ADVENTURE_BCtool

A tool for setting boundary conditions , material properties and other advanced conditions

Version: 2.0

User's Manual

June 8, 2012 Ver.2.0

May 23, 2013 Ver.2.0Rev.A

May 24, 2013 Ver.2.0Rev.B

Feb. 26, 2014 Ver.2.0Rev.C

Revision history

2013/05/23 Ver.2.0 Rev. A

2013/05/24 Ver.2.0 Rev. B

2014/02/26 Ver.2.0 Rev. C

Table of contents

Part I

1. General

ADVENTURE_BCtool is a tool used to paste boundary conditions, material properties, and MPC conditions on a given mesh, and to create finite element analysis data for ADVENTURE_Thermal, ADVENTURE_Solid, and the solvers that can consider MPC* conditions.

Fig. 1-1 shows data dependencies of the BCtool with other modules of the ADVENTURE.

Note *: The MPC is the acronym of Multi-Point Constraint. A constraint expression is represented by a polynomial of the degrees of freedom of nodes.

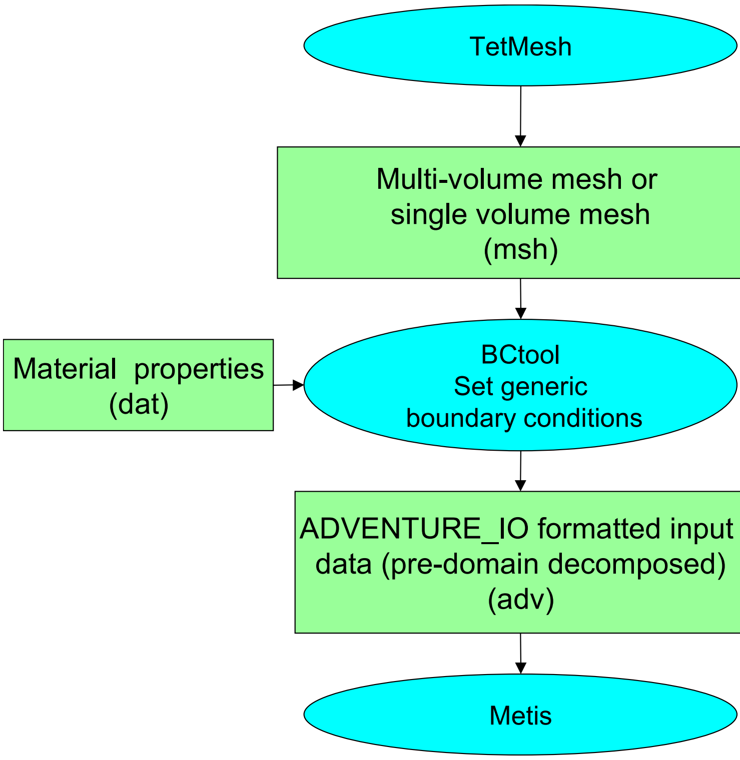


Fig. 1-1 Data dependencies of BCtool with other modules of ADVENTURE

2. Organization and Functions

2.1 Organization

The ADVENTURE_BCtool is a set of tools and consists of the mesh2pch, the BcGUI, the makefem3, the msh2pcm, the PcmMerge, MpcLocal2Global, a2adv, csv2adv, and MpcMasterSlaveTool. Some of the tools consist of multiple commands.

The msh2pch, the BcGUI, the makefem3, the msh2pcm, the a2adv, and the csv2adv are for setting generic boundary conditions, and the PcmMerge, the MpcLocal2Globa and the MpcMasterSlaveTool are tools dedicated to MPCs.

Each tool is summarized in Table 2.1-1.

Table 2.1-1 Summary of tools

Name of tools	Description	Remarks
a2adv	A generic tool to convert ASCII data (in .a format) to binary data of the ADVENTURE_IO format.	For generic boundary conditions.
BcGUI	A 3D and interactive application for setting boundary conditions.	For generic boundary conditions.
msh2pch	A tool to extract surfaces from an msh model.	For generic boundary conditions. Consists of the faceOfMesh, and the makepch.
msh2pcm	A tool to extract surfaces from a multi-volume mesh model.	For generic boundary conditions.
makefem3	A tool to make an integrated input file for the Solid and the Thermal.	For generic boundary conditions.
MpcLocal2Global	A tool to convert MPC conditions of multiple types to those in the ADVENTURE_IO format.	Associated with MPCs.
MpcMasterSlaveTool	A tool to automatically generate fixed MPC conditions between surface groups in contact.	Associated with MPCs.
PcmMerge	A tool to merge multiple meshes into that applicable to MPCs.	Associated with MPCs.
csv2adv	A tool to convert a CSV data file to an ASCII data file (.a format) which is an input for the a2adv.	For generic boundary conditions.

2.2 Functions

Types of supported elements

Linear tetrahedron, quadratic tetrahedron, linear hexahedron, and quadratic hexahedron

Analysis types that are supported

Elastic stress analysis

Elastic-plastic stress analysis

Thermal stress analysis

Heat transfer analysis

Transient stress analysis

Where boundary conditions can be set

Mesh surfaces that are grouped.

Primary nodes on the boundary of the mesh surface groups

Boundary conditions that can be applied

Load（X, Y, and Z direction, normal to a surface group）

Displacement（X, Y, and Z direction）

Gravitational acceleration

Temperature

Heat flux

Heat transfer

Heat radiation

MPC conditions that ban be applied

Fixed

Rigid Beam: 5 types

Simple Beam

Linear combination expression

Material properties that can be treated（Multiple materials are supported）

Young's modulus

Poisson's ratio

Work hardening parameter

Initial yield stress

Mass density

Linear expansion coefficient

Reference temperature

3. Operating Environment

Only the BcGUI operates in the environment of both Linux and Windows The msh2pch, the makefem3, the msh2pcm, the PcmMerge, the MpcLocal2Global, the a2adv, the csv2adv and the MpcMasterSlaveTool run principally on the Linux environment. They might run on Cygwin, but these tools will not be supported on Cygwin.

For more information about the tools other than those described in Section 3.1 through Section 3.4, refer to "2. The Operating Environment and Compilation" in Part II for the PcmMerge, "3. Operating Environments" in Part III for the MpcMasterSlaveTool, "3. The Operating Environment" in Part IV for the a2adv, and "2. The Operating Environment and Compilation", in Part V for the MpcLocal2Global, respectively.

3.1 Operating Environment of the BcGUI

BcGUI was tested and validated in the following operating environment.

Cent OS 4.4 (32bit)

Ubuntu 10.04 LTS Lucid Lynx (32bit, 64bit)

Microsoft Windows 7 Home Premium (64bit) SP1

3.1.1 How to Output Logs

Logs of this program will not be output normally. When logs need to be reported because of bugs, edit the following two lines in a file called ADVENTURE_BcGUI_Ver_2_0.lax in the folder where this module was installed.

```
lax.stderr.redirect=
lax.stdout.redirect=
```

should be changed as follows.

```
lax.stderr.redirect=console
lax.stdout.redirect=console
```

When this module is started in the usual way after saving, the "Command Prompt" window is displayed and logs will appear in it. Note that the "Command Prompt" window will disappear when this module is finished. If you specify a specific file name instead of console, the "Command Prompt" window will not be displayed, and logs will be written in the file. Note that in the windows environment, be sure to write two "\"s each like C:\\tmp\\ADVENTURE_BcGUI_Ver_2_0.log.

3.1.2 Modification of the Maximum Value of the Heap Memory

The heap memory of this module may become insufficient depending on the size of pch / pcg model files to be displayed. At that time, the symptom will be that the program will freeze. If the log of the program can be seen according to the instruction in Section 3.1.1, the last line of the log will be as follows.

java.lang.OutOfMemoryError

At that time, edit the following line in the ADVENTURE_BcGUI_Ver_2_0.lax, and increase the maximum value and the initial value of heap memory. Default values are 1024M bytes and 128M bytes, respectively.

```
# LAX.NL.JAVA.OPTION.JAVA.HEAP.SIZE.INITIAL
# -----------------------------------------
# Initial heap memory

lax.nl.java.option.java.heap.size.initial=128M

# LAX.NL.JAVA.OPTION.JAVA.HEAP.SIZE.MAX
# -------------------------------------
# max heap memory

lax.nl.java.option.java.heap.size.max=1024M
```

3.2 Operating Environment of the msh2pch and the msh2pcm

The msh2pch and the msh2pcm were tested and validated in the following operating environment.

Ubuntu 10.04 LTS Lucid Lynx (64bit)

3.3 Operating Environment of the makefem3

The makefem3 was tested and validated in the following operating environment.

Ubuntu 10.04 LTS Lucid Lynx (64bit)

Make GNU Make 3.81

Compiler: GCC 4.4.3

Library: The standard C library

* The PATH must include the directory of the a2adv.pl.

3.4 Operating Environment of the csv2adv

The csv2adv was tested and validated in the following operating environment.

Ubuntu 10.04 LTS Lucid Lynx (64bit)

Perl 5.10.1

4. How to Install

As for installation order relative to other modules, since the a2adv requires the ADVENTURE_IO, ADVENTURE_IO should be installed before this module. In order to compile the ADVENTURE_IO module, no other modules are required.

As for other tools, installation should be performed as follows. In the root directory of the BCtool package, open the Makefile using a text editor. The ADVIO_PREFIX near the top line designates the directory where the ADVENTURE_IO was installed, and the PREFIX designates the directory where the BCtool will be installed. Usually, it is all right to specify a same directory. Then, move to the root directory of the BCtool package in a terminal, and execute the following command.

```
% make
% make install
```

To install each individual tool, see "2. The Operating Environment and Compilation" in Part II for the PcmMerge, "4. How to Install" in Part III for the MpcMasterSlaveTool, "4. How to Install" in Part IV for the a2adv, "2. Operating Environment and Compilation" in Part V for the MpcLocal2Global.

4.1 Installation of the BcGUI

As for the BcGUI, it is pre-compiled. Therefore, no compilers are needed. Its language is Java (JDK 1.5.0 or later), and as a 3D library, the JOGL (Ver.1.1.1) is used. An interactive installer is provided.

In order to install the BcGUI on Windows, use the AdvBcG2032.exe (this is for a 32bit architecture, and for an amd64 architecture, use the AdvBcG2064.exe). Launch an installer and follow its instruction. As for Linux, use the AdvBcG2032.bin (this is for a 32-bit architecture, and for an amd64 architecture, use the AdvBcG2064.bin) and run it as follows.

```
% sh AdvBcG2032.bin
```

In the case of the Linux version, it is necessary to obtain the root privileges if it is installed in a system area. No root privilege is required for installation to the user's area. In addition, a warning like Fig. 4.1-1 may appear on the installer's last screen. Since there is no problem in using the program, please click on the "Done" as it is.

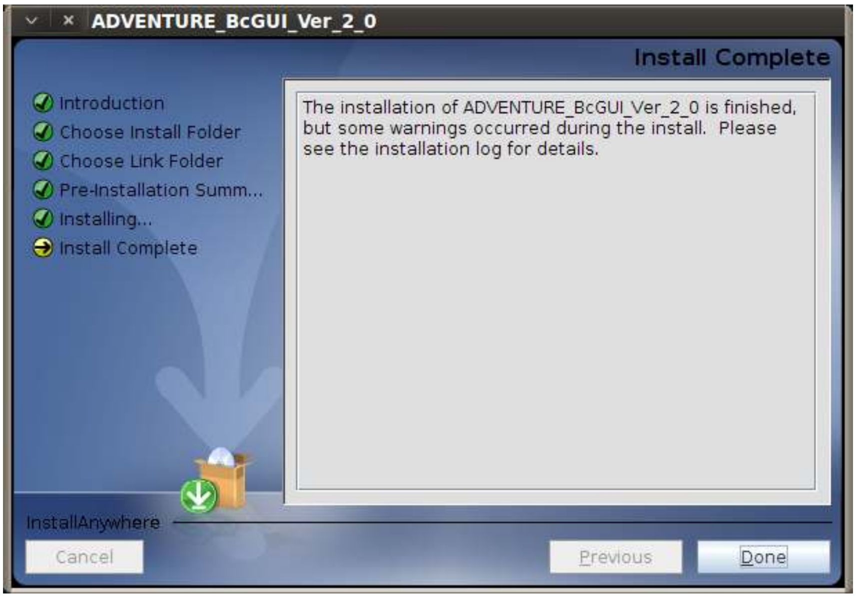


Fig. 4.1-1 A warning on the last screen of the installer

4.2 Uninstalling the BcGUI

If Windows Vista is the case, please start by selecting the "Start" and "Programs and Features" in the "Control Panel" (if Windows 7 is the case, start by selecting the "Start", "Control Panel", and "Uninstall a program" in "Programs" group). Look for the "ADVENTURE_BcGUI_Ver_2_0" from the list, select it and click the "Uninstall and change" button (if Windows 7 is the case, click the "Uninstall" button). Then, Fig. 4.2-1 will pop up. Click the "Uninstall" button (Fig. 4.2-2). When uninstallation is complete, Fig. 4.2-3 will be displayed. Click on the "Done" button.

Fig. 4.2-1 Start of uninstallation

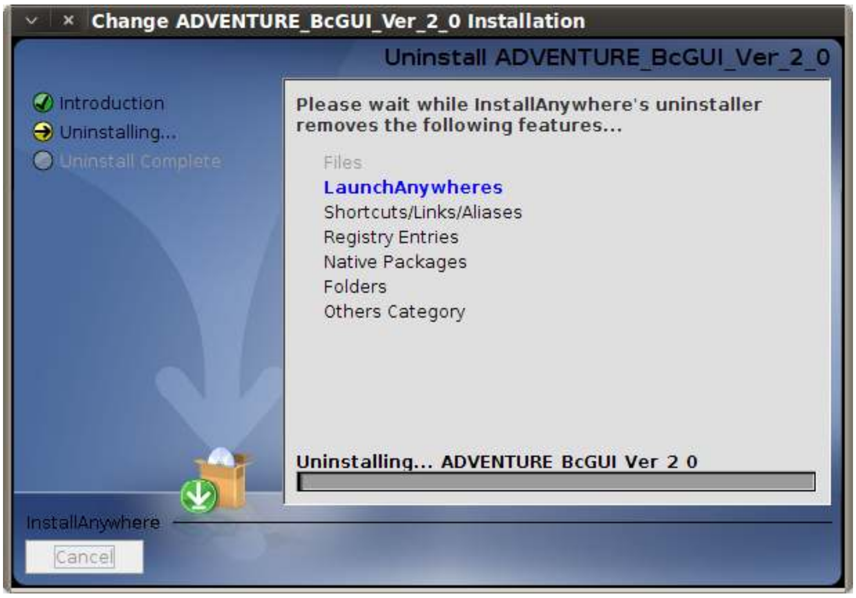


Fig. 4.2-2 Uninstallation is in progress

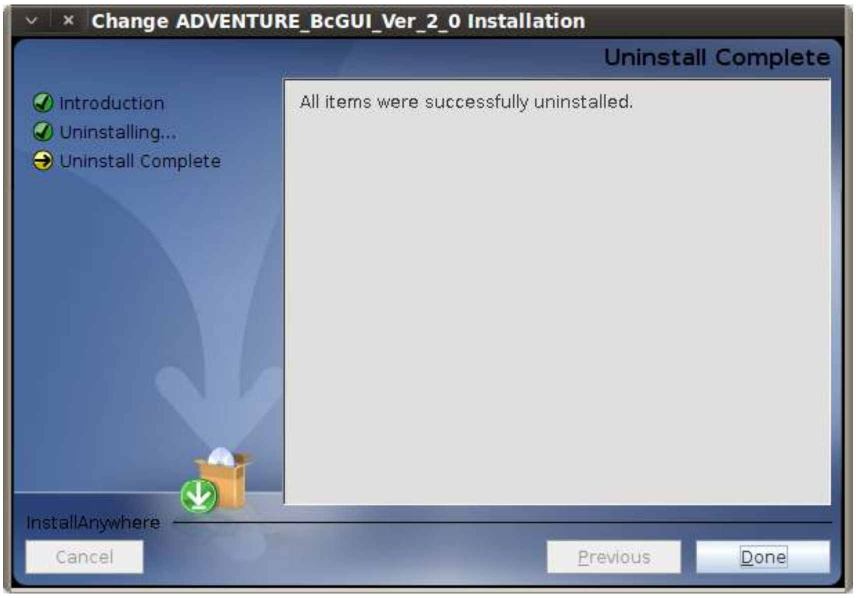


Fig. 4.2-3 Uninstallation is complete

4.3 Installation of the csv2adv

Copy the csv2adv.pl in some directory.

5. How to Use

5.1 Summary

The use of the BCtool can be roughly divided into the following three types.

(1)To create data which include boundary conditions for structural analyses such as constrains and loads (to be called as normal structural analysis boundary conditions later).

(2)To create data for structural analyses which include MPC conditions among multiple volumes and normal boundary conditions.

(3)To create data for thermal analyses which include thermal boundary conditions such as temperature and heat flux.

Fig. 5.1-1 shows the data flow for the first use type. The first use type utilizes the msh2pch, the BcGUI, and the makefem3 that calls the a2adv internally. In the case of dynamic analyses, the csv2adv is also used to create time history data. In the second use type, the way to set MPC conditions is divided into two. The first case is where fixed MPC conditions are created automatically between multiple volumes in contact, and the tool MpcMasterSlaveTool is used. The tool MpcMasterSlaveTool consists of two commands the MPC_mshmrg.pl and the MPC_assem2. The data flow about this method is shown in Fig. 5.1-2. The other case is where various MPC conditions are created manually between multiple volumes, and both an interactive operation of the BcGUI and the MpcLocal2Global are used. The data flow for the latter method is shown in Fig. 5.1-3.

The third use type utilizes the msh2pch, the BcGUI, and the makefem3 that calls the a2adv internally. Fig. 5.1-4 shows the data flow about this method.

