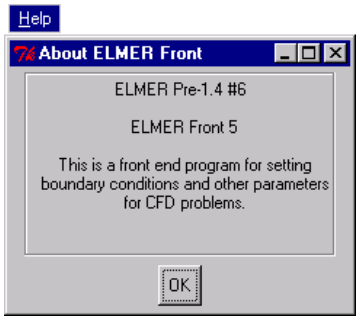


Figure 2.37: *System info window*

The System info window displays some information on operation system and on the external libraries which are needed for ELMER. These are non-editable information fields.

2.16.3 About

The About window displays information on the current ELMER version.



2.17 Model window toolbar

Figure 2.38: *Model window toolbar*

The following toolbar buttons in the ELMER Front main window are used to control how the model geometry is displayed in the Model window:

- Arrows on the left side are used for moving, rotating and zooming the model
- Rotate X, Y, Z checkboxes: model rotation axis is locked to the selected axis.
- Draw bodies: a CAD geometry (2D only) is drawn in solid format. For a mesh geometry this drawing mode displays the volume mesh by drawing the element edges
- Draw surfaces: mesh geometry is drawn using filled boundary elements (only for 3D models)
- Draw edges: CAD geometry is drawn using edges, mesh geometry is drawn using boundary element edges
- CAD geometry: activates/deactivates the display of the CAD geometry (if CAD geometry is available)
- Mesh geometry: activates/deactivates the display of the mesh geometry (if mesh is geometry available)

- Auto load mesh: when selected a mesh is automatically loaded whenever a model file is loaded. If the mesh is very large it may be useful to deactivate the autoload behaviour

- Select bodies: the bodies to be displayed are can be selected here

- Display: redisplays the Model window on the desktop if it has been closed or minimized

- Reset: resets the the graphics displyed in the Model window to its initial state

2.18 Model Window

The Model window is opened automatically when a geometry input file is read into ELMER Front. This window is used to display the model geometry and to offer a flexible interface for the model. When model objects are selected in the Model window, these selections are reflected in the object listboxes in data entry panels. On the other hand, selections made in these panels are highlighted in the Model window.

In the following example a boundary has been selected in the Model Window by double-clicking on it. As are result, the correponding boundary, the parent body and the attached boundary condition set rows are displayed as selected in the Boundary conditions window.

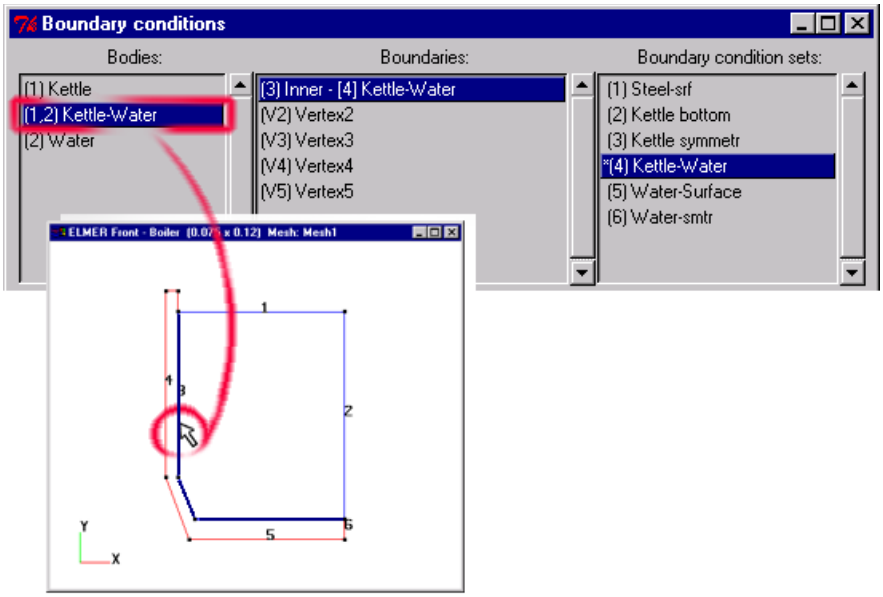


Figure 2.39: *Model window example*

The model can be manipulated in the window either by dragging it with the mouse or using the arrow buttons in the main window. Dragging with the left mouse button moves the model in the window. Dragging with the right mouse button rotates the model. The model can be scaled (zoomed) with the middle mouse button (or pressing left and right buttons simultaneously when using a 2-button mouse).

Bodies are displayed using different colors. These colors can be defined using the the Bodies command in the Edit menu.

All body elements (inner and outer boundaries, edges and vertices) are uniquely numbered. Same numbers are also used to identify these elements in the data entry windows. The Labels command in the Display menu can be used to switch theses labels on and off in the Model window.

2.19 Control toolbar

Most buttons in this toolbar have an equivalent command in menus.

- Save model: save model file

- Mesh: opens the Define Mesh window
- Load/Unload mesh: a two state button for loading and unloading the mesh
- Solve: starts the solver
- Results: starts the postprocessor
- Break: interrupts current ELMER Front processing
- Clear: deletes all current messages in the message area
- Process: opens the process table
- Info: opens the model info window

2.20 Message area

The window area in the middle of the main window is used to display information of various session events. This is a cumulative list of warnings, error messages and other session related incidents. This list can be cleared using the Clear button in the main window.