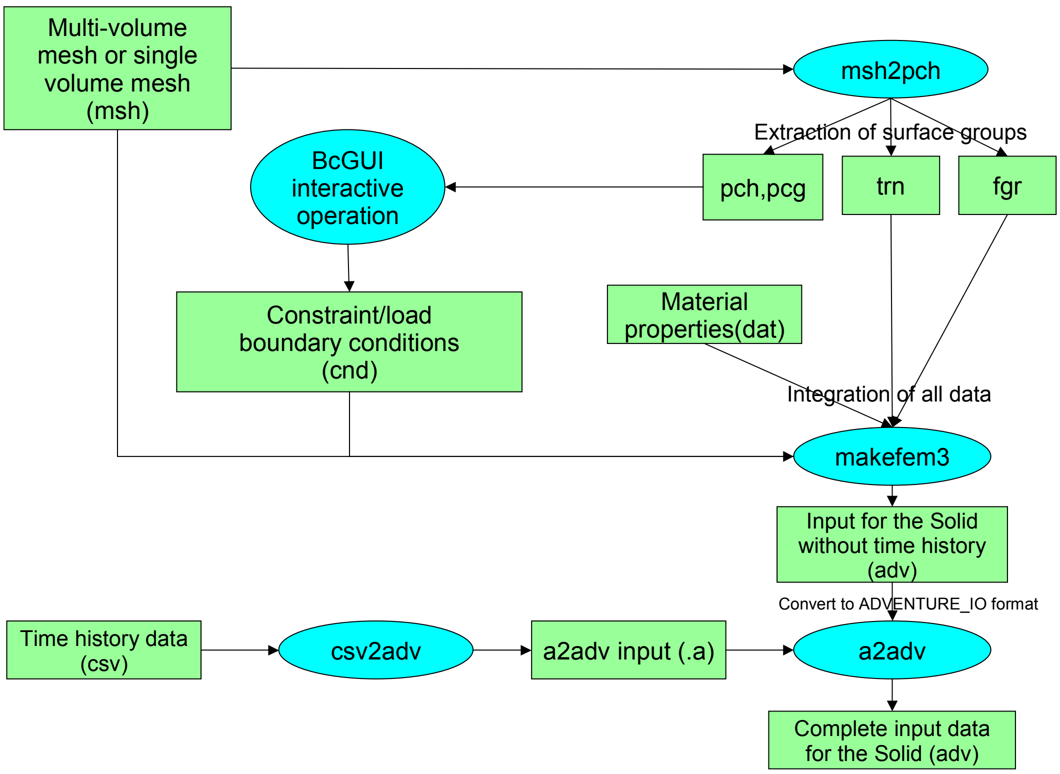


Fig. 5.1-1 The data flow for the case of setting normal boundary conditions

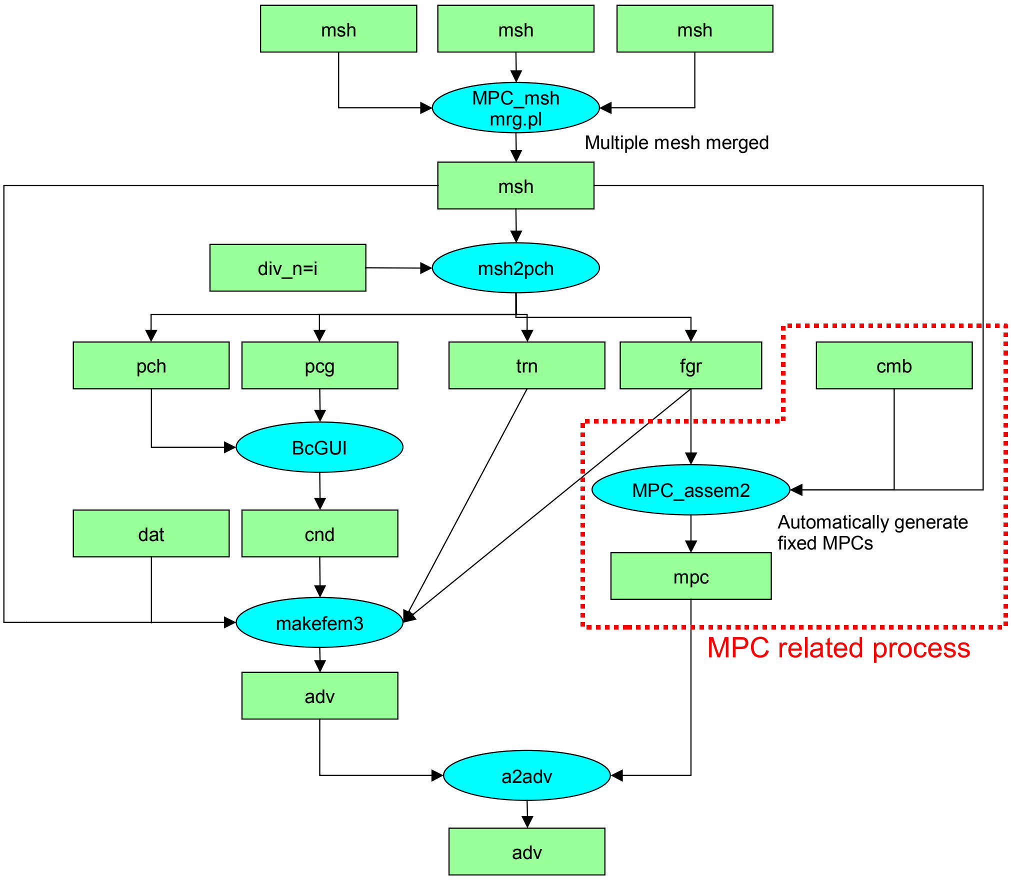


Fig. 5.1-2 Data flow when MpcMasterSlaveTool is used

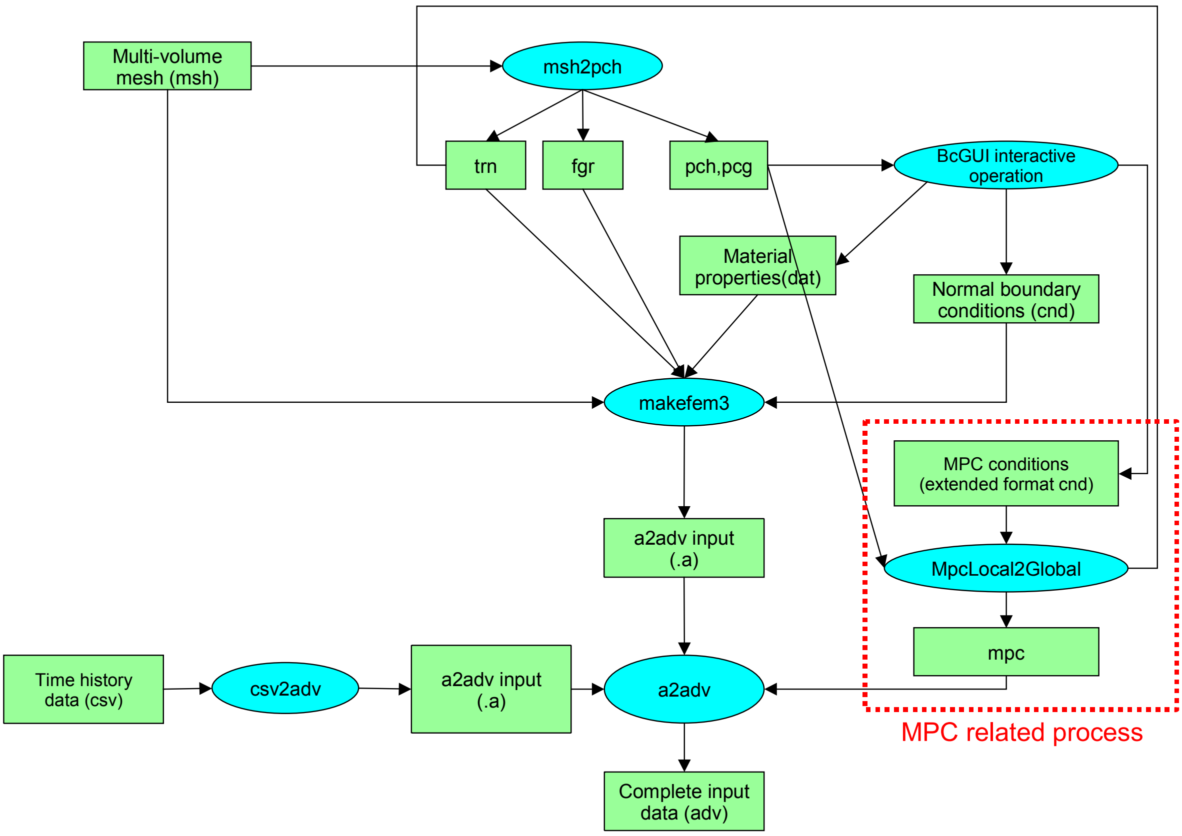


Fig. 5.1-3 Data flow diagram when using the MpcLocal2Global

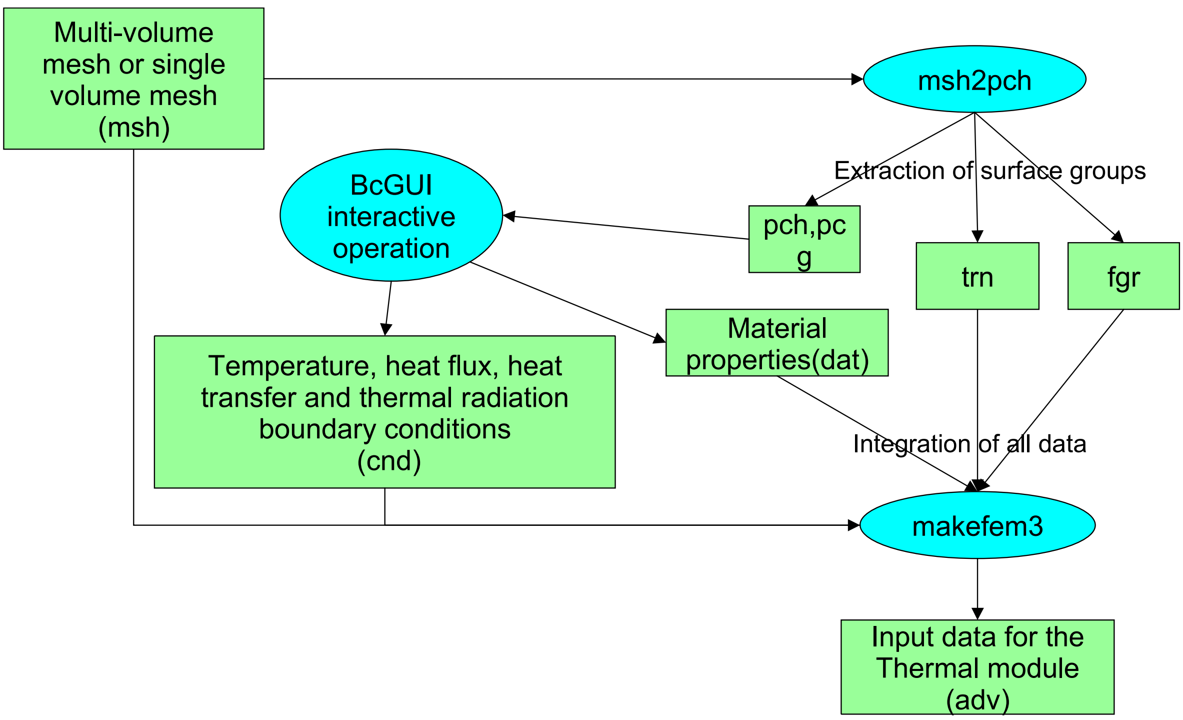


Fig. 5.1-4 Data flow diagram of when creating data for the Thermal

5.2 How to Use at the Time of Setting Normal Structural Analysis Boundary Conditions

To attach boundary conditions and material properties to the mesh using the ADVENTURE_BCtool, the following five steps of procedure are needed.

(1) Extraction of the surface of the mesh,

(2) Setting boundary conditions by the BcGUI (including setting of material properties),

(3)Creating an integrated input file for the Solid,

(4)Adding time history data

5.2.1 Step 1 Extraction of Mesh Surface

At this step, the data of extracted mesh surface groups are converted into the GUI input format. The following input and output files are used.

Input file:

Mesh data file (extension: .msh)

Output file:

Mesh surface data file (extension: .fir)

Extracted surface mesh data file (extension: .pch)

Surface patch group data file (extension: .pcg)

Global index file (extension: .trn)

At this step, the shell script msh2pch is used for conversion. There are two arguments in msh2pch command line.

```
% msh2pch mshFile div_n
```

mshFile : mshFile is the name of a mesh file. The command is applicable to both a single-volume and a mulple-volume file can be converted.

div_n : div_n is a denominator that divides 90 degrees to obtain the dihedral angle which works as a criterion for mesh surface group extraction. A surface element, which is separated from the neighboring element by the dihedral angle more than the specified value, will be classified to a different surface group.

Example-1） The filename of mesh data is Model.msh, and a dihedral angle criterion is 30 degrees (30 degrees = 90 degrees/3). Then,

```
% msh2pch Model.msh 3
```

Example-2） The filename of mesh is Model.msh, and a dihedral angle criterion is 45 degrees (45 degrees = 90 degrees/2). Then,

```
% msh2pch Model.msh 2
```

An appropriate dihedral angle varies from model to model and should be adjusted by the user after checking whether the grouping is detailed enough or not by displaying the topology with BcGUI.

If the filename of mesh data is Model.msh and if a dihedral angle is 90/N degrees, the following output files will be created:

Model_N.fgr	: A mesh surface data file
Model_N.pch	: An extracted surface mesh data file
Model_N.pcg	: A surface patch group data file
Model_N.trn	: A global index file

5.2.2 Step 2 Setting of Boundary Conditions by BcGUI

Boundary conditions can be set to mesh by an interactive executable BcGUI. The following input and output files are used.

Input files:

Extracted surface mesh data file (extension: .pch)

Surface patch group data file (extension: .pcg)

Output file:

Analysis conditions file (extension: .cnd)

5.2.2.1 Launching BcGUI

If the Windows version is used, launch it by clicking a shortcut that was made when the application was installed. Similarly launched with the Linux version.

It is also possible to launch the BcGUI by designating runtime arguments as follows.

```
% ADVENTURE_BcGUI_Ver_2_0 pchFile pcgFile [-icnd cndFile] [-ocnd outFile]
```

pchFile	is the extracted surface mesh data file.
pcgFile	is the surface patch group data file.
cndFile	is the file with analysis conditions which is read automatically at the moment of BcGUI startup.
outFile	is the file with analysis conditions which is created automatically when BcGUI is terminated.
[…]	the options in brackets can be omitted.
-icnd	is the option used to read automatically an existing file with conditions for analysis at the start of BcGUI.

-ocnd is the option used to set a condition file, which will be saved automatically when the BcGUI is terminated.

The following description assumes the application is launched through a shortcut. If it is started up by a command, section 5.2.2.2 can be skipped.

After startup of BcGUI, a default window will appear on the screen as shown in Fig. 5.2.2.1-1.

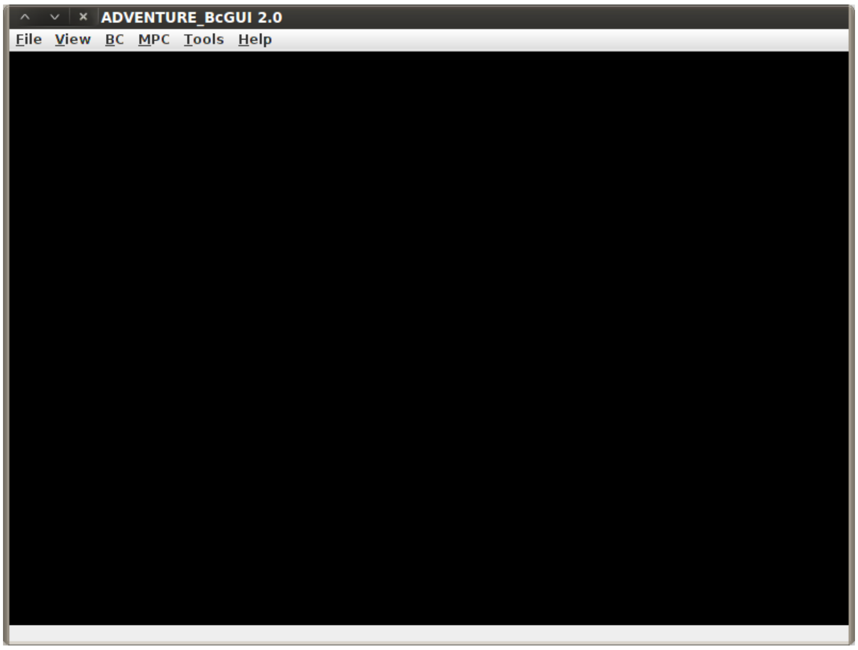


Fig. 5.2.2.1-1 Initial window

5.2.2.2 How to Select Model Files

Select "File" and then "Open File" in the menu bar. A dialog appears as shown in Fig. 5.2.2.2-1.

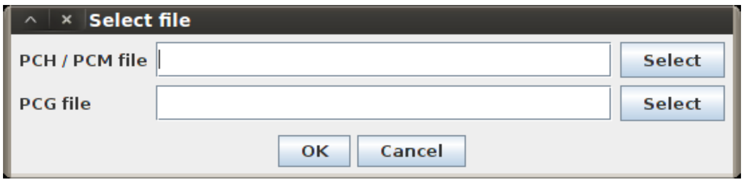


Fig. 5.2.2.2-1 A dialog to select pch/pcg files

Two "Select" buttons are available. Choose a pch file by the upper button and choose a corresponding pcg file by the lower button. Then the chosen files are shown as in Fig. 5.2.2.2-2.

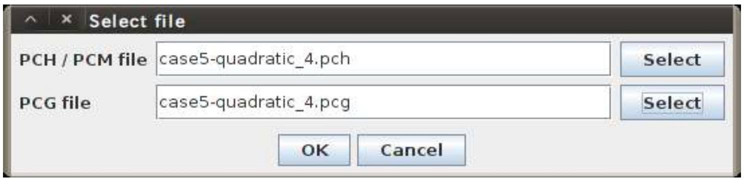


Fig. 5.2.2.2-2 After selection of pch/pcg files

If case5-quadratic_4.pch and case5-quadratic_4.pch in the samples folder in the installation folder are selected and the ”OK” button is clicked, a model is loaded and the application window looks as shown in Fig. 5.2.2.2-3.

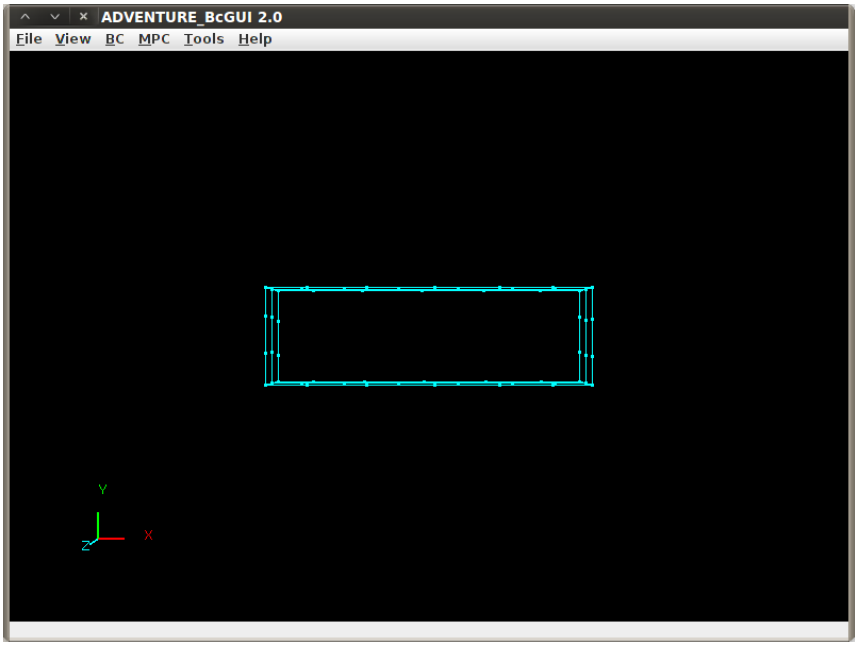


Fig. 5.2.2.2-3 Initial view when a model case5-quadratic_4 is displayed

5.2.2.3 How to Change User Viewpoint by the Mouse

Translation	Press the left mouse button and move the mouse.
Rotation	Press the mouse’s wheel button and move the mouse.
Zooming in	Press the right mouse button and move the mouse downward.
Zooming out	Press the right mouse button and move the mouse upward.

5.2.2.4 How to Select a Linear Node on the Surface and a Surface Group

Select a node	Left-click (pick) a node
Select a surface group	Pick a node first, and right-click once. Then a surface group adjacent to the node is selected. Other surface groups adjacent to the node can

be selected one after another by pressing the right mouse button repeatedly. After the last surface group is selected, the first surface group is selected again.

Select a node inside a surface group — Only while setting MPC conditions, a node inside a surface group can be picked. After a surface group turns red by picking, left click a node inside the surface group.

The picked object is highlighted in yellow (in the case of an MPC condition, red is used). A node or a surface group to give a boundary condition to is picked first, and then a menu item for boundary conditions are selected (in the case of an MPC condition, picking of an object will become possible only after a menu item for the MPC conditions is selected from the menu bar and a dialog pops up).

When picking is performed, information of a currently selected node in the pch file or that of a currently selected surface group in the pcg file is shown in the status bar at the bottom of the application window.

Picking of a surface group can be cancelled by pressing the ESC key. As for a node, if the picked node is picked again, selection is cancelled. (in the case of MPC conditions, picking can be cancelled by clicking the "Clear" button in the MPC setting dialog).

5.2.2.5 How to Set Boundary Conditions (Constraints on a Surface Group)

Pick a node on an edge adjacent to the surface group to be constrained as shown in Fig. 5.2.2.5-1.

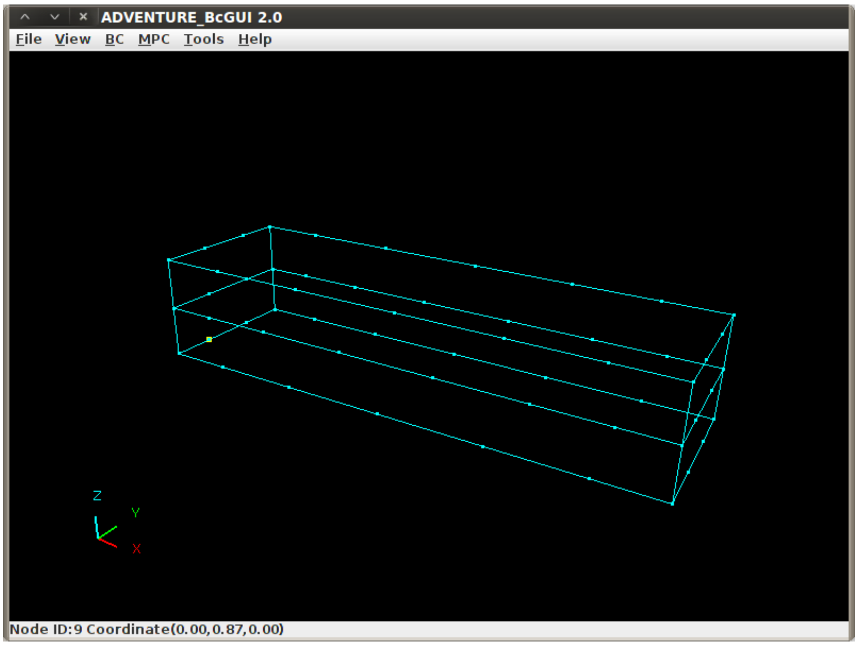


Fig. 5.2.2.5 -1 Selection of a node

Then, right-click twice, and a surface group at the end of the part is highlighted (Fig. 5.2.2.5-2).

The number of the right-click necessary to highlight the same surface group may vary evey time even if a same node is selected.

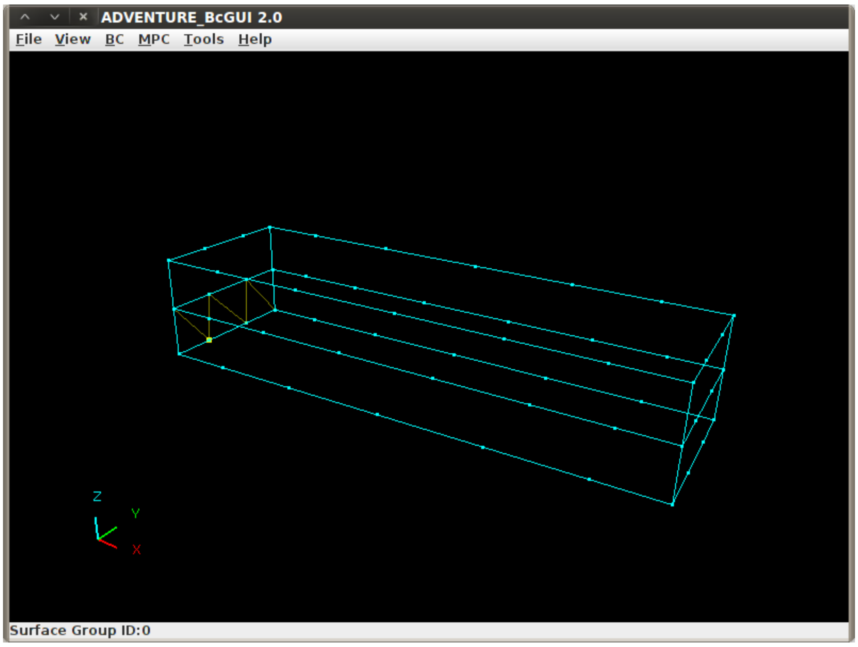


Fig. 5.2.2.5-2 How the end surface group to be constrained is highlighted

Then select "BC", "BC(Solid)" and "Add Displacement" in the menu bar, and a dialog in Fig. 5.2.2.5-3 will pop up.

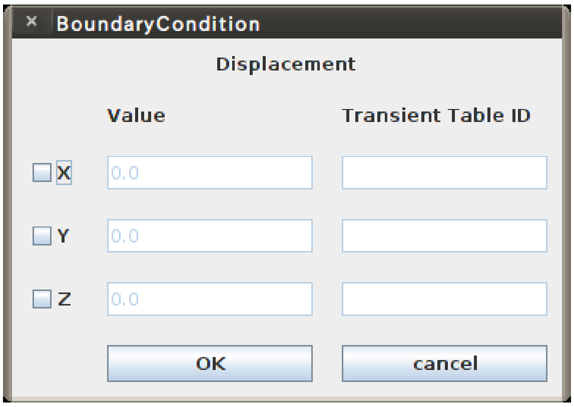


Fig. 5.2.2.5-3 A dialog for setting constraints (before entering values)

Check all the directions to apply constraints (e.g. Fig. 5.2.2.5-4). If values are entered, forced displacements can be designated. "Transient Table ID" means the ID of the time history data table that will be prepared in Section 5.2.5. It is called time history ID in Section 5.2.5. It is used only if a transient boundary condition is given. A different ID can be designated for a different degree of freedom. For each checked checkbox, an ID can be designated in the "Transient Table ID" input field. The entered ID must correspond to an ID given in the time history data table.

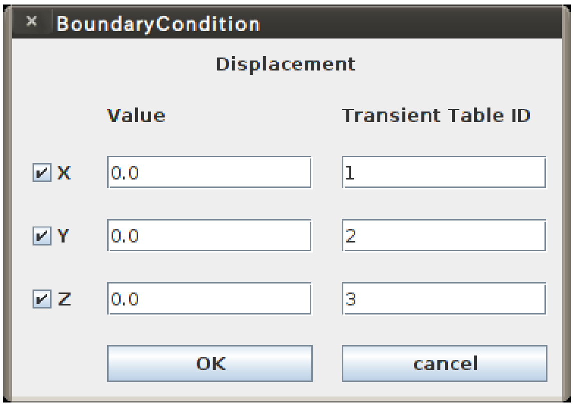


Fig. 5.2.2.5-4 A dialog for setting constraints (after values are entered)

After necessary values are entered, click "OK".

5.2.2.6 How to Display Boundary Conditions (Checking a Surface Group with Constraint)

Cancel selection of a surface group if any. Then select "View", "Boundary Condition" and "View Displacement" in the menu bar. As shown in Fig. 5.2.2.6-1, the surface group with constraints will be highlighted in green.

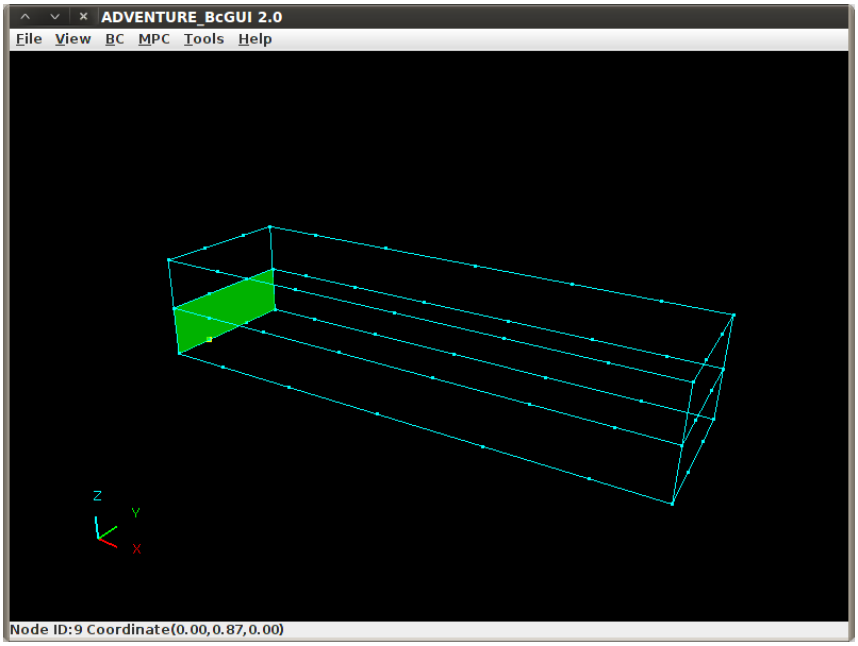


Fig. 5.2.2.6-1 How constraints on a surface group look

To clear display of constraints, select "View", "Boundary Condition" and "None" in the menu bar.

5.2.2.7 How to Set Boundary Conditions (Constraints on a Node)

As shown in Fig. 5.2.2.7-1, pick a node to apply constraint.

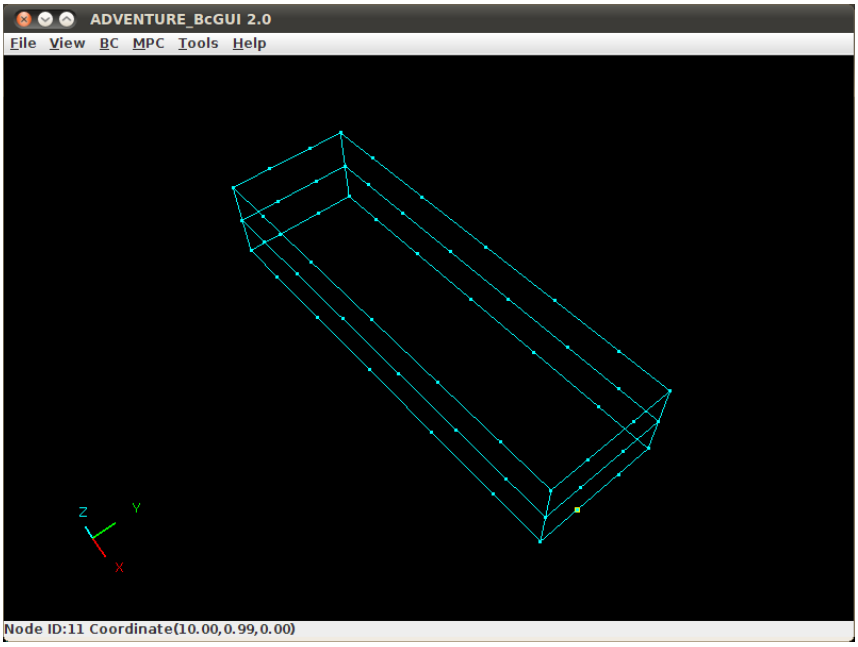


Fig. 5.2.2.7-1 How the selected node to apply constraint looks

Then choose "BC", "BC(Solid)" and "Add Displacement" in the menu bar, and a dialog for setting nodal constraints will appear.

The way and the items to set are the same as those shown for constraints on a surface group in section 5.2.2.5. How the dialog looks after values are entered is shown in Fig. 5.2.2.7-2.

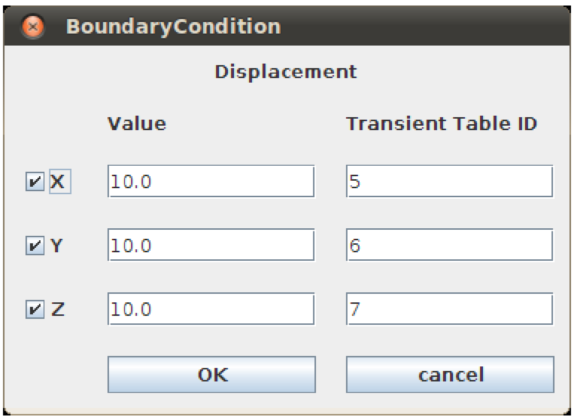


Fig. 5.2.2.7-2 A dialog for setting constraints (after values are entered)

5.2.2.8 How to Display Boundary Conditions (Checking a Node with Constraint)

Cancel selection of a node if any. Then select "View", "Boundary Condition" and "View Displacement" in the menu bar. As shown in Fig. 5.2.2.8-1, constraints are shown as a directed line originating from the node with constraints.

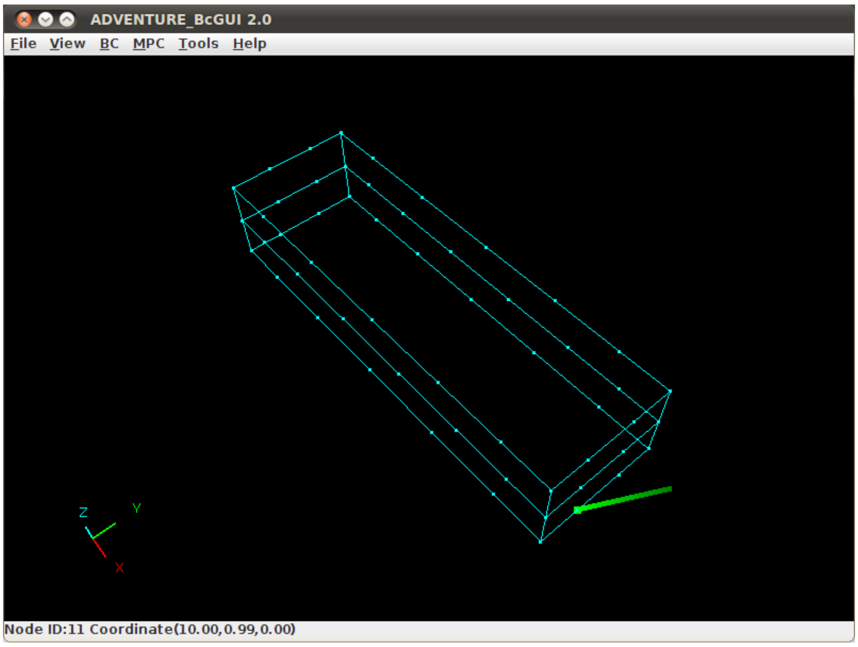


Fig. 5.2.2.8-1 How constraints on a node look

5.2.2.9 How to Set Boundary Conditions (Load on a Surface Group)

Pick a node on an edge adjacent to the surface group to be loaded as shown in Fig. 5.2.2.9-1.

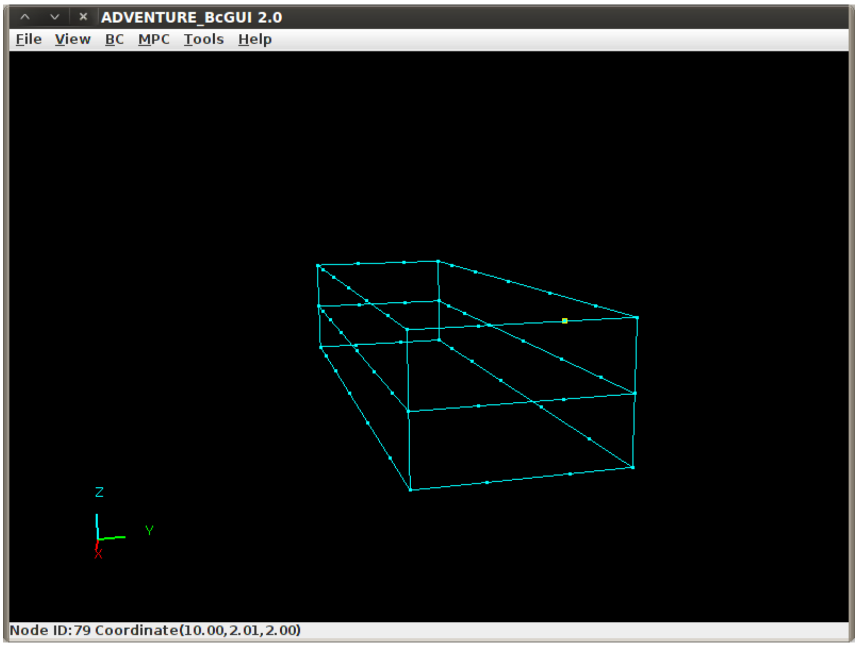


Fig. 5.2.2.9-1 Selection of a node on an edge

Then, right-click, and a surface group at the end of the part is selected (Fig. 5.2.2.9-2).

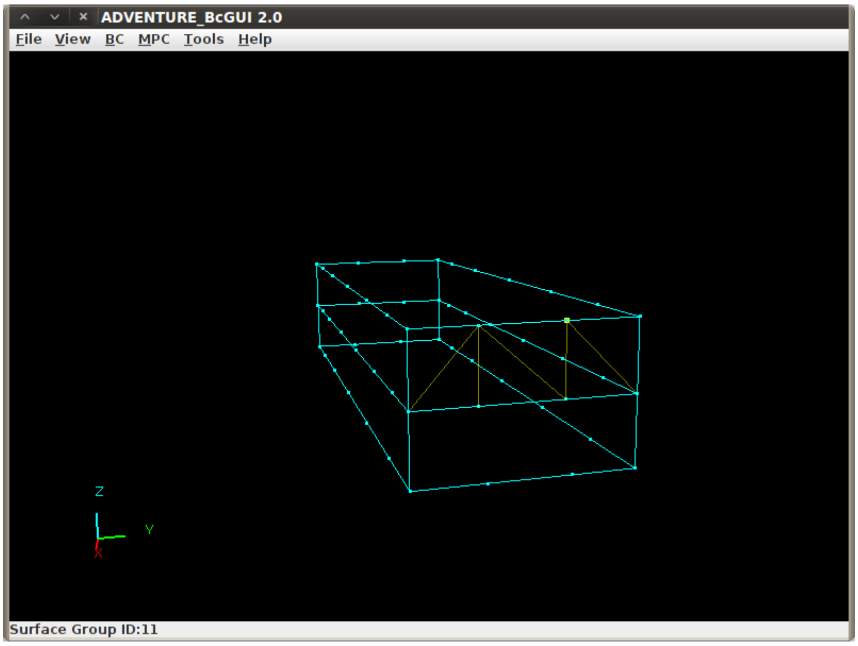


Fig. 5.2.2.9-2 How the selected end surface group to be loaded looks

Then select "BC", "BC(Solid)" and "Add SurfaceTraction" in the menu bar, and a dialog for setting a load will pop up as shown in Fig. 5.2.2.9-3.

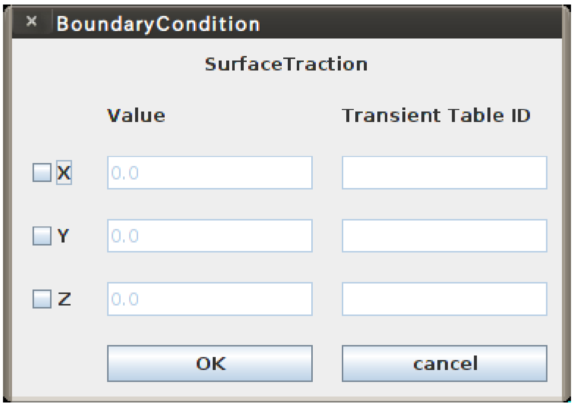


Fig. 5.2.2.9-3 A dialog for setting a load (before entering values)

Check all the directions to apply load, and enter a load component in a ”Value” field. Note that the unit of load is per unit area. Fig. 5.2.2.9-4 shows an example of load setting.

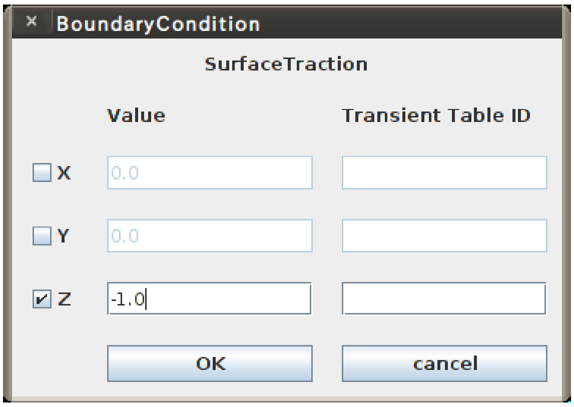


Fig. 5.2.2.9-4 A dialog for setting a load (after values are entered)

5.2.2.10 How to Display Boundary Conditions (Checking a Surface Group with a load)

Cancel selection of a surface group if any. Then select "View", "Boundary Condition" and "View SurfaceTraction" in the menu bar. As shown in Fig. 5.2.2.10-1, the surface group with a load will be highlighted in magenta.

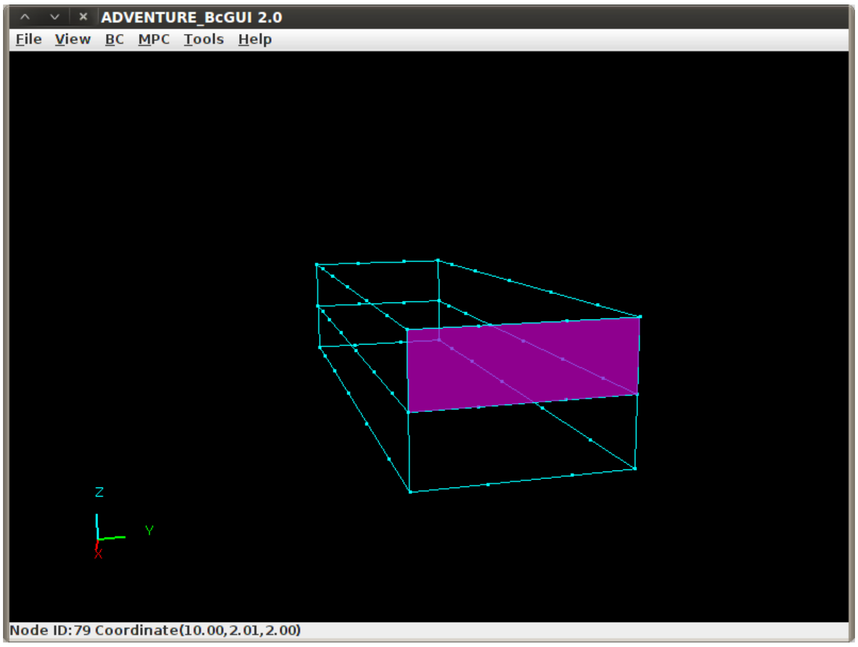


Fig. 5.2.2.10-1 How a load on a surface group looks

5.2.2.11 How to Set Boundary Conditions (Load on a Node)

As shown in Fig. 5.2.2.11-1, pick a node to apply load.

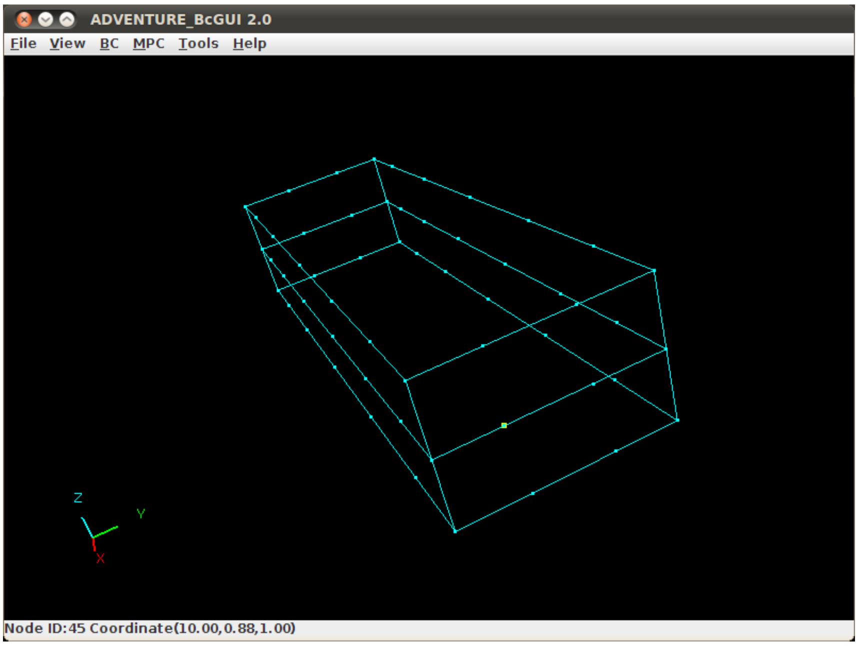


Fig. 5.2.2.11-1 How the selected node to apply load looks

Then choose "BC", "BC(Solid)" and "Add PointLoad" from the menu bar, and a dialog for setting a nodal load will appear.

The way and the items to set are the same as those shown for load on a surface group in section 5.2.2.9. How the dialog looks after values are entered is shown in Fig. 5.2.2.11-2.

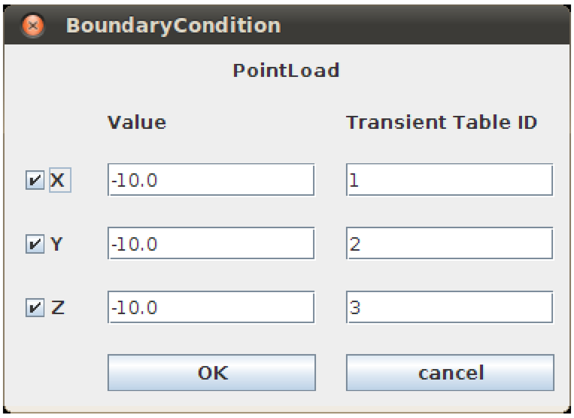


Fig. 5.2.2.11-2 A dialog for setting load (after values are entered)

5.2.2.12 How to Display Boundary Conditions (Checking a Node with load)

Cancel selection of a node if any. Then select "View", "Boundary Condition", "View PointLoad" in the menu bar. As shown in Fig. 5.2.2.12-1, a loads is shown as a directed line originating from the node with a load.

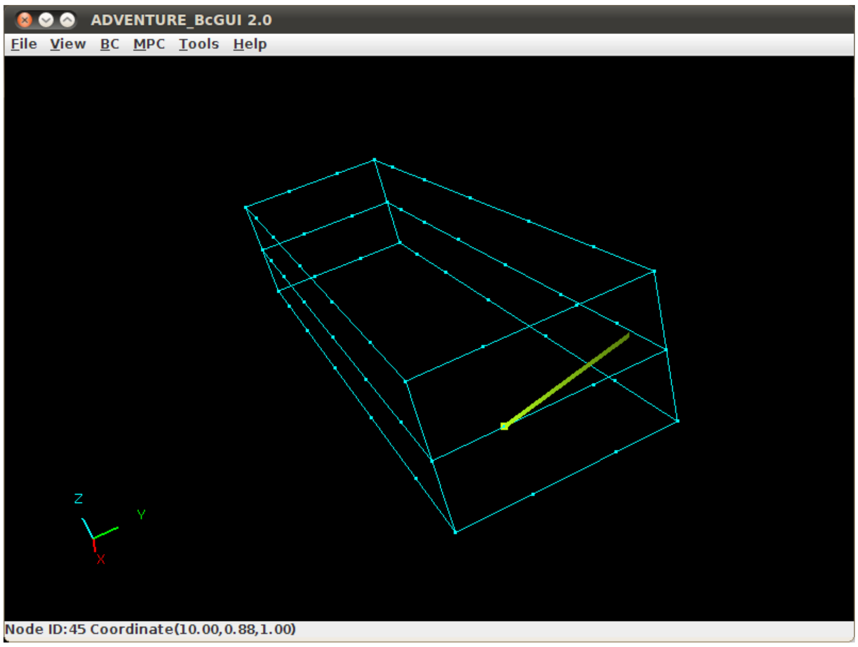


Fig. 5.2.2.12-1 How a load on a node looks

5.2.2.13 How to Set Gravity Acceleration

Select "BC" > "Gravity Acceleration" in the menu bar, and a dialog in Fig. 5.2.2.13-1 will appear.

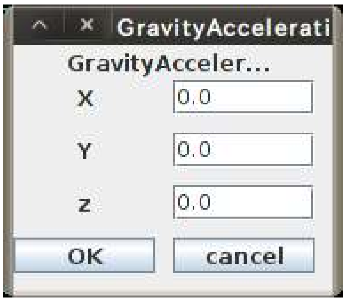


Fig. 5.2.2.13-1 A dialog for setting gravity acceleration (before entering values)

Enter each component of a gravity acceleration vector. The dialog will look like Fig. 5.2.2.13-2 for example. Then Click "OK".