2.21 Status area

Fields at the bottom of the main window display information on the current status of the model. Red colored fields are meant to warn the user that the model may be incompletely defined concerning the parameters displayed in the status field. However, it is up to the user to decide if this really is the situation.

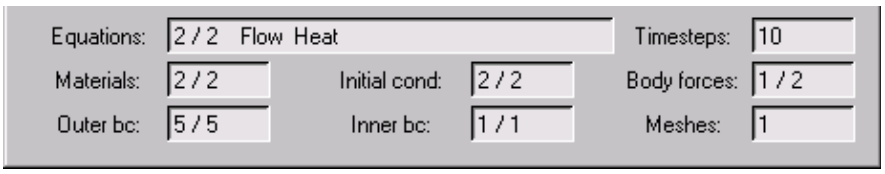


Figure 2.40: *Status area*

- Equations: nof bodies having an equation / total nof bodies in the model and names for the defined equations
- Timesteps: total nof timestesp defined for the problem. In a steady state case the maximum nof iterations
- Materials: nof bodies having a material definition / total nof bodies
- Initial cond: nof bodies having an initial condition / total nof bodies
- Body forces: nof bodies having a body force definition / total nof bodies
- Outer bc: nof outer boundaries having a boundary condition / total nof outer boundries
- Inner bc: nof inner boundaries having a boundary condition / total nof inner boundries
- Meshes: nof currently active meshes

Chapter 3

ELMER Geometry File format

3.1 Introduction

This appendix describes the ELMER Geometry File format. This geometry file is for implementing basic geometries which do not actually need the use of a full featured CAD program. The geometry primitives which can be defined in this file are limited to polylines and circular arcs, but in practice it is possible to define quite complicated geometries with these primitives.

The data in the file is devided into sections and these sections must come in a specific order in the input. However, some of the sections are optional. Each section starts with a keyword and ends with the keyword `End`. Data items within a section also start with a specific keyword.

Below is the list of the section keywords in the order they must be in the input. The letter N after the section name indicates that multiple section entries are allowed, but each section entry must be identifed by a unique id number. For example, there can be multiple edge sections in the input, but they must all be after the vertices and before the edge loops and bodies.

`Header`

`Vertices`

`Edge` N (optional)

`Edge Loop` N (optional)

`Body` N

This order means that the geometry is defined hierarchically. Lower lever elements must be given before the higer level elements which refer to them. First all vertices are defined, then the edges using theses vertices, then edge loops and finally the bodies using these edge loops.

However, it is possible to define a body using only edges or vertices. For example, if the geometry consists of one rectangular body, the four corner vertices are enough to define the geometry. On the other hand, if the geometry consists of multiple bodies which have common boundaries, it would be more effective to first define boundary edges which define the bodies, because same edges could be used in multiple bodies. Edge loops would be useful if some bodies are enclosed within other bodies, because then same edge loops could be used in multiple bodies.

Data values can be either integers, reals (decimal numbers) or strings. Corresponding data type identifiers are `Integer`, `Real` and `String`. Normally it is not necesary to specify the type of the data, because it can be concluded from the context,

It is not either necessary to specify the size of the data, because data values are read till the next keyword and this way the size can be concluded from the data itself. The size can be given using the `Size` keyword and one or two integers depending if data forms an array or a table.

If data forms a table, like the vertex points in the Vertices section, then the data type and size specifiers are always needed. The size keyword must be before the type keyword.

Accordingly, for example a list of edges could be given simply:

```
Edges 1 2 3 4
```

or

```
Edges
  Size 4
      1 2 3 4
```

or in the fully sepcified form

```
Edges
  Size 4
    Integer
      1 2 3 4
```

A string value could be given like:

```
Name "First object"
```

or

```
Name
  String
    "First object"
```

The layout of the input data is free as long as numeric values are separated with at least one space. Of course, using linefeeds and indenting makes data easier to check and read for users.

On each line, characters after '!' or '#' are interpreted as comments.

3.2 The format

In the following format description one of the letters (`i, r, s`) after a keyword means that some data value is needed in that place and the type of this data should be integer, real or string.

A string value must be given in quotes if it contains spaces or if it is given without a preceiding 'String' data type specifier. Otherwise quotes are not necessary, but it is safer to always quote the strings. Quotes within quotes are not allowed.

All coordinate walues and other geometrical input parameters are by default in meters. However, it is possible to define an other input unit using the Unit keyword in the Header section. For example, if the user wants to enter input data in millimeters, the input unit should be set to 0.001. All geometry related values (coordinates, circle radius etc.) in the input file would be multiplied with this factor. In the model the geometry data is always stored as meters.