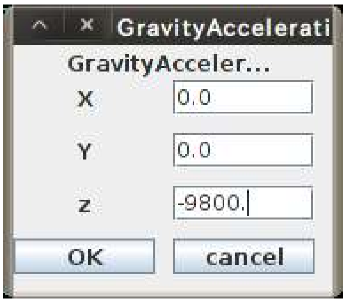


Fig. 5.2.2.13-2 A dialog for setting gravity acceleration (after values are entered)

5.2.2.14 How to Set Other Boundary Conditions

If "BC", “BC(Solid)” and “Add PointLoad(Nonlinear)” is chosen in the menu bar, a load history table for a static nonlinear analysis can be designated to a node through a time history ID.

If "BC", “BC(Solid)” and “Add Displacement(Nonlinear)” is chosen in the menu bar, a forced displacement history table for a static nonlinear analysis can be designated to a node and to a surface group through a time history ID.

If "BC", “BC(Solid)” and “Add InitialVelocity” is chosen in the menu bar, an initial velocity can be set to a node and to a surface group.

If "BC", “BC(Solid)” and "Add Velocity" is chosen in the menu bar, a velocity can be set to a node and to a surface group.

If "BC", “BC(Solid)” and "Add Acceleration" is chosen in the menu bar, an acceleration can be set to a node and to a surface group.

If "BC", “BC(Solid)” and "Add Pressure" is chosen in the menu bar, pressure can be set to a surface group.

If "BC", “BC(Solid)” and “Add SurfaceTraction(Nonlinear)” is chosen in the menu bar, a load history table for a static nonlinear analysis can be designated to a surface group through a time history ID.

How to set values is similar to the way described in detail in the preceding sections from 5.2.2.5 through 5.2.2.12.

5.2.2.15 How to Clear Boundary Conditions

At first, apply a 1.0 load in Z direction on the surface group #11 as shown in the status bar in Fig. 5.2.2.15-1. Then choose

"BC", "BC(Solid)" and "Clear SurfaceTraction" in the menu bar.

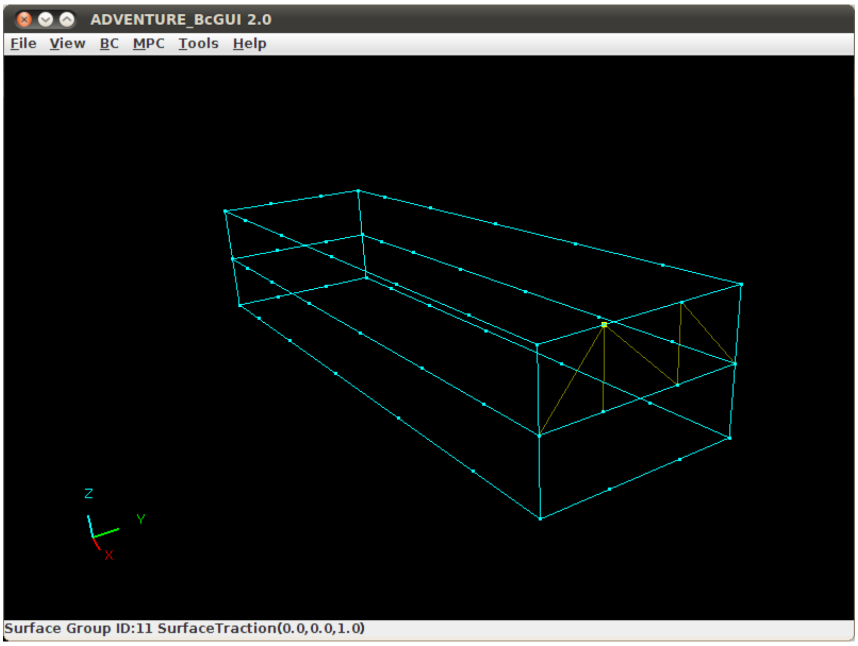


Fig. 5.2.2.15-1 The load condition applied on the surface group #11 is shown in the status bar

The dialog shown in Fig.5.2.2.15-2 will appear and require user's reconfirmation. If OK, click "OK " in the dialog.

Fig. 5.2.2.15-2 A reconfirmation dialog for clearance of surface traction (i.e. load on surface group)

If the surface group #11 is selected again, surface traction information is missing in the status bar as shown in Fig.5.2.2.15-3. If there are multiple instances of a same condition type, all of them will be cleared.

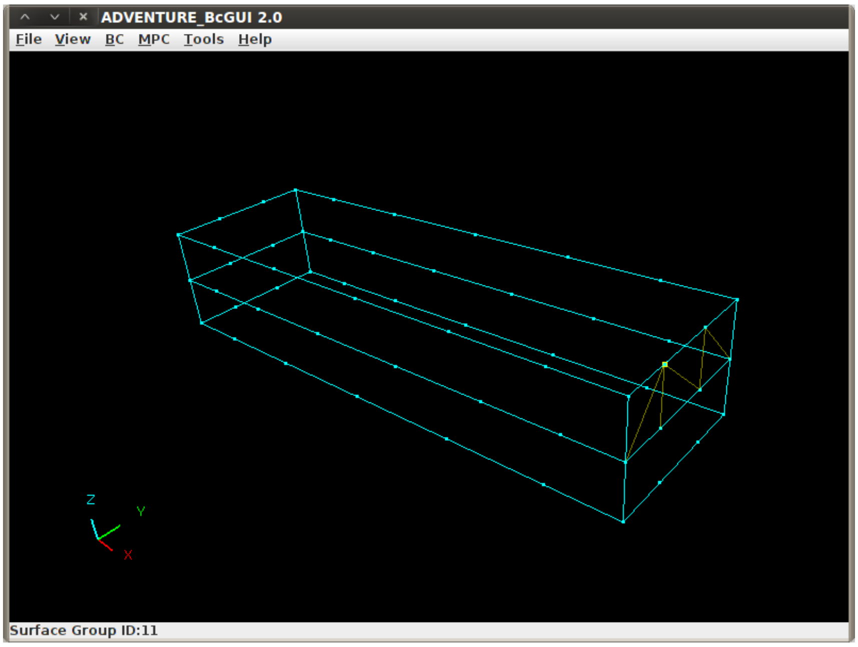


Fig. 5.2.2.15-3 How missing load conditions on surface group #11 look

Select "BC", "BC(Solid)" and "Clear PointLoad" in the menu bar. Operation similar to surface traction follows.

Select "BC", "BC(Solid)" and "Clear Displacement" in the menu bar. Operation similar to surface traction follows.

Select "BC", "BC(Solid)" and "Clear Velocity" in the menu bar. Operation similar to surface traction follows.

Select "BC", "BC(Solid)" and "Clear Acceleration" in the menu bar. Operation similar to surface traction follows.

Select "BC", "BC(Solid)" and "Clear Pressure" in the menu bar. Operation similar to surface traction follows.

5.2.2.16 How to View and Modify Constraints and Load Already Defined

Select a surface group or a node that already has conditions, then select "BC", "BC(Solid)" and "Add Displacement" or "BC", "BC(Thermal)" and "Add SurfaceTraction" in the menu bar. A dialog will pop up that contains conditions already defined. The conditions shown in the dialog can be modified and saved in memory by clicking "OK".

Also, a list of already defined boundary conditions can be displayed by selecting ”View”, “Boundary Condition” and “Cnd format” in the menu bar.

5.2.2.17 Saving Boundary Conditions

If the -ocnd option is designated at startup of the application, saving operation is not necessary.

Select "File" and "Save Condition" in the menu bar, and enter a name of the file to be saved in the file selection dialog that will appear and save it. If the extension "cnd" is missing, it will be automatically appended. If an existing file is designated, a warning will appear.

An example content of a saved file is shown in the Table 5.2.2.17-1. By this operation, all the ordinary boundary conditions set after startup of the application are saved.

Table 5.2.2.17-1 An example content of a saved boundary condition file (*.cnd)

```
gravity 0.0 0.0 0.0
boundary 4
tracOnFaceGroup 11 0 2 -1.0
Transient 1 dispOnFaceGroup 0 0 0 0.0
Transient 1 dispOnFaceGroup 0 0 1 0.0
Transient 1 dispOnFaceGroup 0 0 2 0.0
```

As for the format rules, refer to A.1.9 in the Appendix.

5.2.3 Step 3 Preparation of Material Data

Now prepare material data that will become necessary at Step 4. The multiBrick_V.pcm/multiBrick_V.pcg in the samples folder in the installation folder are used as example files.

5.2.3.1 How to Display Volumes

In the following explanation, The multiBrick_V.pcm/multiBrick_V.pcg in the samples folder in the installation folder will be used. If a user-prepared model is used, a shell script, msh2pcm will become necessary for the purpose of extracting mesh surface and inter-volume boundaries. A data flow for preparation of material data including the process of msh2pcm is shown in Fig.5.2.3.1-1.

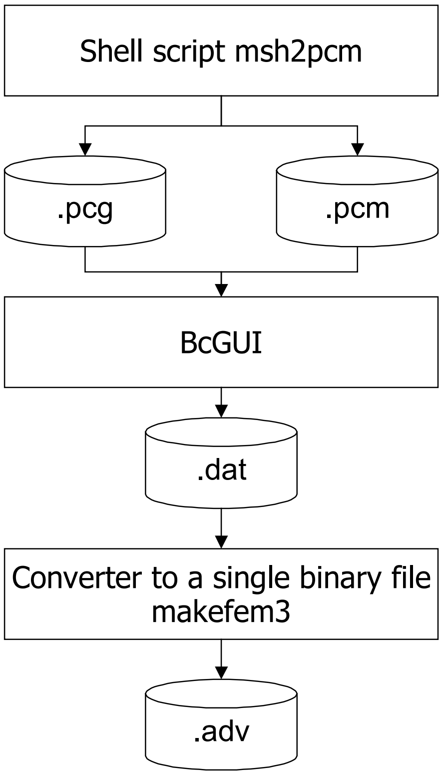


Fig. 5.2.3.1-1 The flow of material data preparation

The output files of msh2pcm are used only for the purpose of displaying volumes.

```
% msh2pcm mshFile
```

mshFile :mshFile is the name of a mesh file.

If the name of a mesh file is Model.msh, the name of the output file will be as follows.

Model_V.pcm : is the extracted surface mesh data file.

Model_V.pcg : is the surface patch group data file.

Then in order to display volumes, launch BcGUI and read files. There are two ways for doing so. Either start it up by a command line with file names given as runtime arguments or launch it through a shortcut and read files by selecting “File” and “Open File” in the menu bar. As for the way to open files through the menu, read section 5.2.2.2.

The following shows how to startup the BcGUI by a command line with file names given as runtime arguments.

```
% ADVENTURE_BcGUI_Ver_2_0 pcmFile pcgFile
```

pcmFile : is the name of an extracted surface mesh data file.

pcgFile : is the name a surface patch group data file.

BcGUI is launched in the volume display mode if the extension of the first argument is pcm (Fig.5.2.3.1-2).

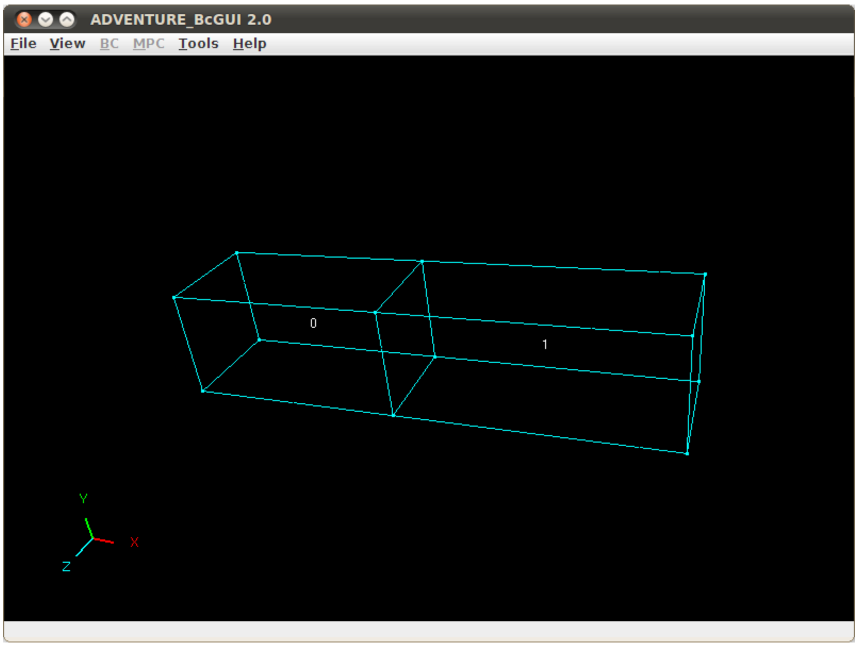


Fig. 5.2.3.1-2 The volume display mode

In the volume display mode, neither a node click nor setting of boundary conditions is possible. By pressing 'n' or 'p' on the keyboard, a volume can be selected sequentially. If 'n' is pressed, a volume with an incrementing number is selected, and if 'p' is pressed, a volume with a decrementing number is selected. The currently selected volume number will be shown in the upper left of the window (Fig. 5.2.3.1-3).

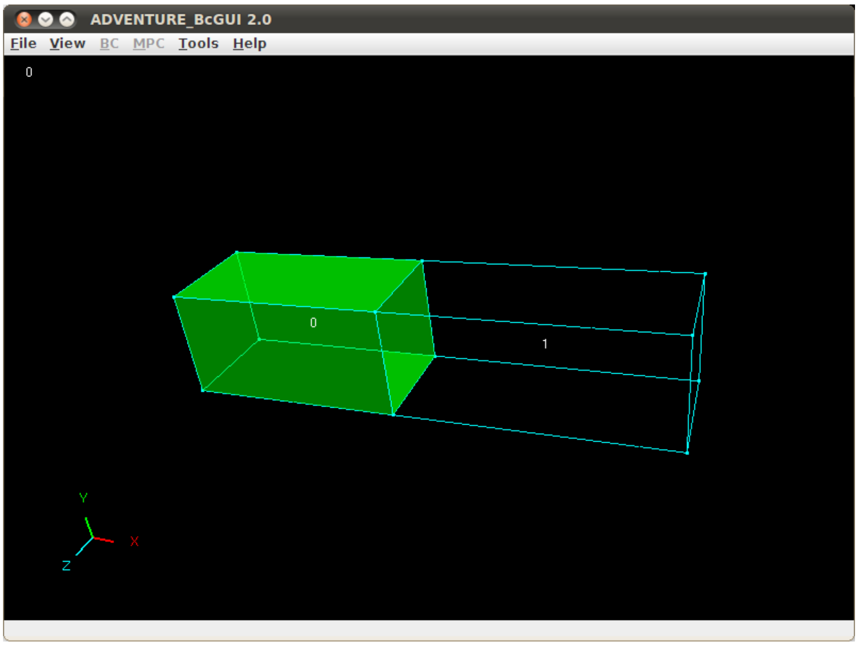


Fig. 5.2.3.1-3 How the selected volume number 0 looks

5.2.3.2 Preparation of Material Data

Launch the BcGUI in the volume display mode, then select "Tools" and "Set Material Properties" in the menu bar, and the "Set Material Properties" dialog will pop up as shown in Fig.5.2.3.2-1. Since switch of a volume number tab in the "Set Material Properties" dialog corresponds to highlighted volume in the main window, a user can enter data while visually confirming which volume is currently handled.

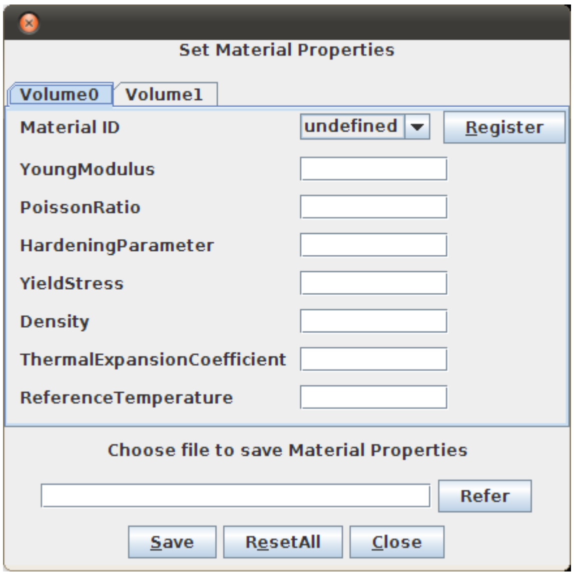


Fig.5.2.3.2-1 "Set Material Properties" dialog

The "Set Material Properties" dialog after all the necessary material data are entered is shown in Fig. 5.2.3.2-2.

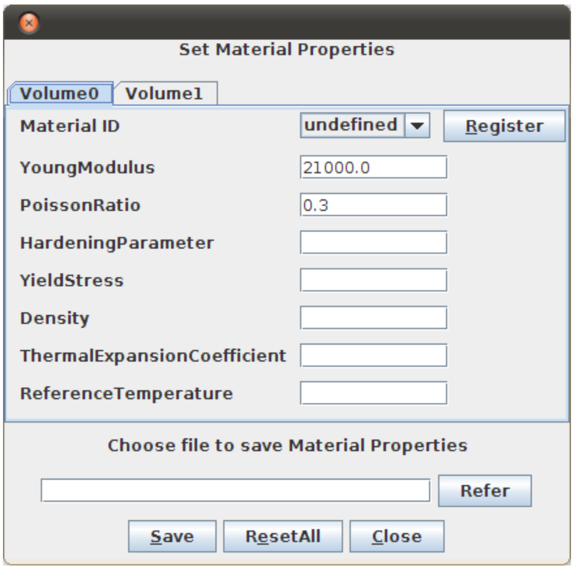


Fig. 5.2.3.2-2 How the material properties setting dialog looks after data are entered

At this stage, a material data set can be registered by clicking the "Register" button. The dialog after registration will look like Fig.5.2.3.2-3. As can be seen in the figure, the "Material ID" label has changed to "ID0".

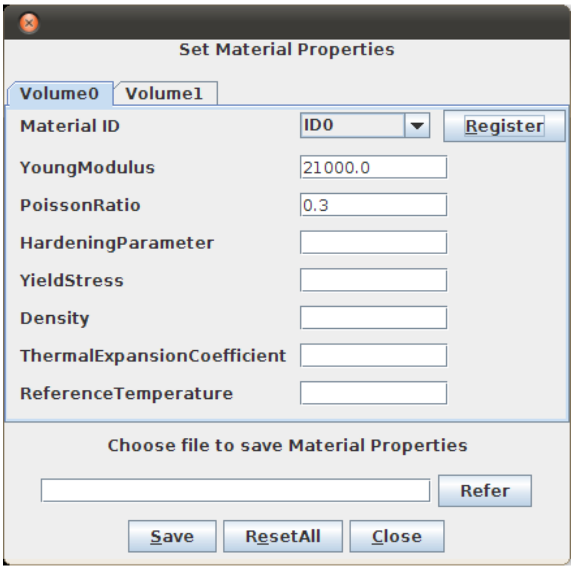


Fig. 5.2.3.2-3 How the dialog after material registration will look

Then, material data are to be set for the other volume. Click the other volume tab, and material data can be entered for the selected volume. The dialog after switching to other volume tab is shown in Fig.5.2.3.2-4.

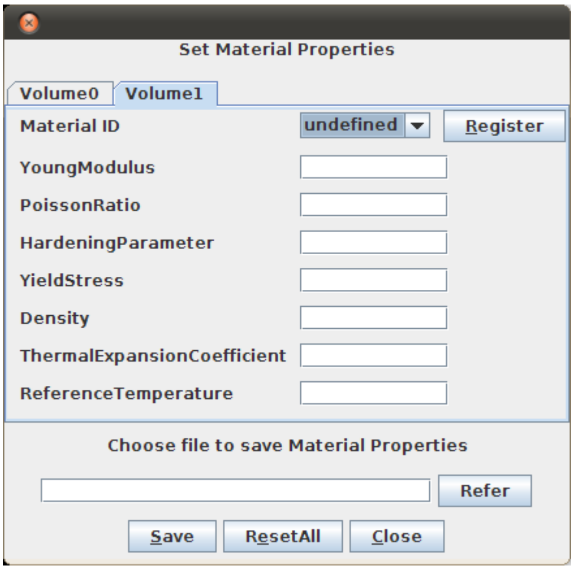


Fig. 5.2.3.2-4 "Set Material Properties" dialog right after the tab for Volume 1 is selected

Note that “Material ID” is “undefined”. If “ID 0”, which has just been registered, is selected from this pull-down menu, it is possible to set the same material data that has just been entered. If new material data are entered and the “Register” button is clicked, a different material data set can be registered.

After material data are entered for all volumes, click the “Refer” button at the bottom of the "Set Material Properties" dialog and select a path of the file to store material data (a material properties file). How the dialog looks after file selection is shown in Fig. 5.2.3.2-5.

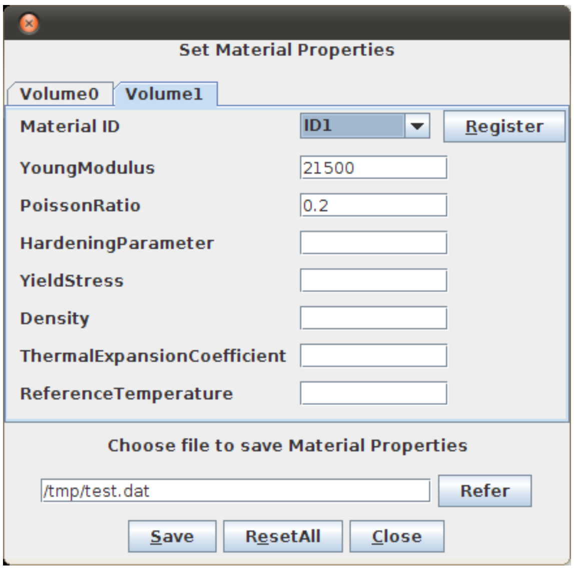


Fig. 5.2.3.2-5 The setting dialog after data storage file selection

After confirming the selected file, click the “Save” button to perform storage.

An example of generated material properties files is shown in Table 5.2.3.2-1.

Table 5.2.3.2-1 An example of generated material properties files

```
#materialInfo
materialN   2
propertyN   2
YoungModulus   21000
PoissonRatio   0.3
YoungModulus   21500
PoissonRatio   0.2
#volumeInfo
volumeN   2
0
1
```

5.2.4 Step 4 Creation of an Integrated Input File for Solid

Boundary conditions and material properties are added to mesh and and an integrated FEA model file of ADVENTURE_IO format is created.

At this step, input files and output files are as follows.

Input files:

Mesh data file	(extension is msh)
Mesh surface data file	(extension is fgr)
Analytical conditions data file	(extension is cnd)
Global index file	(extension is trn)
Material properties data file	(extension is dat)

An output file:

Integrated FEA model file	(extension is adv)

At this step, makefem3 is used. The command will output an integrated FEA model file that is compatible with ADVENTURE_Metis Ver.1, if used without any option (by default). If the command is performed with a -v2 option, a file compatible with ADVENTURE_Metis Ver.2 will be created.

The following arguments should be specified with makefem3 in the command line.

```
% makefem3 [Option] mshFile fgrFile cndFile datFile advFile [-t trnFile]
```

Input files:

mshFile	:mesh data file name
fgrFile	:mesh surface data file name
cndFile	:analytical conditions data file name
datFile	:material properties data file name
trnFile	:global index file name

The output file:

advFile	:integrated FEA model file name

Option:

-v2	:a file compatible with ADVENTURE_Metis Ver.2 is created

Arguments in the parentheses […] can be omitted.

Specify a trnFile, if the node numbers presented in a cndFile correspond to pch node numbers. The command will convert node numbers corresponding to pch node numbers to those corresponding to msh node numbers and perform data processing. If the node numbers presented in a cndFile correspond to msh node numbers, a trnFile can be omitted.

5.2.5 Step 5 Addition of Time History Data

At this step, the way to prepare time history data is described. In the case of dynamic analyses, boundary conditions can be time dependent if time history data are used. There are two ways to provide time history data either through ADVENTURE_IO format or by a CSV format that the solvers can directly reads. When the data is provided through ADVENTURE_IO format, there are two ways. One way is to provide as an advFile that only contains time history data. The other way is to append the time history data to the integrated FEA model file. This manual describes the way to prepare ADVENTURE_IO formatted data and append them to an integrated FEA model. To convert time history data files to an a2adv formatted file, a Perl script csv2adv will be used.

The following arguments should be specified with csv2adv in the command line.

```
% csv2adv.pl [option]  <output_file> <input_file> [<input_file> ...]
```

Input files:

<input_file>	:CSV formatted input file names, multiple files are possible.

An output file:

<output_file>	:An output text file created for the a2adv command

Options

-version, -v	:Version information of csv2adv is displayed.
-help, -h	:Help information is displayed.

Input files must be CSV (Comma Separated Value).

The file formats are shown in the following. When multiple input files are designated, an error occurs if a same time history ID is used for multiple files. A single file can contain multiple time history data.

```
"ID",<a time history ID>
"Time","Load"
<time0>,<load0>
…

["ID",<another time history ID>
"Time","Load"
<time0>,<load0>
…]
```

Output files are text files that can be read by the a2adv command. The output file format is shown below. Multiple time history data can be included in a single file unless there are duplicate IDs.

```
TimeHistory <The total number of time history datasets>
fega_type=Void
format=f8f8
history_id=<a time history ID>
<time0> <load0>
...

[TimeHistory <The total number of time history datasets>
fega_type=Void
format=f8f8
history_id=<a time history ID>
<time0> <load0>
…]
```

After a time history input file is converted to that with the a2adv format by the csv2adv command, it will be appended to an existing integrated input file for the Solid by means of the "add" option of the a2adv command.

```
% a2adv.pl -add <input_file> [<input_file> …] <output_file>
```

Input files

<input_file> :Input files for the a2adv, multiple files can be designated.

An output file

<output_file> :An integrated input file for the Solid.

5.3 How to Set MPC Conditions between Multiple Volumes

5.3.1 MpcMasterSlaveTool

The MpcMasterSlaveTool is a tool for automatically generating fixed MPC conditions between contacting multiple volumes. The data flow is shown in Fig. 5.1-2. For its usage, refer to the chapters 1, 2, 5, and 6 in the Part III and to the section A.1.12 in the Appendix.

5.3.2 MpcLocal2Global

The MpcLocal2Global is a tool for manually generating various MPC conditions among multiple volumes. In order to use the MpcLocal2Global, a multiple-volume mesh file is necessary, which is generated by means of the PcmMerge. For the usage of the PcmMerge, , refer to the chapters 1, 2, 3, and 4 in the Part II and to the chapter A.1 in the Appendix. The interactive processing with the BcGUI described in the section 5.5.3 as well as the use of the MpcLocal2Global enables a user to manually generate MPC conditions. The data flow about this method is shown in Fig. 5.1-3. For the usage of the MpcLocal2Global, refer to the chapters 1, 2, 3, 4, and 5 in the Part V and to the chapter A.1 in the Appendix.

5.4 How to Set Boundary Conditions for Thermal Analyses

In the case of setting conditions for thermal analyses, the BcGUI is used for boundary conditions for thermal analyses, and a text editor is used to create material properties data.

The overall flow is comprised of four steps:

(1)Surface extraction of a volume mesh

(2)Boundary condition setting by the BcGUI

(3)Preparation of material properties data with a text editor

(4)Creation of an integrated input file for the Thermal module

5.4.1 Step 1 Extraction of the Mesh Surface

For how to extract the surface of mesh, refer to the section 5.2.1 "Step 1 Extraction of Mesh Surface".

5.4.2 Step 2 How to Set Boundary Conditions for Thermal Analysis by BcGUI

To set boundary conditions, The BcGUI, a GUI based tool, is used. Input files and an output file are as follows.

Input files:

Extracted surface mesh data file	(extension: .pch)
Surface patch group data file	(extension: .pcg)

Output file:

Analysis conditions file	(extension: .cnd)

For details on how to use the BcGUI, refer to sections 5.2.2.2 to 5.2.2.4. Also, regarding how to check the boundary conditions and how to save the content, refer to sections 5.2.2.16 and t.2.2.17.

5.4.2.1 How to Set Boundary Conditions (Those on a Surface Group for Thermal Analysis)

As shown in Fig. 5.4.2.1-1, select a surface group that is subject to the boundary conditions for thermal analysis. Then, select the "BC" > ”BC(Thermal)” > "Add Temperature" in the menu, and the dialog shown in Fig. 5.4.2.1-2 will be displayed. By using this dialog, temperature specified boundary conditions can be set on the surface group. The time history data is ignored in thermal analysis, therefore even if a value is entered to "Transient Table ID", it will in fact never be used in analysis.

Select the "OK" button after entering the temperature conditions, and then the conditions are actually attached to the surface group.

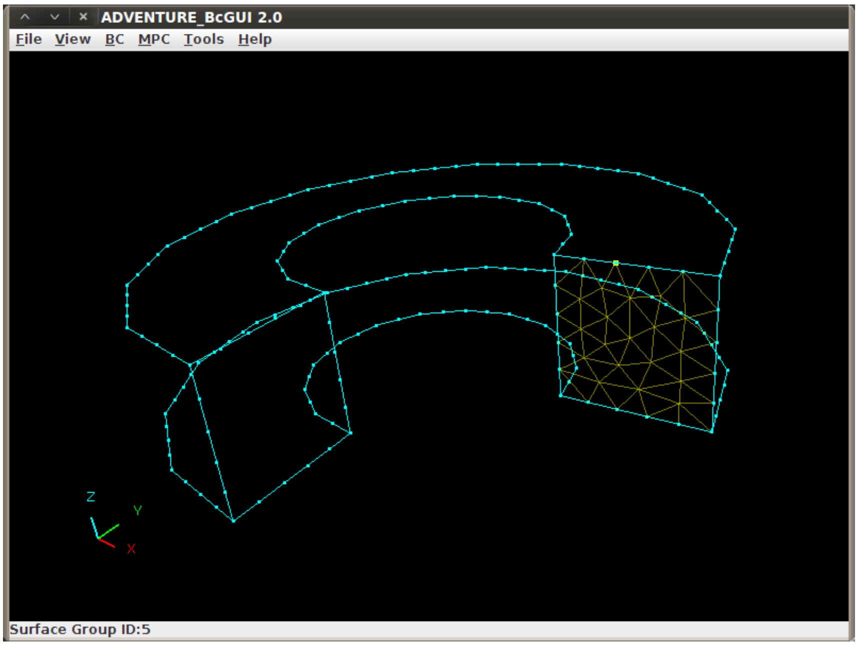


Fig. 5.4.2.1-1 Surface group selection

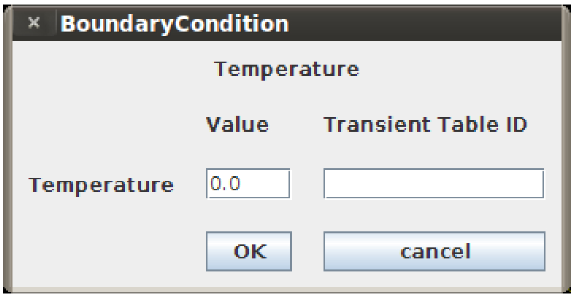


Fig. 5.4.2.1-2 The dialog for setting a boundary condition on the surface group

5.4.2.2 How to Check the Boundary Conditions (Confirmation of a Surface Group Subjected to Boundary Conditions for Thermal Analysis)

After deselecting the surface group, select the "View"> "Boundary Condition"> "View Temperature" in the menu. The surface group that is given temperature specified boundary conditions is highlighted in red as shown in Fig. 5.4.2.2-1.

To turn off the display of the temperature specified boundary conditions, select the "View"> "Boundary Condition"> "None" in the menu.

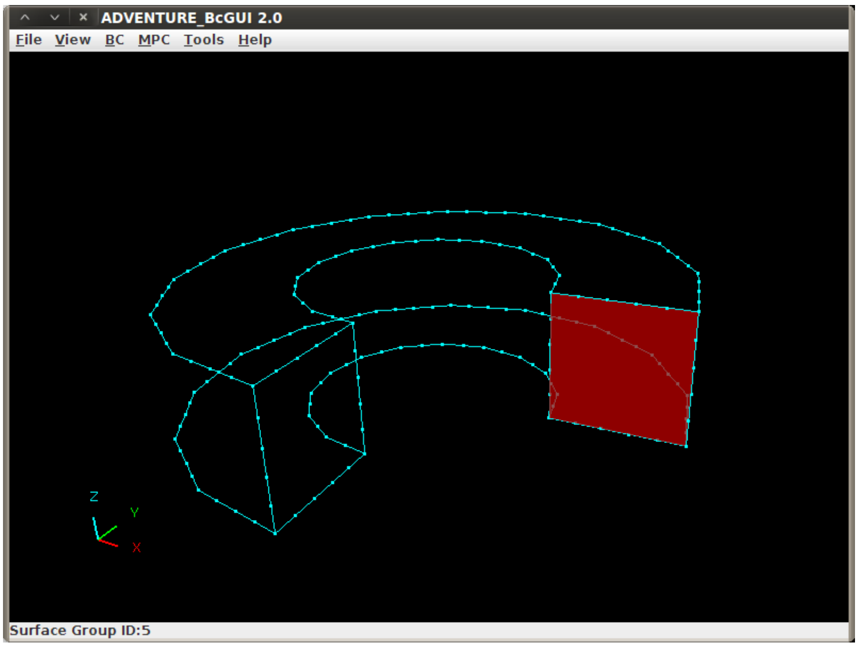


Fig. 5.4.2.2-1 Screen where a surface group given a temperature specified boundary condition is highlighted

5.4.2.3 How to Set Boundary Conditions (Those on a Node for Thermal Analysis)

As shown in Fig. 5.4.2.3-1, select the node for which you want to set the boundary conditions for thermal analysis. Then, select the "BC" > ”BC(Thermal)” > "Add Temperature" in the menu, and the dialog shown in Fig. 5.4.2.3-2 will be displayed. By using this dialog, temperature specified boundary conditions can be set on the node. Since the time history data is ignored in thermal analysis, even if a value is entered to "Transient Table ID", it will in fact never be used in analysis.

Select the "OK" button after entering the temperature conditions, and then the conditions are actually attached to the node.

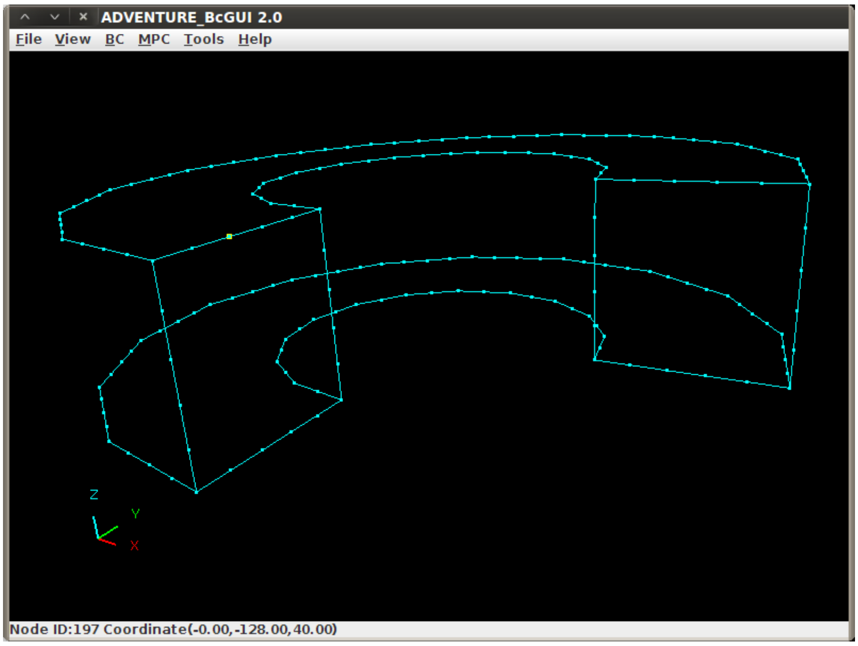


Fig. 5.4.2.3-1 Node selection

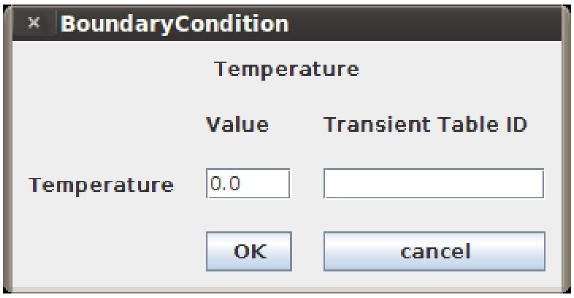


Fig. 5.4.2.3-2 The dialog for setting a boundary condition on a node

5.4.2.4 How to Check the Boundary Conditions (Confirmation of a Node Subjected to Boundary Conditions for Thermal Analysis)

After deselecting the node, select the "View"> "Boundary Condition"> "View Temperature" in the menu. The node that is given a temperature specified boundary condition is highlighted in red as shown in Fig. 5.4.2.4-1.

To turn off the display of the temperature specified boundary conditions, select the "View"> "Boundary Condition"> "None" in the menu.

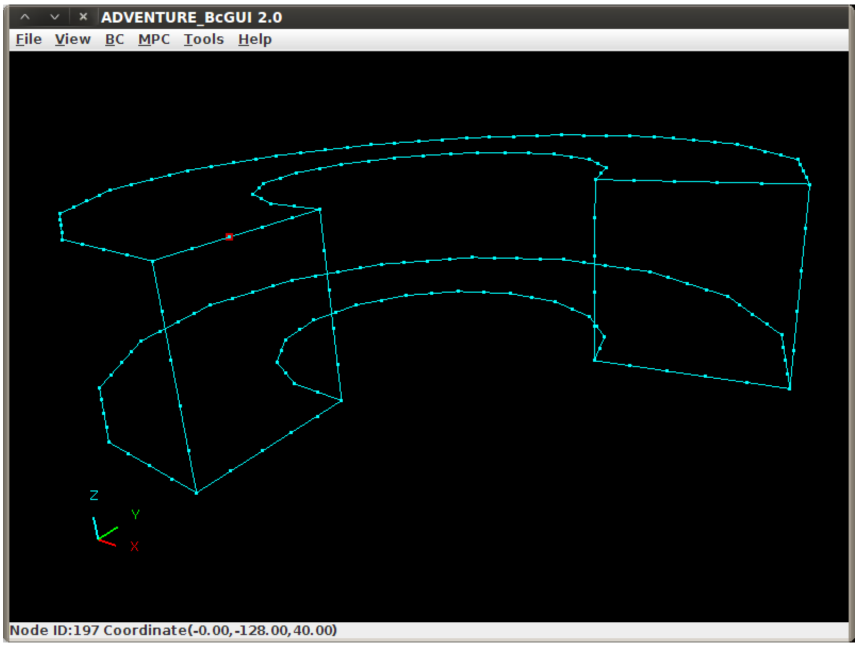


Fig. 5.4.2.4-1 Screen where a node given a temperature specified boundary condition is highlighted

5.4.2.5 Other Boundary Conditions for Thermal Analysis

A heat flux boundary condition can be attached to a surface group by selecting "BC" > “BC(Thermal)” > “Add HeatFlux” in the menu.

A heat transfer boundary condition can be attached to a surface group by selecting "BC" > “BC(Thermal)”>“Add HeatRadiation” in the menu

A heat radiation boundary condition can be attached to a surface group by selecting "BC" > “BC(Thermal)” > “Add HeatRadiation”

The setting method is similar to the one up to now.

5.4.2.6 Clearing Boundary Conditions

Set a temperature specified boundary condition on the surface group number 5 first as shown in the status bar of Fig. 5.4.2.6-1, then select

“BC” > "BC(Thermal)" > "Clear Temperature" .

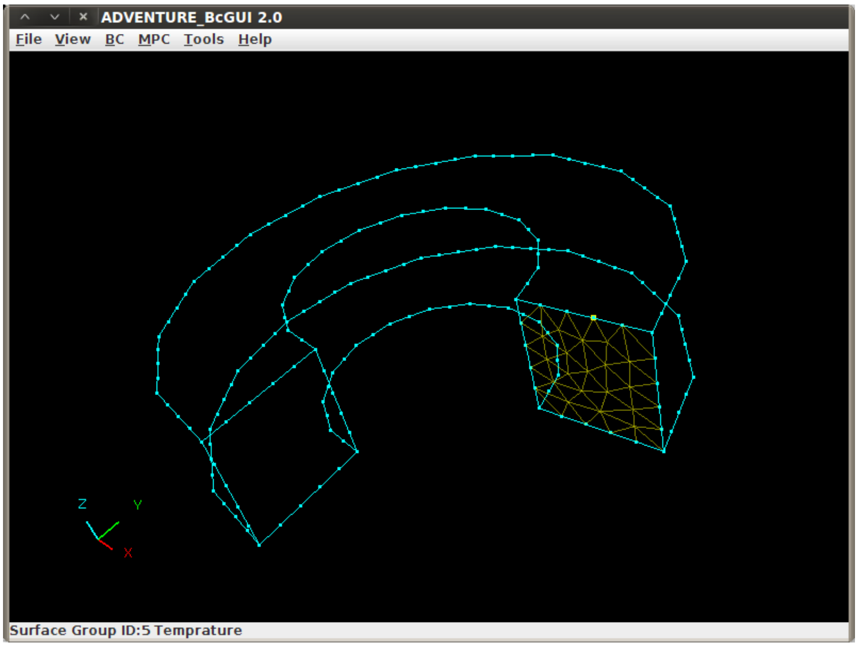


Fig. 5.4.2.6-1 The temperature specified boundary condition of surface group 5 being displayed in the status bar

Then in the dialog shown in Fig. 5.4.2.6-2, the application will ask for re-confirmation. Click on the "OK" if it is OK to perform deletion.

Fig. 5.4.2.6-2 Confirmation dialog for deletion of a temperature specified boundary condition

If the surface group number 5 is selected again, display of temperature specified boundary condition will have disappeared as shown in Fig. 5.4.2.6-3 from the status bar. If conditions of a same type were created, all of them will be deleted.

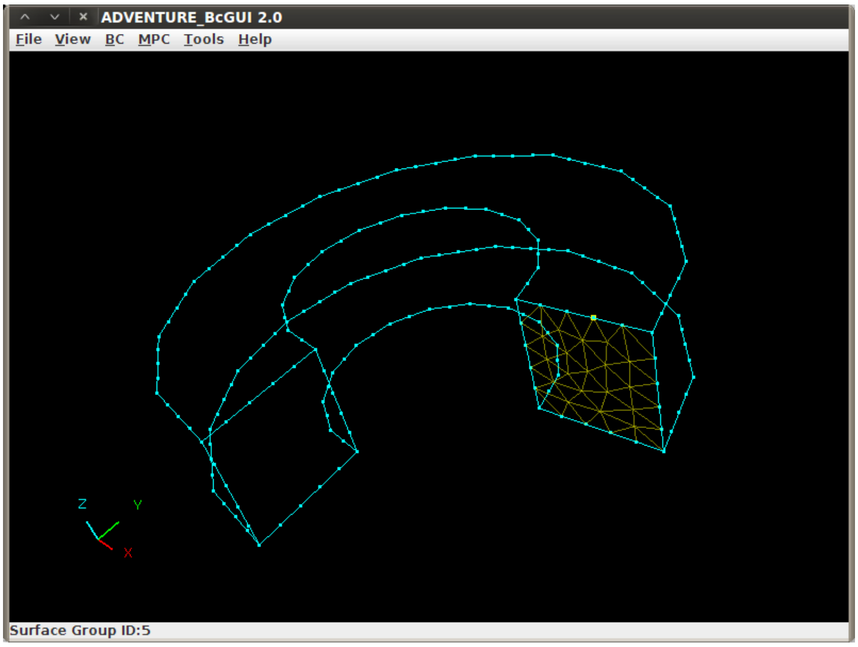


Fig. 5.4.2.6-3 How the status bar looks when the temperature specified boundary condition on the surface group # 5 is erased

"BC" > "BC(Thermal)" > "Clear HeatFlux"
The same is true as in the case of the temperature specified boundary conditions.

"BC" > "BC(Thermal)" > "Clear HeatTransfer"
The same is true as in the case of the temperature specified boundary conditions.

"BC" > "BC(Thermal)" > "Clear HeatRadiation"
The same is true as in the case of the temperature specified boundary conditions.

5.4.3 Step 3 Creation of Material Data by a Text Editor

Material data of thermal analysis will be created by means of a text editor. For material properties that can be specified and for details of the format, please refer to the Appendix Section A.1.7.

5.4.4 Step 4 Creating an Integrated Input File for the Thermal

For information about creating an integrated input file for the Thermal, please refer to the Section 5.2.4 "Step 4 Creation of an Integrated Input File for Solid".