3.2.1 The `Header` section

```
Header
  Model Name s          ! an optional string
  Model Directory s     ! an optional string
  Dimension i           ! currently only value 2 is accepted
```

```
  Unit r                ! input unit for all the geometry related data
                        ! (coordinates, circle radius, center etc.)
                        ! default is 1.0
                        ! 0.001 <--> input data is in millimeters
                        ! 1000  <--> input data is in kilometers etc.
End
```

3.2.2 The `Vertices` section

```
Vertices
  Points                ! keyword for the point table, must be given

    Variable Index      ! an optional keyword telling that the first
                        ! value  in each data row is and index (id)
                        ! for the vertex

    Size i1 i2          ! table size. If two integer values are given,
                        ! then i1 tells the number rows and i2 tells the
                        ! number of coordinate values per row. If only one
                        ! integer value is given, it tells the number
                        ! coordinate values per row. In this case the table
                        ! must be ended with the End keyword. See the
                        ! example file below

       Real             ! keyword for the data type in the table.

       (i1) r11 r12 ... ! coordinate values. Integers i1, i2, ...
       (i2) r21 r22 ... ! are given only if the keyword Variable Index is
                        ! given for an indexing argument variable (see above)

    (End)               ! this keyword which is needed when only one size
                        ! parameter is given.
End
```

Note: Vertices are always given as a table of points. Vertex coordinates can be either 2D or 3D coordinates. The number of coordinates must always be given (using the `Size` keyword). If an index variable is given in the beginning of each table row, this value is used as the vertex id. Index values should be unique and the index variable (an argument variable in general) is not included in the size parameter. The number of values per data row should refer only to the actual data values.

If no index variable is used, vertices are implicitely numbered using integers starting from one.

3.2.3 The `Edge` section

```
Edge i                  ! the unique id number i must be given
   Name s               ! optional, a string

   Geometry s           ! optional, a string, values are: Linear, Circle

   Center r             ! optional, two or three real coordinate values
                        ! defining the center of a circle.

   Radius r             ! optional, a real value for the circle radius

   Vertices i           ! vertex indices defining either a line
                        ! (can be a polyline) or an circular arc
End
```

Note: a full circle can be defined by given the center and the radius. A circular arc can be defined by the center and two end vertices or by three vertices.

3.2.4 The `Edge Loop` section

```
Edge Loop i             ! the unique id number i must be given

   Edges i              ! the edge ids (in any order) defining the loop
End
```

Note: Although the order of edge ids in the input is not important, these edges should always create a closed loop which does not interect with itself or with any other edges.

3.2.5 The `Body` section

```
Body i                   ! the unique id number i must be given

   Name s                ! optional, a string
```

```
   Color i                ! optional, four integers 0...255 (RGB values
                          ! and an alpha value)

   Vertices i| Polygon i  ! the vertex indices defining the body

   Edges i                ! the edge ids (in any order) defining the body

   Edge Loops i           ! the loop ids defining the body. The outermost loop
                          ! must always be given first, after that the inner
                          ! loops (if any) in any order
End
```

Note: The keywords Vertices and Polygon are equivalent. Again, the order of edge ids is not important, but these edges should create a well defined edge loop.

3.3 An example file

The following example geometry is taken from the `Boiler.egf` file which can be found in the directory `samples/cad_files` in the ELMER distribution. However, some clarifying comments are added here. This is also the same geometry which is used as an example in the ELMER User Guide. It defines two bodies, a kettle and the water in the kettle. Only the left half of this symmetric geometry is defined here.

```
Header
  Model Name "Boiler" ! model name is optional, it can be defined later
  Dimension 2         ! dimension must be given. Currently always 2
End                   ! each section ends with the keyword End

Vertices
  Points              ! a table of points starts
    Variable Index    ! an 'argument' variable is used to index the points
    Size 2            ! points as 2D coordinates
    Real              ! type must be given for a table
     1   0.000 0.000  ! note: one argument value (vertex id) and a 2D point
     2   0.000 0.010
     3  -0.063 0.010
     4  -0.070 0.030
     5  -0.070 0.110
```

```
     6  -0.070 0.120
     7  -0.075 0.120
     8  -0.075 0.030
     9  -0.065 0.000
     10  0.000 0.110
   End                  ! an End is needed to indicate the end of the table
End                     ! this is the section End

Edge 1                  ! edge number one
 Name "Surface"         ! water surface
 Vertices 5 10          ! 5 and 10 are indices in the vertex table
End

Edge 2
 Name "Water-smtr"      ! the water boundary at the symmetry line
 Vertices 10 2
End

Edge 3
 Name "Inner"           ! the inner boundary between the kettle and water
 Vertices 2 3 4 5       ! this is a polyline edge.
End

Edge 4
 Name "Kettle-srf"      ! the outer boundary of the kettle, excluding
                        ! the bottom
 Vertices 5 6 7 8 9     ! an other polyline edge
End

Edge 5
 Name "Bottom"          ! the bottom of the kettle
 Vertices 9 1
End

Edge 6
 Name "Kettle-smtr"     ! the kettle boundary at the symmetry line
 Vertices 1 2
End


Body 1                 ! body number one is the kettle
  Name "Kettle"
  Color 255 0 0 255    ! color by RGB codes and an alpha value (0...255)
```

```
  Edges 3 4 5 6        ! defined by four edges (3 and 4 are polylines!)
End

Body 2                 ! body number two is the water in the kettle
  Name "Water"
  Color 0 0 255 255
  Edges 1 2 3          ! three is the minimal number of edges in 2D!
End
```

Chapter 4

ELMER Front settings file format

When an ELMER Front session is started, values for various control variables can be initialized from an input file. This appendix describes the structure of this file and the keywords which are available for setting initial values to these variables.

All keywords given in the file should be within a single section called 'User Settings'. So, the basic structure for the file is the following:

```
User Settings
   Keyword value
   Keyword value
   ...
End
```

The order of the keywords is free. In the following, all keywords starting with 'Default' are keywords whose values can be stored only in a settings file. Other keyword values can be stored also in a model file and they will be used automatically when the model file is loaded. However, the user can prevent this by defining the settings parameter 'Default Use Model Settings' to be false. This can be done either in a settings file or by using the Settings command in the Edit menu.

On each line, characters after '!' or '#' are interpreted as comments.

A string value must be given in quotes if it contains spaces or if it is given without a preceiding 'String' data type specifier. Otherwise quotes are not necessary, but it is safer to always quote the strings. Quotes within quotes are not allowed.

A logical value can be either 1 (meaning true) or 0 (meaning false).

In the following keywords are in the same order as they are displayed in the Settings window. The meaning of each keyword is explained in the context of the Edit menu in Chapter 2.

The keyword values given below are just for an example. The user should of course enter own values for the keywords which he or she defines in the settings file.

```
Default Model Directory
   String "/elmer/Models"

Default Cad Files Directory
   String "/elmer/data/cad_files"

Default External Mesh Files Directory
   String "/elmer/data/mesh_files"

Default Include Path
   String "/elmer/input"

Default Log Directory
   String "/elmer/work"

Default Results Directory
   String "/elmer/work"

! Should the settings stored in a model file used when it is loaded
!
Default Use Model Settings
   Logical 0

Auto Load Mesh
   Logical 1

Auto Save Model
   Logical 1

Auto Save Solver Input
   Logical 1

! Command for an external logfile browser (here in read only mode)
!
Browser Command
```

```
   String "C:\Program Files\TextPad\TXTPAD32.EXE -r"

! Command for an external editor (for editing Solver input file etc.)
!
Editor Command
   String "C:\Program Files\TextPad\TXTPAD32.EXE"

! The following browse mode flags are available
! Logfile: ouput is directed to a logfile
! Shell:   output goes to system console
! None:    no output at all
!
Browse Mode Gebhardt Factors
   String None

Browse Mode Mesh
   String Shell

Browse Mode Procedure Compiler
   String Logfile

Browse Mode Solver
   String Logfile

Browse Mode View Factors
   String Logfile
```

Chapter 5

ELMER Definitions File format

5.1 Introduction

This appendix describes the ELMER Definitions File format.

By default, ELMER Front supports only a set of predefined equations (Navier-Stokes, KE-Turbulence, Heat Equation, Stress Ananlysis and Advection Diffusion Equation). This means that the data entry panels contain only fields for these equations.

Using a definition file the user can define new panel fields and modify the properties of the existing fields. When a definition file is loaded into ELMER Front, the new user defined equations behave practically similarily as the predefined equations. However, there is one limitation. There is no mechanism to define procedures for checking the data integrity. The normal data checking (checking entry values for numeric fields etc.) is automatically in use, but a more complicated 'cross-checking' is not possible for the user defined fields.

It should be also noted that the definition file is only meant to create the user interface for the new equations. Any procedures (dynamic libraries) needed for solving the equations must be defined elsewhere.

A basic structure in a definition file is an equation section. In this section the user defines the name of the new equation, the variable to be solved and the procedure (the library and the function names) for the equation. An equation block can add a completely new equation or it can modify an existing equation. However, it should be noted that in general is not useful to add new fields to the predefined equations (like Heat Equation), because the existing (predefined) code for these

equations would not use any of these new fields.

An equation definition contains normally also some panel sections where the new fields related to the equation are added to each relevant panel (Materials, Initial Conditions etc.). When a panel section is within an equation section, the field definitions will affect only the equation in question. When a panel section is outside any equation section, the field definitions will affect all equations. For example, a new material panel field would be added to all equations.

It is also possible to include other definition files within a definition file. The include command must be outside any equation or panel sections.

5.2 ELMER Front model file and user definitions

When ELMER Front starts, it creates an internal list of field definitions which are related to the predefined equations mentioned above. If a definition file is loaded, this list is enlarged and modified and these changes are reflected in the data entry panels. So, loading a definition file can be seen as a mechanism to add new fields to the system.

When a model file is saved, all new field names and their values are saved with the model file. However, it is important to notice that the field property definitions for the new fields are not stored in the model file. If the model file is loaded later in a new ELMER Front session, without first loading the original definition file(s), these new fields are considered unknown for the system and all the data related to them is removed from the internal data structures. In this stage the original model file is still of course unmodifed, but if the user now saves the modified model data, all the data related to the new fields is permanently lost.

So, it is essential to load proper definition files into the system before loading a model file which contains user defined fields. It is the user's responsibility to keep record of the definition files related to any model file.

5.3 Loading definition files

A definition file is loaded automatically if a file named 'front.edf' is found in the 'ELMER_HOME/lib' directory or if the user has defined an environment variable named 'ELMER_FRONT_DEFINITIONS' and it contains a file name refering to a definition file.

A definition file can be also loaded at the start time by using the command line argument:

- --definitions-file=*definitions file name*

A definition file can be loaded also during the ELMER Front session with the Load definition file command in the File menu.

Definition files are searched and loaded in the above order and any later definition overwrites a possible previous definition.