5.5 Other Features of the BcGUI

5.5.1 Reading of boundary conditions

"File" > "Read Condition"

This is the ability to read an existing file of the normal boundary conditions after reading a model. In the file selection dialog, choose a boundary condition file (*.cnd) conforming to the model that is currently shown. No particular change occurs on the screen. In the manner described in Section 5.3.2, check the surface groups with constraints. They should look the same as Fig. 5.3-5. If directions and values should be checked as well as surface groups, check in the manner described in Section 5.3.5. Please check in the manner described in Section 5.3.4 and Section 5.3.5, when load conditions are to be confirmed.

"File" > "Quit"

This is a function to terminate this application. A dialog of reconfirmation is displayed. Click on the "OK" to proceed.

5.5.2 View

"View" > "Wire"

This is a function for switching the display mode to the wire frame mode, the surface mesh mode, and the painted mode. Default at start-up, is the wire frame mode.

"View" > "Patch"

This is a function to display the surface mesh (patch). Since the backside is also displayed, it is hard to see a complex shaped model. Refer to Fig. 5.5.2-1.

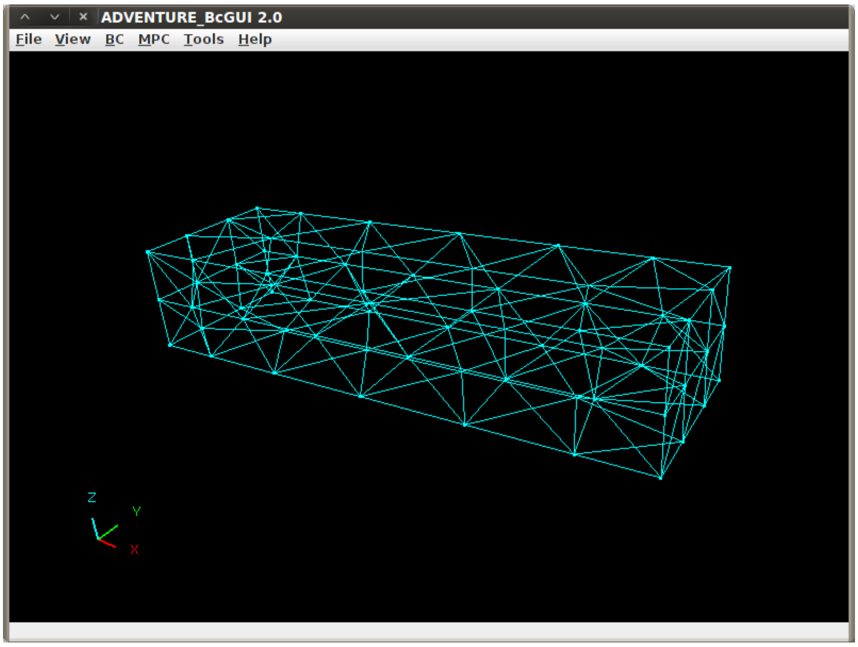


Fig. 5.5.2-1 How the mesh display mode looks

"View" > "Surface"

This is a function to draw a model in the painted mode. It hides the inside. Refer to Fig. 5.5.2-2.

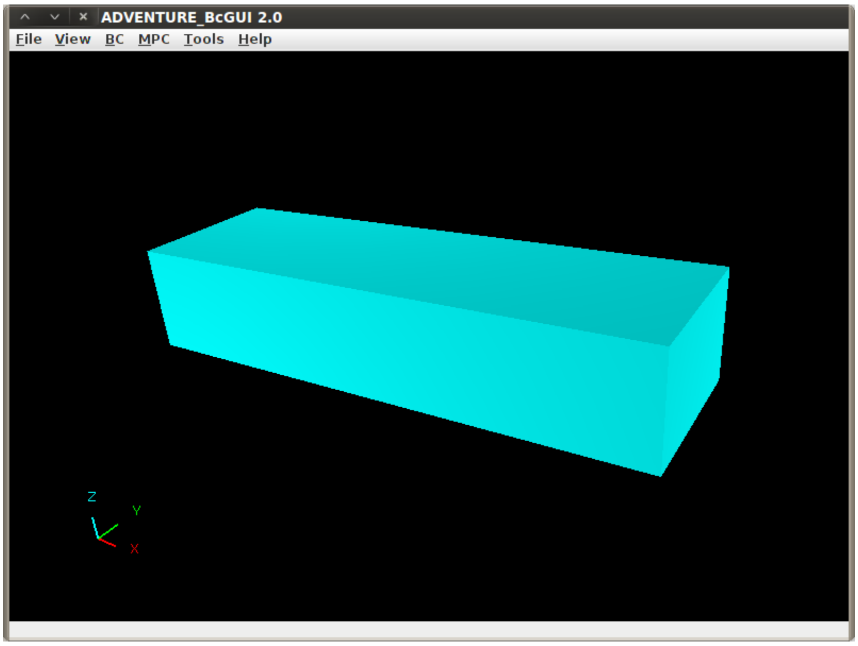


Fig. 5.5.2-2 How the painted mode looks

"View" > "Selected Surface" > "Patch"

The selected surface group will be displayed in the surface mesh mode. Either the painted mode or this mode can be chosen. This is the default mode. Refer to Fig. 5.5.2-3.

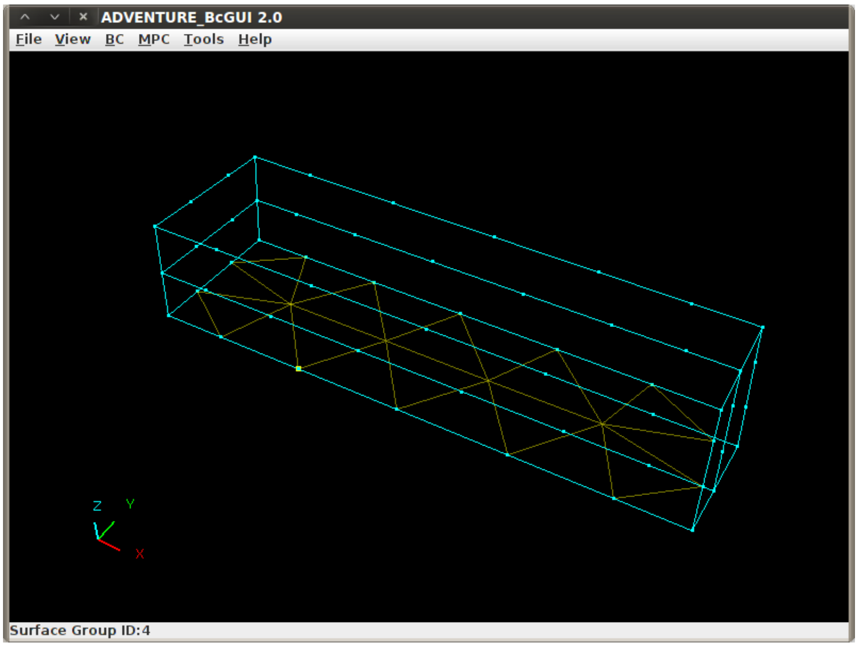


Fig. 5.5.2-3 How the selected surface looks in the surface mesh mode

"View" > "Selected Surface" > "Surface"

The selected surface group will be displayed in the painted mode. Refer to Fig. 5.5.2-4.

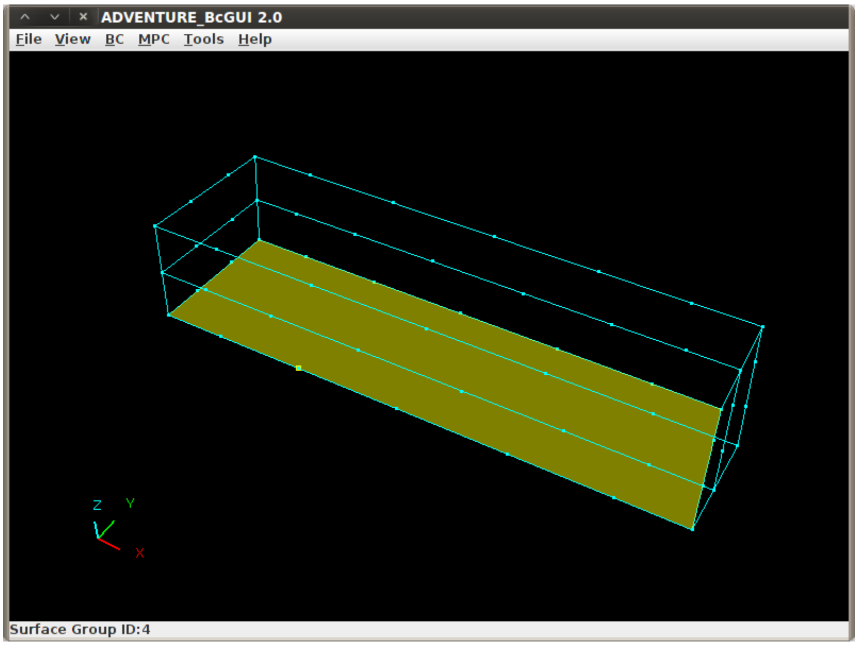


Fig. 5.5.2-4 How the selected surface group looks in the painted mode

"View" > "Projection" > "Perspective"

The perspective projection will be selected as projection method. Either the perspective projection or the orthogonal projection can be selected. Default is the perspective projection.

"View" > "Projection" > "Orthographic"

The positive projection will be selected as projection method. As shown in Fig. 5.5.2-5, this is a projection method with no sense of perspective. It is useful, for example, to compare the dimensions of different surface groups.

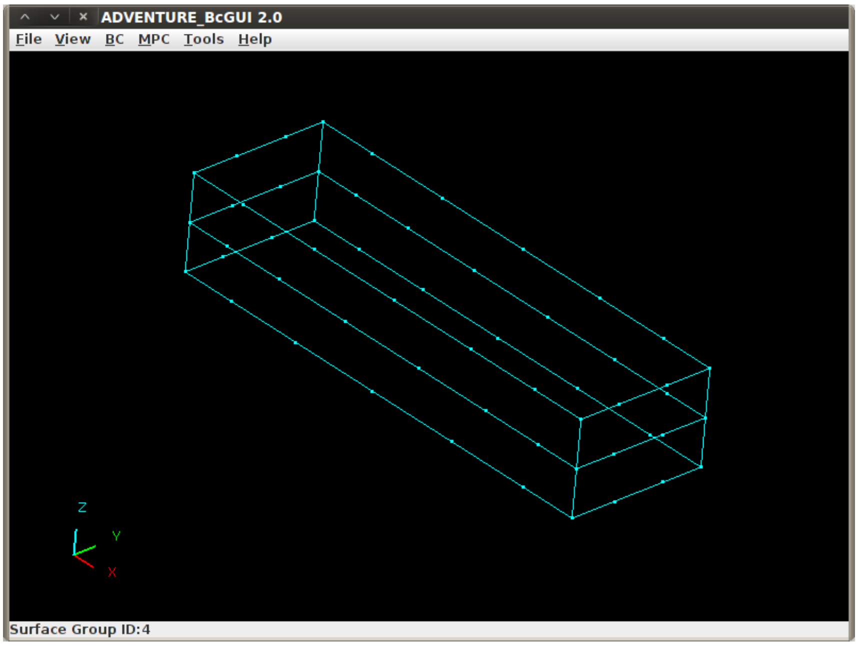


Fig. 5.5.2-5 How a model looks in the orthogonal projection

"View" > "Reset"

The point of view will be reset to the default position, namely the Z-axis positive side will be returned to the direction from the screen toward the user. The perspective/orthogonal projection mode is not reset and remains in the selected state.

"View" > "Exploded"

It is possible to make use of this feature if a model is made of multi-volumes and the nodes on the boundary between volumes are separate for each volume. Such a model is often used to set the MPC conditions. In many cases, MPC conditions are set to a pair of surface groups on the inter-volume boundary or to a node pair on these surface groups. However, the surface groups are often in contact or too close to each other to pick a surface group or a node. This function is intended to make these surface groups and nodes pickable by displaying each volume more separate than actual.

Dialog shown in Fig. 5.5.2-6 will be displayed. The volumes look more separate when the slider knob is mouse-dragged to the right, and look closer appear when dragged to the left.

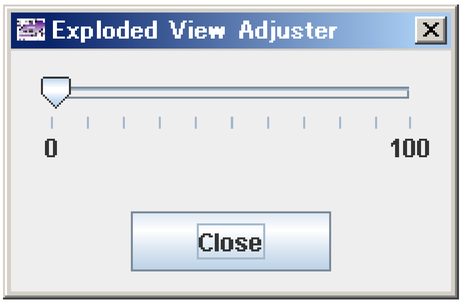


Fig. 5.5.2-6 Separation adjustment dialog for a multi-volume model with separate inter-volume boundary.

5.5.3 Setting of the MPC Conditions

"MPC" > "Rigid Beam I" > "Surface Group to Surface Group"

This is a capability to specify a surface group pair for Rigid Beam I. The dialog shown in Fig. 5.5.3-1 will be displayed.

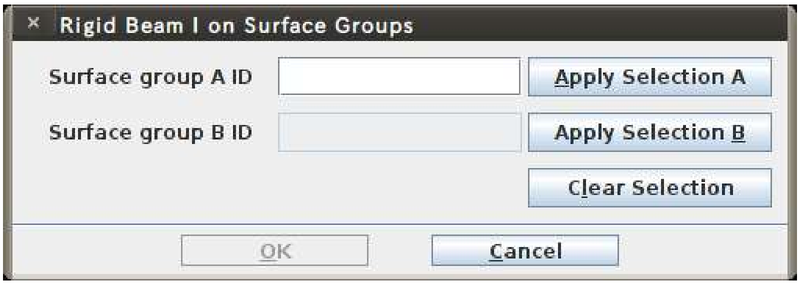


Fig. 5.5.3-1 The dialog for setting a Rigid Beam I between a surface group pair (before selecting surface groups)

From now, a surface group pair will be selected. Let's call each surface group A and B. There is no difference between the A and B. The first selected surface group will just become A. First, surface group A will be specified.

Typically, the two surface groups are in contact with each other. A surface group cannot be directly clicked. First of all, select a point belonging to the surface group by left clicking. For more explanation see Section 5.2.2.4. Display multiple volumes in a separate manner ahead if necessary (refer to Section 5.5.2).

Figure 5.5.3-2 shows how a surface group A is selected.

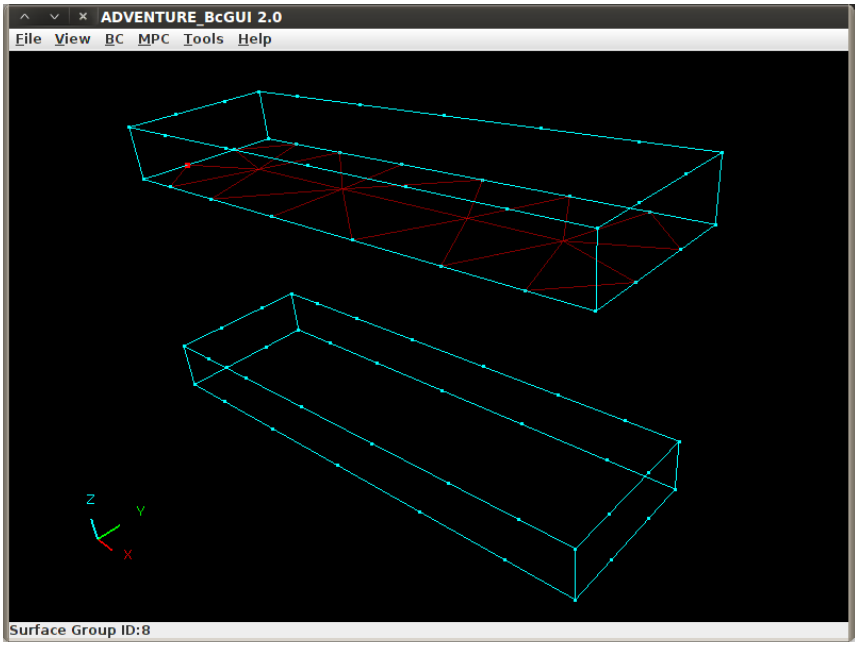


Fig. 5.5.3-2 How selection of a surface group A looks

Then, the ID of the selected surface group A will be shown in the dialog in Fig. 5.5.3-1 (refer to Fig. 5.5.3-3 for updated display).

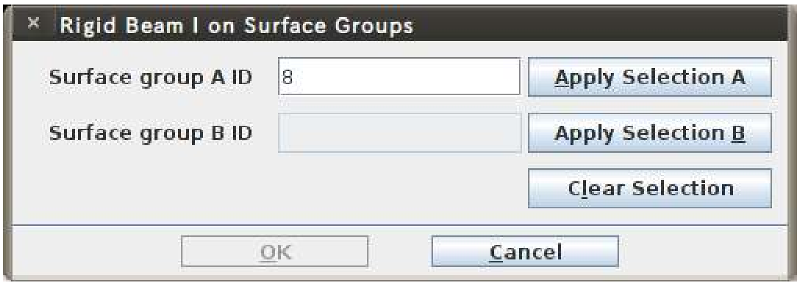


Fig. 5.5.3-3 The dialog for setting a Rigid Beam I between two surface groups

(After selection of surface group A)

By clicking on the "Apply Selection A" button in the dialog box shown in Fig. 5.5.3-3, the selection of the surface group A will become definite. Then, select a surface group B. Figure 5.5.3-4 shows how selection of a face group B looks.

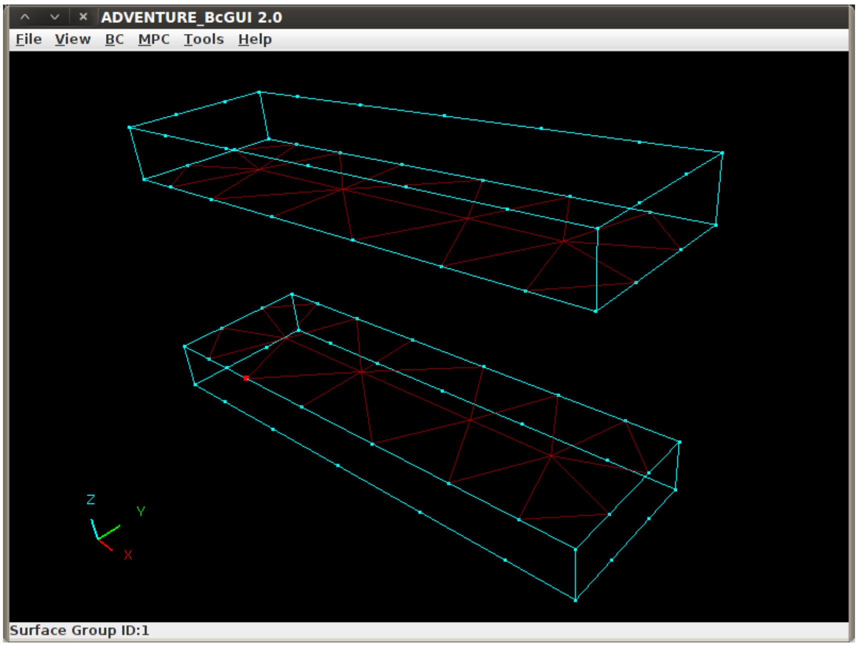


Fig. 5.5.3-4 How selection of a surface group B as well as A looks

Similarly, the ID of the surface Group B will be displayed in the dialog shown in Fig. 5.5.3-3. Click the "Apply Selection B" button and selection of the face Group B will become definite and the "OK" button is ready for a click. Clicking the "OK" button will complete the setting.

"MPC" > "Rigid Beam II" > "Node to Node"

This can define a Rigid Beam II between a node pair. The dialog in Fig. 5.5.3-5 will be displayed. Since the change of operation from the Rigid Beam I to this beam is only that nodes are selected for this beam whereas surface groups are selected for Rigid Beam I, refer to Section 5.2.2.4 for description of how to select a node inside a surface group.

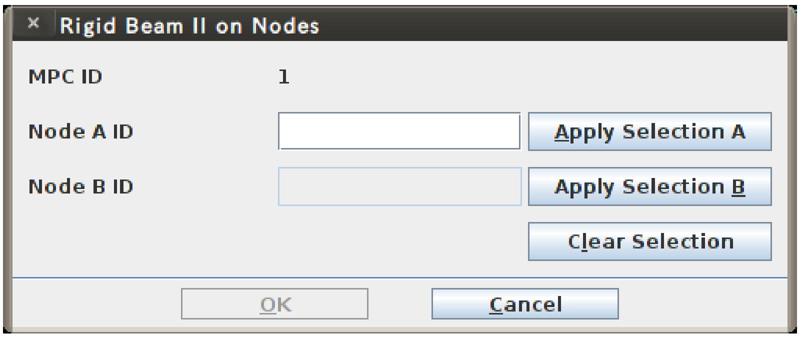


Fig. 5.5.3-5 The dialog for defining a Rigid Beam II between a node pair

(Before node selection)

"MPC" > " Rigid Beam II " > "Surface Group to Surface Group"

This can define a Rigid Beam II between a surface group pair. The dialog in Fig. 5.5.3-6 will be displayed.

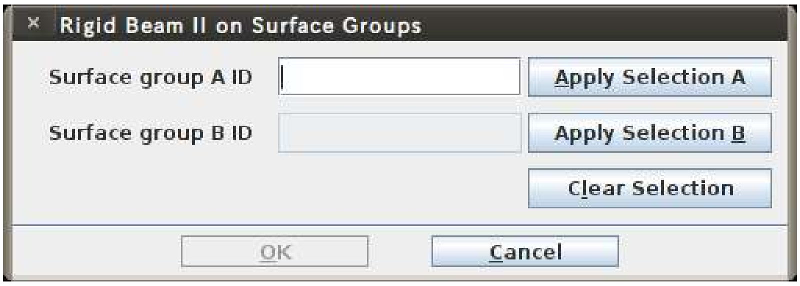


Fig. 5.5.3-6 The dialog for defining a Rigid Beam II between a surface group pair
(Before surface group selection)

Because operation is the same as for the Rigid Beam I between a surface group pair, description will be omitted.

"MPC" > "Rigid Beam III" > "Node to Node"

This can define a Rigid Beam III between a node pair. The dialog shown in Fig. 5.5.3-7 will be displayed. Since the change of operation from the Rigid Beam I to this beam is only that nodes are selected for this beam whereas surface groups are selected for Rigid Beam I, refer to Section 5.2.2.4 for description of how to select a node inside a surface group.