5.4 The format

The basic structure of a definition file is as follows:

```
Equation equation name
  equation properties ...
  Panel panel name
    Field name
      field properties ...
  End Panel
End Equation
Include path
Panel panel name
  Field name
    field properties ...
End Panel
```

In the following format description one of the letters (`s, r, i, l`) after a keyword means that some data value is needed in that place and the type of this data should be a string, real, integer or logical.

A string value should be given in quotes if it contains spaces. Quotes within quotes are not allowed.

A logical value should True (or 1) or False (or 0).

On each line, all characters after '!' or '#' are interpreted as comments.

Lines can be continued by entering '
' or '&' as the last non-blank character in the line. Any leading white space is removed from the input line before it is added to the line being continued.

5.4.1 The `Equation` section

An equation definition is always a separate section, it cannot be inside any other section.

```
Equation s                ! Equation name (s) is needed

  Solver Procedure s1 s2 ! Library (s1) and function (s2) names are needed
  Variable s              ! Variable name (s) is needed
  Variable DOFs i         ! Degrees of freedom (default 1)
  Status Line Name s      ! Status line name for the eq. (default UDn)
  Process Table Name s    ! Process table name for the eq. (default UDn)

  ! NOTE: These properties define how the variable of the equation
  ! is displayed in the Initial Condition and Boundary Condition
  ! panels, where it is added automatically (for entering initial
  ! values and Dirichlet boundary condition values).
  ! To see the role of each property and the default values, see
  ! the field property explanations below
  !
  ! NOTE: The variable field is labelled autmatically based on the
  ! variable name, but the user can also set the label here.
  !
  Can Be Procedure l
  Can Be Table l
  Hide l
  Index Type s
  Label s
  Label Width i
```

```
  Label By Coordinate l
  Initial Value r
  Subpanel i
  Unit Label s
  Widget Width i

  ! Panel field definitions given under an equation section
  ! are active only for the specified equation.
  ! There can be multiple panel sections in an equation section
  !
  Panel s                ! Panel name (s) is needed
    ...                  ! Add panel definitions here
  End Panel              ! End Panel keyword is needed

End Equation             ! End Equation keyword is needed
```

5.4.2 The `Panel` section

When a panel section is outside any equation section, the field definitions affect all equations. There can be multiple panel sections.

```
Panel s                  ! Panel name (s) is needed

  Field s                ! Field name (s) is needed

    ! NOTE: This is a list of all possible field properetie
    ! in the panel section.
    ! To see the role of each property and the default values, see the field
    ! property explanations below
    !
    Add Absolute Box l
    Can Be Procedure l
    Can Be Table l
    Hide l
    Index Type s (Pre, Post)
    Initial Value (s/r/i/l)
    Label s
    Label By Coordinate l
    Label Width i
    Subpanel i
    Type s  (String, File, Real, Integer, Logical)
```

```
    Unit Label s
    Values {list of values}
    Widget s (Entry, Check Box, Browsable File, Selection List)
    Widget Width i

  Field s                ! A new field starts
    ...
  End Field              ! NOTE: End Field is optional!

End Panel                ! End Panel is needed
```

5.4.3 The `Include` section

An include section must be outside any equation and panel sections.

```
Include s               ! Path (s) for the include file
```

5.4.4 Panel names

The panels where new fields can be added are predefined and they are identified in the definition file by the names used in the ELMER Front menus. An alternative name is given in parenthesis (it is the name used in the sif-file, when applicable):

- Model Parameters (Model Parameter)
- Simulation Parameters (Simulation)
- Physical Constants (Constant)
- Equations (Equation)
- Body Parameters (Body Parameter)
- Boundary Parameters (Boundary Parameter)
- Initial Conditions (Initial Condition)
- Body Forces (Body Force)
- Materials (Material)
- Boundary Conditions (Boundary Condition)
- Solver Settings (Solver)

5.4.5 Field properties

The field properties are the following:

```
Add Absolute Box l

   If true, an Abs check box is added to the right side of the
   field.
   This property is meaningul only for a Browsable File field.
   By default, a browsed file name is stored with full path.
   If this check box is visible in the panel and it is not checked, only
   the file name is stored.
   Default value is False.

Can Be Procedure l

   If true, the Procedure check box in the  panel is made active when
   the cursor is on the field. This property is meaningul only for
   numeric entry fields.
   Default value is True.

Can Be Table l

   If true, the Table check box in the  panel is made active when
   the cursor is on the field. This property is meaningul only for
   numeric entry fields.
   Default value is True.

Hide l

   If true, the field is not displayed in the panel, but it is still
   written to the sif-file (if a non-blank Initial Value is defined)
   Default value is False.

Index Type s (Pre, Post)

   The variable name of an user defined equation can be used to "index"
   field names. This means that the variable name can be part of the
   field name. If the index-type is set to Pre, the field mames will
   be in the format 'VariableName FieldName'. If the index-type is Post,
   the format is 'FieldName Of VariableName'.
   For example, if one of the variables in a user defined equation were
```

```
   named as 'Oxygen' and a material panel field named
   as 'Diffusivity' were defined, a Pre-index would create a field
   named as 'Oxygen Diffusivity' and a Post-index would create a field
   named as 'Diffusivity Of Oxygen'. These name would be used also in
   the sif-file.
   No default value.

Initial Value (s/r/i/l)

   Field's initial value. This value should match the value type
   of the field.
   No default value.

Label s

   Field's label in the panel.
   No default value.


Label By Coordinate l

   If true, a coordinate symbol is added to the field's label.
   The x-coordiante corresponds to the first degree of freedom etc.
   This property is meaningul only for the variable of the equation and
   it is not applicable to the normal panel fields.
   Default value is False.

Label Width i

   Label's width in characters. This property can be used to align
   widgets in the panel by defining the same label width for each
   field to be (vertically) aligned.
   Default is the length of the label string (Label + Unit Label).

Subpanel i

   Field are grouped into subpanels (columns) in the panels.
   A panel can can be reasonably devided into 3 or 4 subpanels,
   depending on the maximum field width (label width + widget width).
   Default value is 1, but in the Equation panel the first
   available subpanel is 2. which also the default value (the
   first subpanel is reserved for the equation list)
```

```
Type s  (String, File, Real, Integer, Logical)

   Field's data type: Default is Real

Unit Label s

   This label is added after the field label.
   No default value.

Values {list of values}

   A list of values in the format: {First Second "Third value" ...}
   NOTE: The braces must be entered!
   This property is meaningful only for a Selection List field.
   No default value.

Widget s (Entry, Check Box, Browsable File, Selection List)

   The widget type of the field in the panel.
   Browsable File is an entry field with a file browser button, its
   default data type is File.
   Selection List is a drop-down menu for a list of predefined values.
   Default widget type is Entry, but for a logical field the default
   type is Check Box.

Widget Width i

   Widget's width in the panel measured as characters.
   Default value for an Entry field is 12 and for a
   Browsable File field it is 15.
```

5.5 An example file

The following example file is a modified version of the sample definition file which is delivered with the ELMER package (.../samples/other_files/front.edf). The sample file defines all fields needed for the Helmholtz solver. This equation is an example of a 'user defined' equation and it (ie. the dynamic library) is also delivered with the ELMER package.

The modifications here are quite artificial and they are made only to

illustrate some field properties which are not used in the sample file.
See the comments below.

```
! Frequency is a model wide feature (not related to a specific body
! or boundary), so it must given as model level data.
!
! NOTE: This field could be added also to the Helmholtz solver's
! equation panel
!
Panel "Simulation Parameter"
  Field "Angular Frequency
End Panel

! Some boundary related new fields
!
Panel "Boundary Parameter"

  ! This flag is used to calculate the sound energy at some
  ! interesting boundaries.
  !
  ! NOTE: This routine is not available in the Helmholtz
  ! solver delivered with the system. Here the user has made a special
  ! version for the equation!
  !
  Field "Compute Sound Energy"
    Type Logical

  ! This serves only as an example of the Selection List
  ! field. There is no use for this field in the Helmholtz equation!
  !
  Field "Sound Boundary Type"
    Type String
    Widget "Selection List"
    Values { "Sommerfeld boundary" "Reflective boundary" &
             "Pressure releasing boundary" Other }

    ! NOTE: If no initial value is given for a selection list, it
    ! would be the first list item
    !
    Initial Value "Other"

End Panel
```

```
! A new equation is added
! ========================
!
Equation "Helmholtz Solver"

  Status Line Name "Hlmz"
  Process Table Name "HH"

  ! NOTE: This is a user modified version of the Helmholtz solver
  !
  Solver Procedure "HelmholtzSolve2" "HelmholtzSolver"

  ! The name for the variable to be solved (sound pressure)
  !
  Variable "P"

  ! NOTE: A keyword (here Variable DOFS) can be also entered
  ! in quotes. This is safer for multiword keywords, because an
  ! incorrectly written input like (Variable DOFFs) would not be
  ! interpreted as a Variable keyword with the value "DOFFs 2".
  ! Instead, it could be catched as an error!
  !
  ! Sound pressure is a complex valued variable, so two degrees of
  ! freedom
  !
  "Variable DOFs" 2

  ! The autmatic labels would be "P 1" and "P 2", but it is possible
  ! to define a different label. Now the labels (in the
  ! Initial Condition and Boundary Condition panels) will be
  ! "Sound pressure 1" and "Sound pressure 2"
  !
  Label "Sound pressure"

  Panel "Solver"

    ! We add an extra field to the Helmholtz equation's
    ! solver parameters panel.
    !
    ! NOTE: In practice this option is not useful for the
    ! Helmholtz eqution!
```

```
    !
    Field "Bubbles"
      Type "Logical"
      Initial Value "True"

  End Panel

  ! Sound medium is air, so some proper default values
  !
  Panel "Material"

    ! The user wants to hide the density field completely,
    ! so it is not displayed at all and the value is always
    ! the initial value!
    !
    Field "Density"
      Hide 1
      Initial Value 1.224

    Field "Sound Speed"
      Initial Value 340

    Field "Damping"
      Initial Value 0
  End Panel

  ! Some special Initial Condition panel settings
  !
  Panel Initial Condition

    ! NOTE: Field name is derived from the variable name
    ! and the degrees of freedom value (the latter is used only
    ! if the variable has more than one degrees of freedom!)
    !
    Field "P 1"
      Hide 1
      Initial Value 0.0

    ! NOTE: By default the dof-identifier would be added to the base
    ! label which was earlier set to be "Sound pressure". This panel
    ! specific definition sets the label exactly to the user defined
    ! value
    !
```

```
    Field "P 2"
      Label "Sound pressure B"

  End Panel

  ! These flux and impedance fields are needed for the boundary
  ! conditions. They cannot be concluded from the
  ! variable of the equation (sound pressure), so
  ! the user must add these fields to the system
  !
  Panel "Boundary Condition"

    Field "Wave Flux 1"

      ! This way it is easier to remember that the first
      ! component is the real part and the second component
      ! is the imaginary part
      !
      Label "Wave flux Re"

    Field "Wave Flux 2"
      Label "Wave flux Im"

    Field "Wave Impedance 1"
      Label "Wave impedance Re"

    Field "Wave Impedance 2"
      Label "Wave impedance Im"

  End Panel


End Equation


! An existing equation is modified
! ================================
!
Equation "Heat Equation"

  ! NOTE: Solver settings is a panel where new field can be
  ! added to the predefined equations because the related solving
  ! methods are built into the standard ELMER solver. (They have been
```

```
  ! left as a user defined fields, because they are typically needed
  ! only for some specific problems)
  !
  Panel "Solver"
    Field "Bubbles"
    Type Logical
    Initial Value "True"

    ! This field will be hidden, there is no way to
    ! turn it off!
    !
    Hide "True"

  End Panel

End Equation

! An optional end of file keyword
!
End File
```