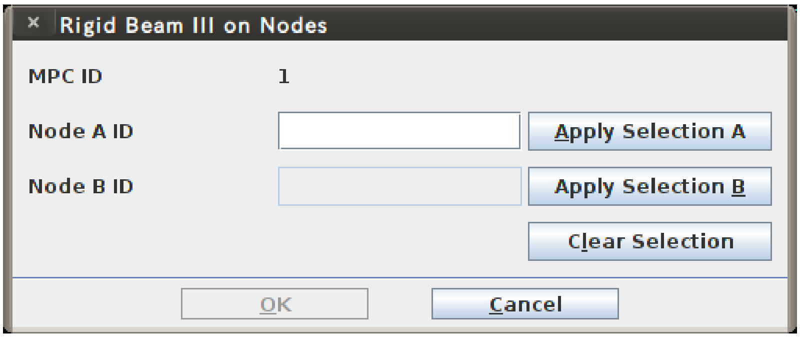


Fig. 5.5.3-7 The dialog for defining a Rigid Beam III between a node pair
(Before node selection)

"MPC" > " Rigid Beam III " > "Surface Group to Surface Group"

This can define a Rigid Beam III between a surface group pair. The dialog shown in Fig. 5.5.3-8 will be displayed. Because operation is the same as for the Rigid Beam I between a surface group pair, description will be omitted.

Fig. 5.5.3-8 The dialog for defining a Rigid Beam III between a surface group pair
(Before surface group selection)

"MPC" > "Rigid Beam IV" > "Node to Node"

This can define a Rigid Beam IV between a node pair. The dialog shown in Fig. 5.5.3-9 will be displayed.

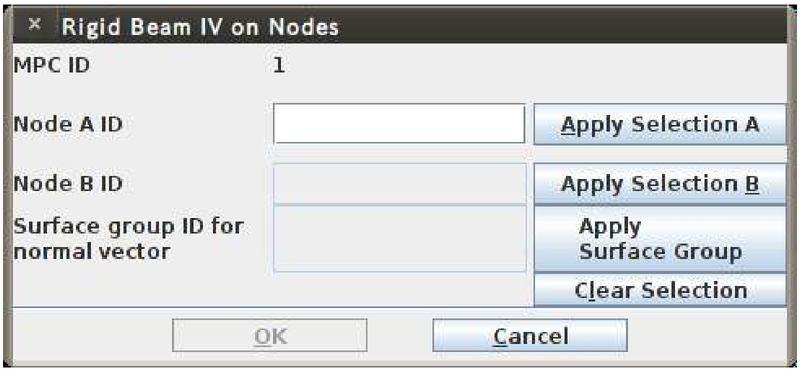


Fig. 5.5.3-9 The dialog for defining a Rigid Beam IV between a node pair
(Before node selection)

Set a node A and a node B by using one of the methods shown in Section 5.2.2.4.

"MPC" > "Rigid Beam IV " > "Surface Group to Surface Group"

This can define a Rigid Beam IV between a surface group pair. The dialog shown in Fig. 5.5.3-10 will be displayed.

Fig. 5.5.3-10 The dialog for defining a Rigid Beam IV between a surface group pair
(Before surface group selection)

Set a surface group A and a surface group B in the same operation as in the case of the Rigid Beam I between a surface group pair. Select by left click a node that belongs to the target surface group, and then change a selected surface group by right clicking until the target surface group is selected.

"MPC" > "Rigid Beam V" > "Node to Node"

This can define a Rigid Beam V that comprises of a virtual node and a connecting node. The dialog shown in Fig. 5.5.3-11 will be displayed.

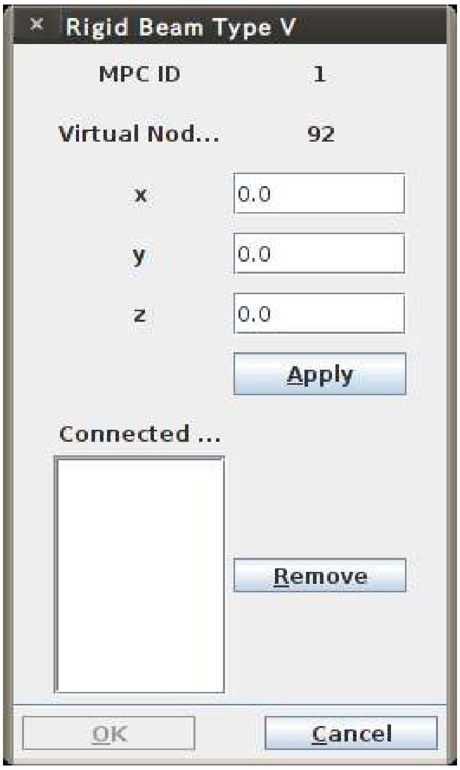


Fig. 5.5.3-11 The dialog for defining a Rigid Beam V
(Before setting)

Enter values of the xyz coordinates of the virtual node since the initial values of xyz have been set to 0.0s. The defined virtual node will appear in the 3D screen if the "Apply" button is pressed. An ID of the virtual node is obtained by adding 1 to the maximum node ID in the currently loaded pch file.

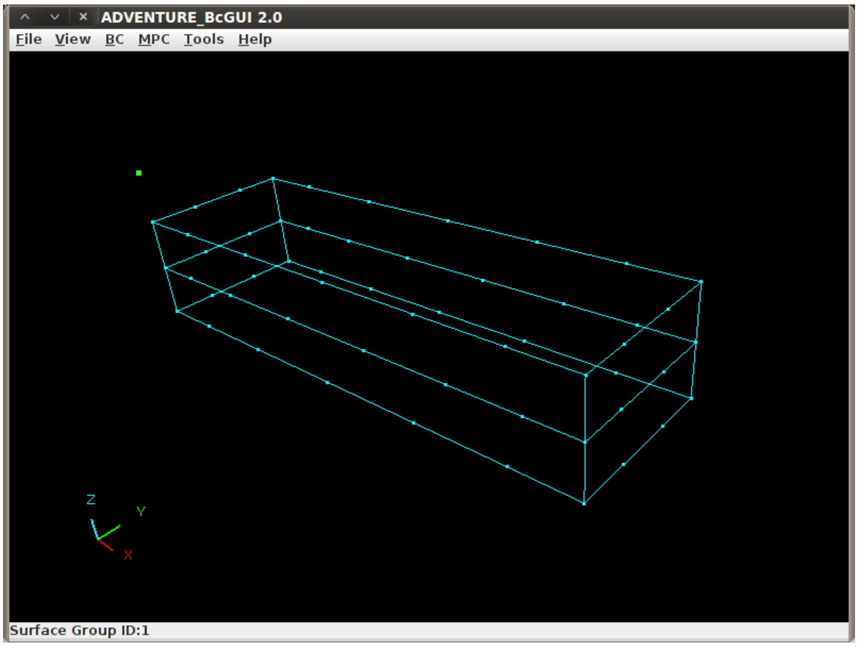


Fig. 5.5.3-12 How a virtual node set to (x, y, z) = (0.0,0.0,3.0) looks

Then select a node to be connected to the virtual node by a left click. The IDs of the selected node will be displayed in the list box "Connected Nodes".

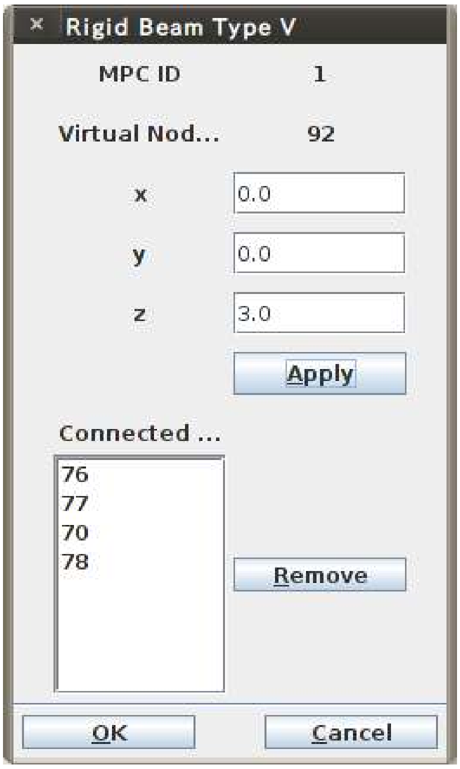


Fig. 5.5.3-13 How selection of the nodes connected to the virtual node looks

Selection of nodes to be connected can be cancelled by first selecting a node number by a left click and then pressing the "Remove" button. After three or more nodes connected to the virtual node are selected, the "OK" button will be ready for click. (However, the "Apply" button must also be clicked at least once) Setting will be completed when the "OK" button is clicked.

"MPC" > "Simple Beam" > "Node to Node"

This can define a Simple Beam that comprises of a node pair and load. The dialog shown in Fig. 5.5.3-14 will be displayed.

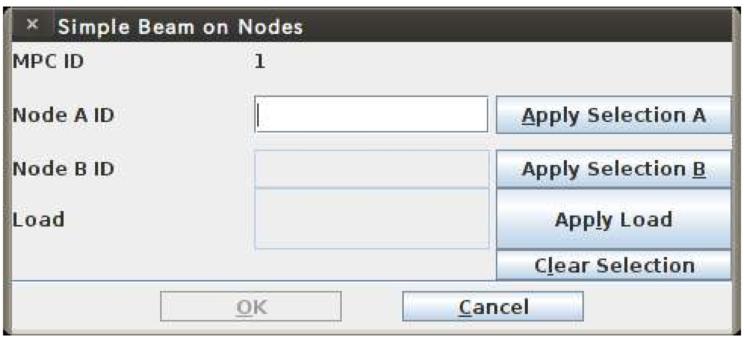


Fig. 5.5.3-14 The dialog for defining a Simple Beam (Before node selection)

Set a node A and a node B by using one of the methods shown in Section 5.2.2.4. After entering a load in the "Load" text field, press the "Apply Load" button. Clicking the "OK" button will complete the setting.

"MPC" > "Linear MPC"

This can define an arbitrary MPC. This is the most generic representation of MPC conditions. This method requires the user to specify the coefficient for each degree of freedom of each node and the right-hand side term. The number of nodes is arbitrary. The dialog in Fig. 5.5.3-15 will appear.

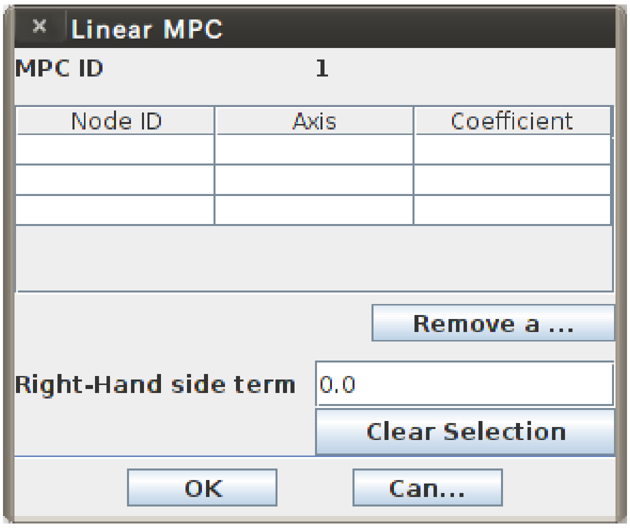


Fig. 5.5.3-15The dialog for setting an arbitrary MPC (Before setting)

If a node is selected by a left click, its number will be displayed in the column of "Node ID" as shown in Fig. 5.5.3-16. The 0, 1, and 2 in the "Axis" column means the x-, y-, and z-axis direction respectively. Enter a value of the coefficient for each axis in the "Coefficient" column. If the "Remove a Node" button is pressed, selection of the node of the currently selected row will be canceled.

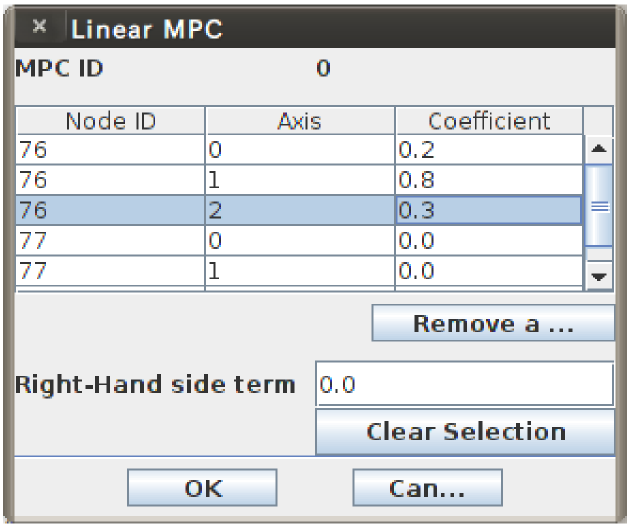


Fig. 5.5.3-16 How node selection and entered coefficients look

Enter a value of the right-hand term in the "Right-Hand side term" text field. Clicking the "OK" button will complete setting. Also, by pressing the "Clear Selection" button, all the selection of nodes, coefficients and the right-hand side term will be cancelled.

5.5.4 Setting Gravitational Acceleration

"BC" > "Gravity Acceleration"

The dialog in Fig. 5.5.4-1 will appear.

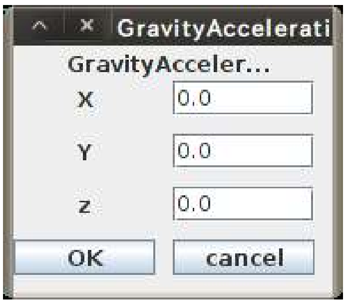


Fig. 5.5.4-1 The dialog for setting gravitational acceleration (Before setting)

Enter each component of the gravity acceleration vector. For example, the dialog will look like Fig. 5.5.4-2. Then click the "OK" button.

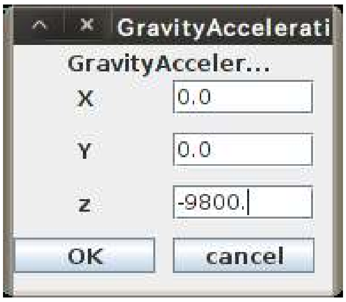


Fig. 5.5.4-2 The dialog for setting gravity acceleration (After values are entered)

5.5.5 Product Information

"Help" > "About"

In Fig. 5.5.5-1, the dialog for product information is shown.

Fig. 5.5.5-1 The dialog for product information

5.5.6 Setting of the Initial Conditions for Dynamic Analysis

"BC" > "BC(Solid)" > "Add InitialVelocity"

The dialog shown in Fig. 5.5.6-1 will appear. The basic operation is the same as in the case of setting ordinary boundary conditions in Section 5.2.2.5. Set an initial velocity on a node or on a surface group by the "Add InitialVelocity" menu item, and then delete one by the "Clear InitialVelocity". Also, of initial conditions for dynamic analysis only an initial velocity can be set.

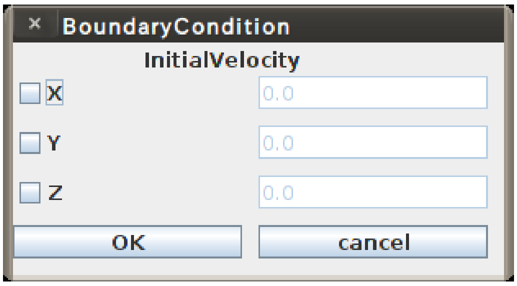


Fig. 5.5.6-1 The dialog for setting an initial condition for dynamic analysis

6. File Formats

For the format of each file type that is used in BCtool, refer to Appendix A.1.

7. Examples

7.1 PcmMerge

Refer to Chapter 4 of the Part II.

7.2 BcGUI

7.2.1 Double Nuts (samples/BcGUI/doubleNut.files/)

The geometry is shown in Fig. 7.2.1-1.

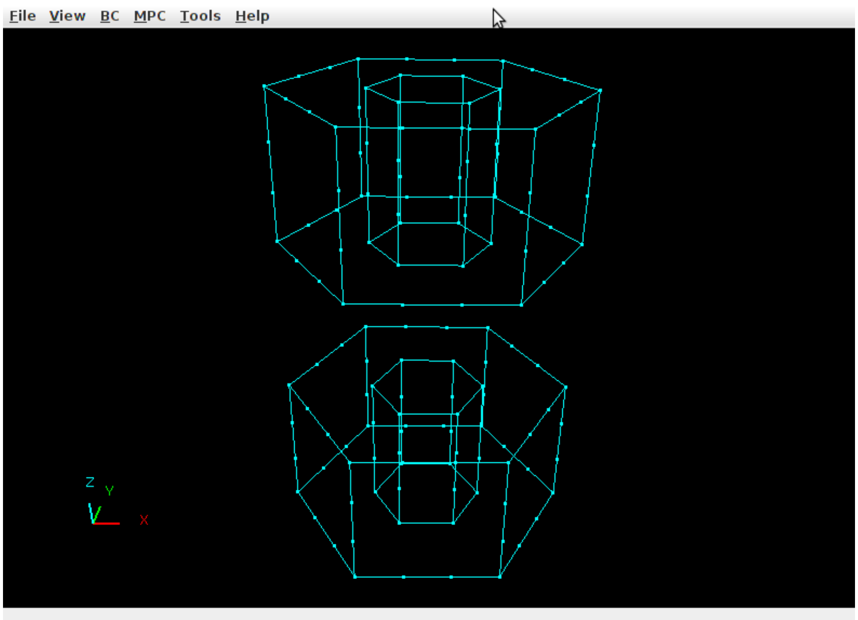


Fig. 7.2.1-1 The geometry of the Double Nut

Input files

Solid_4.0.pch

Solid_4.0.pcg

Output files

MpcI_N.cnd	An example of a Rigid Beam I between nodes
MpcI_SG.cnd	An example of a Rigid Beam I between face groups
MpcII_N.cnd	An example of a Rigid Beam II between nodes
MpcII_SG.cnd	An example of a Rigid Beam II between face groups
MpcIII_N.cnd	An example of a Rigid Beam III between nodes
MpcIII_SG.cnd	An example of a Rigid Beam III between face groups
MpcIV_N.cnd	An example of a Rigid Beam IV between nodes
MpcIV_SG.cnd	An example of a Rigid Beam IV between face groups
MpcV_N.cnd	An example of a Rigid Beam V between nodes
MpcLM_N.cnd	An example of an arbitrary MPC

Files to be used in other processes for analysis execution

Solid_4.0.trn	To be used by the MpcLocal2Global and the makefem3
Solid.msh	To be used by the makefem3
Solid_4.0.fgr	To be used by the makefem3
Solid_4.0.cnd	To be used by the makefem3 (ordinary boundary conditions)
Solid_mp.dat	To be used by the makefem3 (an example in which two different materials are assigned to two volumes)
Solid_4.0.adv:	A result file of the makefem3

7.2.2 A Beam on Top of Another (samples/BcGUI/doubleBeam.files/)

Geometry is shown in Fig. 7.2.2-1.

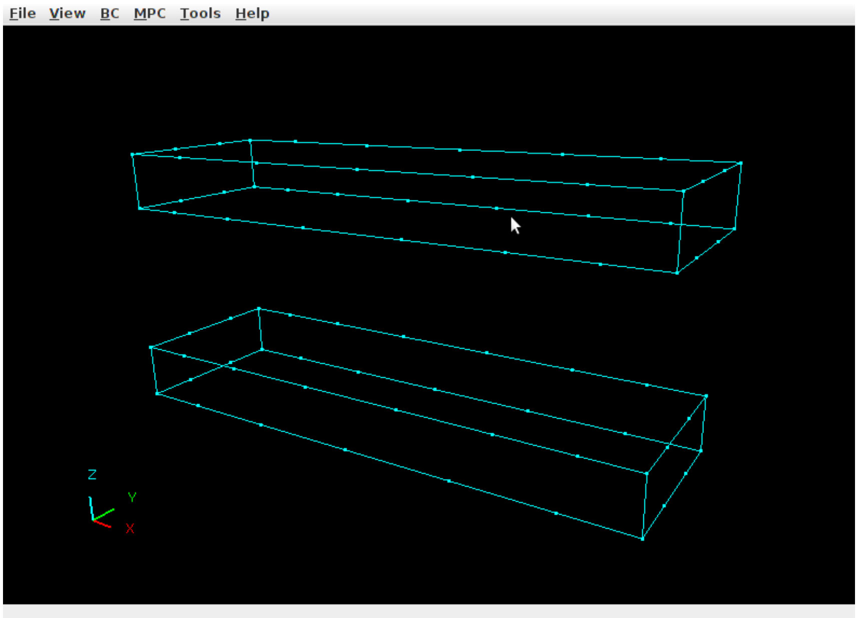


Fig. 7.2.2-1 Geometry of the Double Beam

Input files

Solid_2.0.pch

Solid_2.0.pcg

Output files

MpcI_N.cnd An example of a Rigid Beam I between nodes

Files to be used in other processes for analysis execution

Solid_2.0.trn To be used by the MpcLocal2Global and the makefem3

Solid.msh To be used by the makefem3

Solid_2.0.fgr To be used by the makefem3

Solid_2.0.cnd To be used by the makefem3 (ordinary boundary conditions)

Solid_mp.dat To be used by the makefem3 (an example in which two different materials are assigned to two volumes)

Solid_2.0.adv To be used by the makefem3

7.2.3 A Beam (samples/BcGUI/beam.files/)

Geometry is shown in Fig. 7.2.3-1.