Chapter 6

ELMER Mesh Generator input file format

6.1 Model description

The input model consists of a number of bodies that are divided into layers. The mesh type can be selected separately for each layer. The layers are represented as collections of closed loops of edges: One outer boundary loop and possibly several inner cavity loops. The edges are defined by lists of nodes that define a broken line segment.

Each loop is a sequence of directed edges. The edges are directed so that the defined layer is on the left side of the edges. So the outer loop goes counterclockwise around the layer and the inner loops clockwise around the cavities. Adjacent edges in a loop must share a common node, so that the last node in an edge is the same as the first node in the next one. The first node of the first edge must be the same as the last node of the last edge.

The nodes corresponding to the geometry vertices will have the same ids that the vertices have in the input file. Predetermined inner nodes can also be specified.

6.2 Mesh density control

The mesh density is defined by a h-value, that is, the preferred element side length. The density values can be described at any level of the

model hierarchy: A density value can be given for the whole model, for a body, for a layer, for an edge or for a vertex. The density values given for an entity always overrides those given for the higher level entities.

The density can be given as an absolute value or a value relative to the global density value. The actual density values in the middle of a layer are interpolated from the values on the border. Three types of background meshes can be used for the interpolation:

- An explicit background mesh can be given as a set of points with associated density values. Any mesh density values from the model are ignored.
- A delaunay triangulation can be made from the original geometry vertices.
- An orthogonal grid can be used to calculate the density values as a solution to a Laplacian boundary value problem. The grid density can be chosen from three values.

6.3 File structure

The file consists of a hierarchy of records that contain keywords and data. All keywords end with a colon. The file is divided into words separated by any kind of whitespace characters, so the file may be freely decorated by indentation or adding empty lines. The file is case sensitive.

Most of the fields in the records are mandatory and must be given in a fixed order. Several records may contain an optional mesh density field that has one of two keywords "H:" or "R:". The keyword "H:" is followed by an absolute density value and the keyword "R:" specifies a value relative to the global density value.

The input file is divided into 4 major fields:

- A header
- A list of vertices
- A list of edges
- A list of bodies

There can be lines of comment between these fields. Comment lines begin with an "!" or a "#". The vertex list is only ment for the vertices used for defining edges. The additional inner nodes are listed in the layer records.

6.4 The header

The header starts with the keyword **Geometry2D:** followed by 5 mandatory fields:

- **H:** The global mesh density value.
- **MeshScalingFactor:** The mesh scaling factor. All mesh density values are multiplied by this value.
- **Nodes:** The number of vertex records.
- **Edges:** The number of edge records.
- **Bodies:** The number of body records.

Example:

```
Geometry2D:
H: 0.1
MeshScalingFactor: 1.5
Nodes: 3
Edges: 2
Bodies: 1
```

All the density values in this file will be multiplied with 1.5, including the 0.1 given in the header. The mesh scaling factor is an easy way to change the overall density of the entire mesh.

6.5 The vertices

Each vertex record has the fields:

- **NodeId:** Two positive integers: a unique vertex id and a boundary id. If a boundary element is not required at this vertex, a zero or negative boundary id may be given.

- An optional mesh density field. See above.
- Two vertex coordinates.

Example:

```
NodeId: 1 1
R: 2.0
0 0
NodeId: 2 -1
H: 0.2
1 1
NodeId: 3 1
0 1
NodeId: 4 -1
-1 0
NodeId: 5 -1
-1 1
```

Only vertices 1 and 3 will produce a boundary element. Both the first and second vertices will produce a mesh density of 0.3. The last three vertices do not specify a density of their own.

6.6 The edges

Each edge record has the fields:

- **EdgeId:** two positive integers: a unique edge id and a boundary id.
- An optional mesh density field with an additional keyword "N:" that may be used to specify a fixed number of boundary elements.
- The number of vertices that define the edge. (At least two)
- A list of vertex ids

Example:

```
EdgeId: 1 1
H: 0.05
```

```
3 1 2 3
EdgeId: 2 2
N: 10
2 1 3
EdgeId: 3 3
2 1 4
EdgeId: 4 3
2 4 5
EdgeId: 5 3
2 3 5
```

The first edge is defined by three vertices and specifies a mesh density of 0.075. The second edge is defined by two endpoints and will be divided into 10 equal-sized segments (so there will be 11 nodes on that edge). Because the node 3 didn't specify a mesh density of its own, it will have a density that is the average of the densities of these two edges. That is 0.155.

6.7 The bodies

Each body record has the fields:

- **BodyId:** A unique body identification (a positive integer).
- An optional mesh density field.
- **ElementOrder:** The order of the elements to be created for this body. Either Linear or Parabolic.
- **Layers:** The number of layers in this body.
- A list of layer records.

Each layer record has the fields:

- **LayerId:** A unique layer indentification (a positive integer).
- An optional mesh density field.
- **LayerType:** The layer mesh type. Described below.
- A type-dependent part. Described below.

- **Loops:** The number of loops
- A list of loop records:
  - **LoopId:** A positive integer
  - **Direction:** 1 or -1. If the direction is -1, the loop direction is inverted.
  - **Edges:** The number of edges
  - A list of signed edge ids (negative means backwards)

The layer type can be one of the following:

- **Connect**: Nodes are generated on the layer edges and connected with a delaunay triangulation.
- **VoronoiVertex**: Nodes are inserted one by one into the place that has the greatest distance to the nearest existing node.
- **MovingFront**: Nodes are inserted by taking an accepted mesh segment and inserting a third node so that a good triangle is produced. The entire boundary is used as a starting front.
- **SSMFMovingFront**: The same as MovingFront, but the process is started from a single seed position. Advances to the direction where the distance to the nearest node is greatest.
- **SSSFMovingFront**: The same as SSMFMovingFront, but advances more locally.
- **QuadGrid**: A grid of quadrilateral elements. The layer must have only 1 loop with exactly four edges. The user must also make sure that the opposite edges will be subdivided into equal numbers of elements. This can be done easily with the GridSize-specification described below.
- **TriangleNEGrid**, **TriangleNWGrid**, **TriangleUJNEGrid**, **TriangleUJNWGrid**, **TriangleFBNEGrid** or **TriangleFB-NWGrid**: A QuadGrid that is split into triangles. NE/NW specifies the splitting direction of the first quadrilateral. UJ means an alternating direction and FB is a sceme that alternates only along the first and third edges.

The type-dependent fields in a layer of type VoronoiVertex, MovingFront, SSSFMovingFront or SSMFMovingFront:

- **FixedNodes:** The number of fixed mesh nodes and a list of vertex records.
- **BGMesh:** Background mesh type: One of Delaunay, Grid, SparseGrid, DenseGrid or Explicit.
- If background mesh is Explicit:
  Background mesh size and a list of nodes with two coordinates and a mesh density value. The other selections require no additional parameters.

If the type is SSSFMovingFront of SSMFMovingFront, the following additional fields:

- **Seed:** Seed type: Implicit or Explicit.
  If type is Explicit:
  - **Nodes:** Three node ids that define the initial front.

  If type is Implicit:
  - **Edge:** An edge id. A segment from the middle of the edge is used a the initial front.

Example:

```
BodyId: 1
ElementOrder: Linear
Layers: 1
  LayerId: 1
  R: 0.5
  LayerType: SSSFMovingFront
  FixedNodes: 1
    NodeId: 4 -1 0.5 0.5
  BGMesh: Grid
  Seed: Implicit
    Edge: 2
  Loops: 1
    LoopId: 1
    Direction: 1
    Edges: 2
      1 -2
```

This will create a mesh with one predetermined node at position (0.5, 0.5). The edge 2 has a negative sign, so the nodes of the edge are

taken in a reverse order. Thus this body is a single triangle formed by nodes 1 2 and 3. The triangulation will start a the middle of edge 2. The mesh density is controlled by a cartesian grid.

The type-dependent fields in a layer of type QuadGrid, TriangleNEGrid, TriangleNWGrid, TriangleUJNEGrid, TriangleUJNWGrid, TriangleFBNEGrid or TriangleFBNWGrid are:

- **GridSize:** The number of elements in which to split the odd and even edges of the layer. (Two positive integers.) If this field is omitted the edges are split as defined by the density values. The values given in a GridSize-field override any density values given for the edges.

Example:

```
BodyId: 2
ElementOrder: Linear
Layers: 1
  LayerId: 1
  R: 0.5
  LayerType: QuadGrid
  GridSize: 5 10
  Loops: 1
    LoopId: 1
    Direction: -1
    Edges: 4
4 -5 2 3
```

This body will be a grid of 50 quadrilateral elements. The outer loop is given clockwise, but the direction field tells the program to interpret it backwards. Another way to specify the grid size could be to give "N: 5" to edges 3 and 5 and "N: 10" to edges 2 and 4.