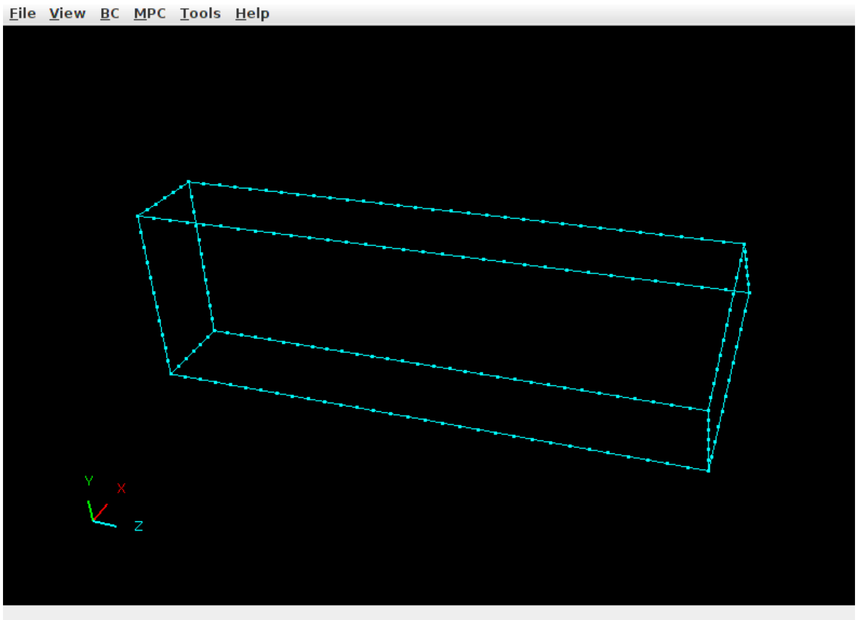


Fig. 7.2.3-1 Geometry of the Beam

Input files

Solid_2.0.pch

Solid_2.0.pcg

Output files

So_loadV.cnd	An example of nodal concentrated load conditions
So_dispV.cnd	An example of nodal forced displacement conditions
So_dispF.cnd	An example of forced displacement conditions on surface groups
So_velocV.cnd	An example of nodal forced velocity conditions
So_velocF.cnd	An example of forced displacement conditions on surface groups
So_accelV.cnd	An example of nodal forced acceleration conditions
So_accelF.cnd	An example of forced acceleration conditions on surface groups
So_presF.cnd	An example of pressure conditions on surface groups
So_tracF.cnd	An example of surface traction conditions on surface groups
Th_tempV.cnd	An example of nodal temperature conditions
Th_tempF.cnd	An example of temperature conditions on surface groups
Th_fluxF.cnd	An example of heat flux conditions on surface groups
Th_transF.cnd	An example of heat transfer conditions on surface groups
Th_radiF.cnd	An example of heat radiation conditions on surface groups

Files to be used in other processes for analysis execution

Solid_2.0.trn	To be used by the MpcLocal2Global and the makefem3
Solid.msh	To be used by the makefem3

Solid_2.0.fgr	To be used by the makefem3
Solid_2.0.cnd	To be used by the makefem3 (ordinary boundary conditions)
Solid_mp.dat	To be used by the makefem3
Solid_2.0.adv	To be used by the makefem3

7.2.4 Hexahedron (samples/BcGUI/doubleHex/)

Geometry is shown in Fig. 7.2.4-1.

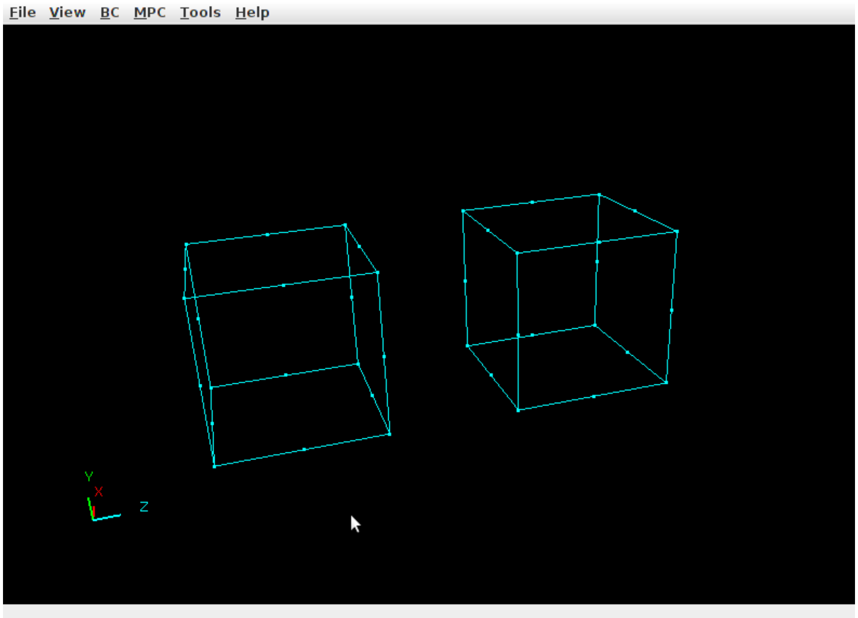


Fig. 7.2.4-1 Geometry of the doubleHex

Input files

doubleHex_2.pch

doubleHex_2.pcg

Output files

Rigid_Beam_I.cnd	An example of a Rigid Beam I
Rigid_Beam_II.cnd	An example of a Rigid Beam II
Rigid_Beam_III.cnd	An example of a Rigid Beam III
Rigid_Beam_IV.cnd	An example of a Rigid Beam IV

7.2.5 MultiBrick (samples/BcGUI/multiVolume/)

Geometry is shown in Fig. 7.2.5-1.

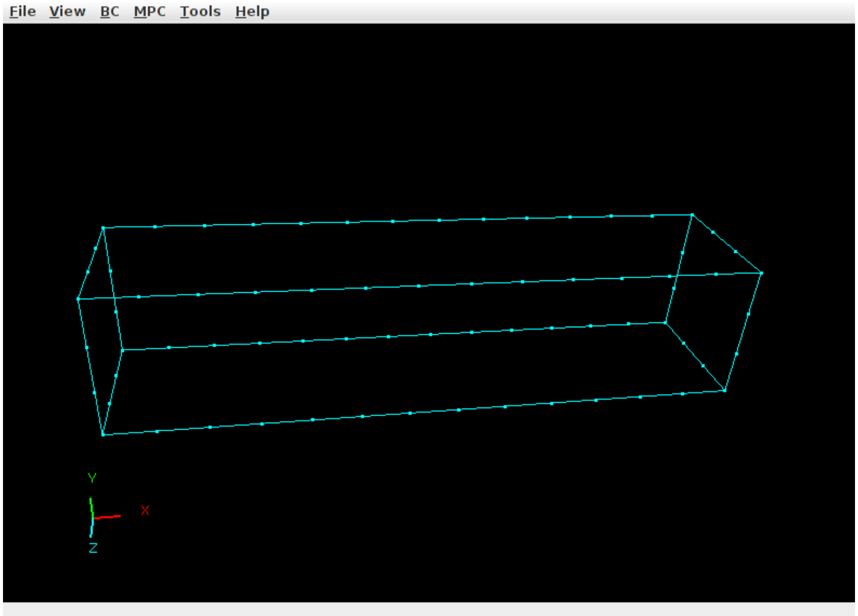


Fig. 7.2.5-1 Geometry of the MultiBrick

Input files

multiBrick_10.pch

multiBrick_10.pcg

multiBrick_V.pcm

multiBrick_V.pcg

Files to be used in other processes for analysis execution

multiBrick.msh	To be used by the makefem3
multiBrick_10.fgr	To be used by the makefem3
multiBrick_10.trn	To be used by the MpcLocal2Global and the makefem3

7.2.6 Twenty-seven Cubes (samples/BcGUI/multiVolume/three^3Box/)

Geometry is shown in Fig. 7.2.6-1.

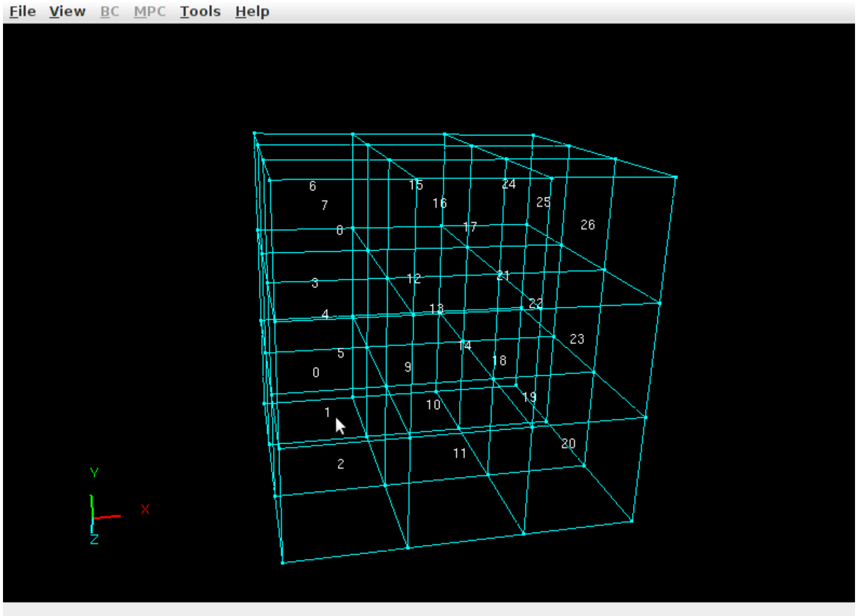


Fig. 7.2.6-1 Geometry of the three^3Box

Input files

Solid26mc_V.pcm

Solid26mc_V.pcg

Files to be used in other processes for analysis execution

Solid26mc.msh To be used by the makefem3

7.2.7 Three Cubes (samples/BcGUI/multiVolume/threeBox/)

Geometry is shown in Fig. 7.2.7-1.

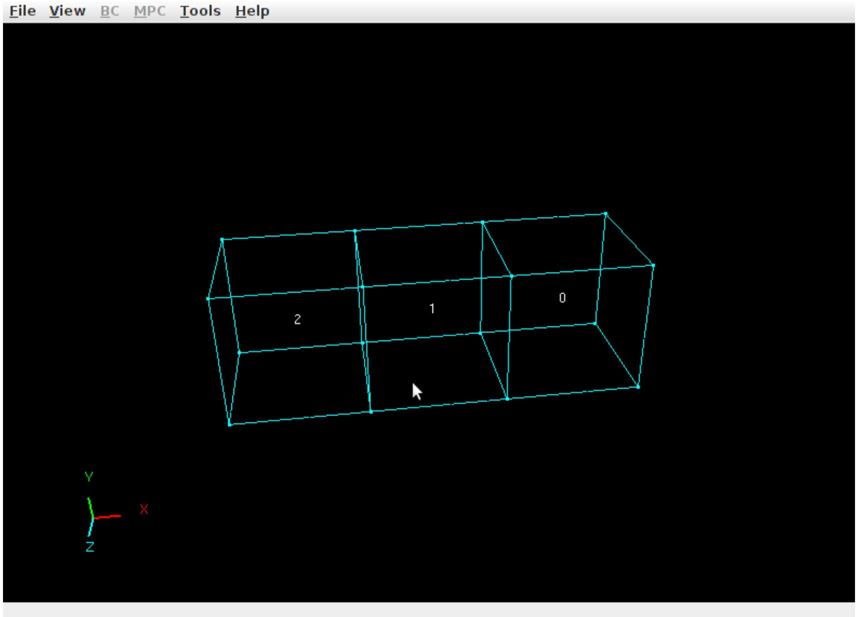


Fig. 7.2.7-1 Geometry of the ThreeBox

Input files

Solid2mc_V.pcm

Solid2mc_V.pcg

Files to be used in other processes for analysis execution

Solid2mc.msh To be used by the makefem3

7.3 makefem3

Description of the example files for the makefem3 will be given.

7.3.1 A Single Volume Tetrahedral Linear Element Mesh (samples/ makefem3/ 1ji_single_tetra/)

The screen that displays the example in BcGUI is shown in Fig. 7.3.1-1.

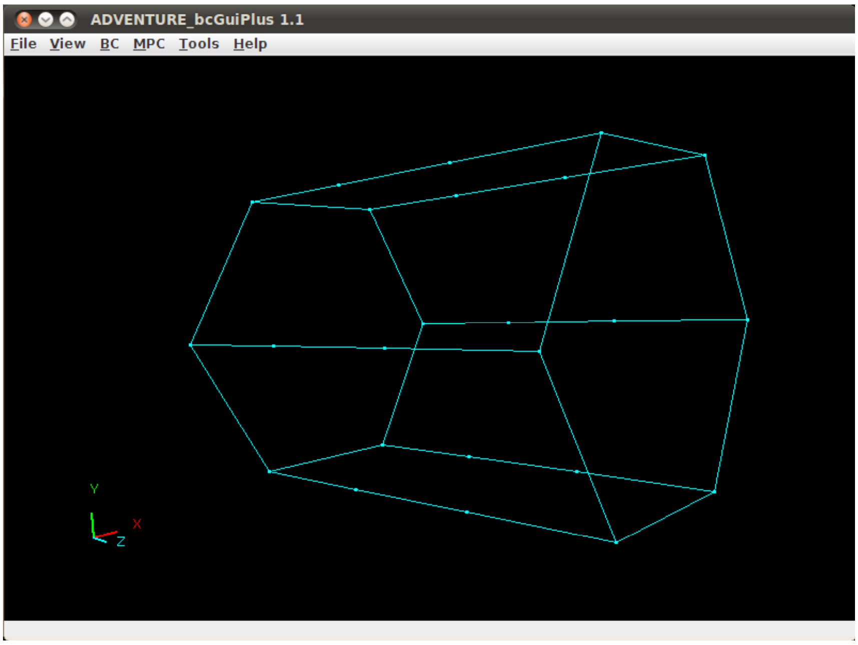


Fig. 7.3.1-1 Geometry of the example of single volume tetrahedral linear element mesh

Input files

model/Solid.msh	A mesh data file of single volume tetrahedral linear elements	
model/Solid_2.0.fgr	A mesh surface data file	
model/Solid_2.0.pcg	A surface patch group data file (Not to be used by the makefem3)	
model/Solid_2.0.pch	A surface patch data file (Not to be used by the makefem3)	
model/Solid_2.0.trn	A global index file	
model/Solid_mp.dat	A material properties data file	(1) (Either (1) or (2).)
model/test_mp.dat	A material properties data file	(2)
cnd/mix.cnd	A boundary condition file	(a) (A file among (a) to (i))
cnd/mix2.cnd	A boundary condition file	(b)
cnd/initveloc/initveloc_v.cnd	A boundary condition file	(c)
cnd/initveloc/initveloc_fg.cnd	A boundary condition file	(d)
cnd/mesh/mesh.cnd	A boundary condition file	(e)
cnd/nl_history_id/dispOnVertex.cnd	A boundary condition file	(f)
cnd/nl_history_id/dispOnFaceGroup.cnd	A boundary condition file	(g)
cnd/nl_history_id/loadOnVertex.cnd	A boundary condition file	(h)

cnd/nl_history_id/tracOnFaceGroup.cnd A boundary condition file (i)

Output files

* Symbols in parentheses refer to the combination of a material properties file and an analysis condition file.

makebc_result/makebc.adv	(1, a)
makefem3_result/cond.adv	(1, b)
makefem3_result/initvelocOnVertex.adv	(1, c)
makefem3_result/initvelocOnFaceGroup.adv	(1, d)
makefem3_result/material.adv	(2, e)
makefem3_result/mesh.adv	(1, e)
makefem3_result/nl_dispOnVertex.adv	(1, f)
makefem3_result/nl_dispOnFaceGroup.adv	(1, g)
makefem3_result/nl_loadOnVertex.adv	(1, h)
makefem3_result/nl_tracOnFaceGroup.adv	(1, I)

7.3.2 A Single Volume Tetrahedral Quadratic Element Mesh (samples/makefem3/2ji_single_tetra/)

The screen that displays the example in BcGUI is shown in Fig. 7.3.2-1.

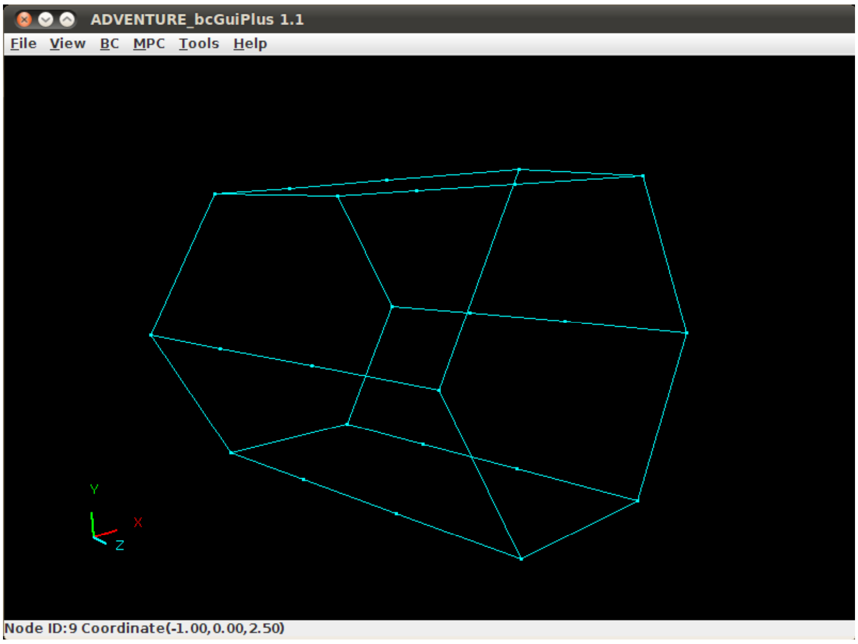


Fig. 7.3.2-1 Geometry of the example of single volume tetrahedral quadratic element mesh

Input files

model/Solid.msh	A mesh data file of single volume tetrahedral quadratic elements
model/Solid_2.0.fgr	A mesh surface data file
model/Solid_2.0.pcg	A surface patch group data file (Not to be used by the makefem3)
model/Solid_2.0.pch	A surface patch data file (Not to be used by the makefem3)
model/Solid_2.0.trn	A global index file
model/Solid_mp.dat	A material properties data file
cnd/all.cnd	A boundary condition file (*It does not include all the boundary conditions)
nonlinear.cnd	A boundary condition file
normal.cnd	A boundary condition file
transient.cnd	A boundary condition file

Output files

result.adv

7.3.3 A Single Volume Hexahedral Linear Element Mesh (samples/makefem3/1ji_single_hexa/)

The screen that displays the example in BcGUI is shown in Fig. 7.3.3-1.

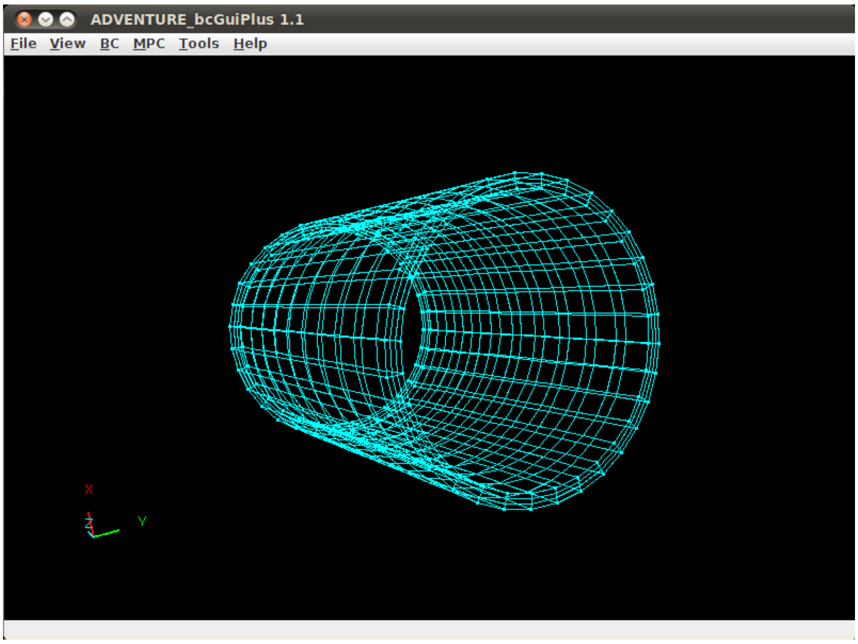


Fig. 7.3.3-1 Geometry of the example of single volume hexahedral linear element mesh

Input files

model/testCylinder_1ji_single.msh	A mesh data file of single volume hexahedral linear elements
model/testCylinder_1ji_single_4.fgr	A mesh surface data file
model/testCylinder_1ji_single_4.pcg	A surface patch group data file (Not to be used by the makefem3)
model/testCylinder_1ji_single_4.pch	A surface patch data file (Not to be used by the makefem3)
model/testCylinder_1ji_single_4.trn	A global index file
model/s_material.dat	A material properties data file
cnd/all.cnd	A boundary condition file(*It does not include all the boundary conditions)

Output files

result.adv

7.3.4 A Single Volume Hexahedral Quadratic Element Mesh (samples/ makefem3/ 2ji_single_hexa/)

The screen that displays the example in BcGUI is shown in Fig. 7.3.4-1.

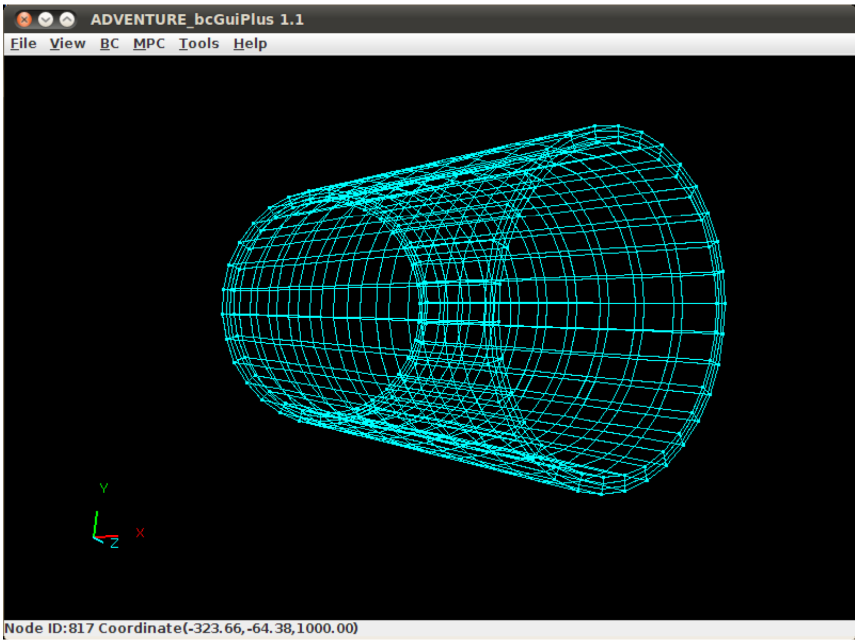


Fig. 7.3.4-1 Geometry of the example of single volume hexahedral quadratic element mesh

Input files

model/testCylinder_2ji_single.msh	A mesh data file of single volume hexahedral quadratic elements
model/testCylinder_2ji_single_4.fgr	A mesh surface data file
model/testCylinder_2ji_single_4.pcg	A surface patch group data file (Not to be used by the makefem3)
model/testCylinder_2ji_single_4.pch	A surface patch data file (Not to be used by the makefem3)
model/testCylinder_2ji_single_4.trn	A global index file
model/s_material.dat	A material properties data file
cnd/all.cnd	A boundary condition file(*It does not include all the boundary conditions)

Output files

result.adv

7.3.5 A Multiple Volume Tetrahedral Linear Element Mesh (samples/ makefem3/ 1ji_multi_tetra/)

The screen that displays the example in BcGUI is shown in Fig. 7.3.5-1.

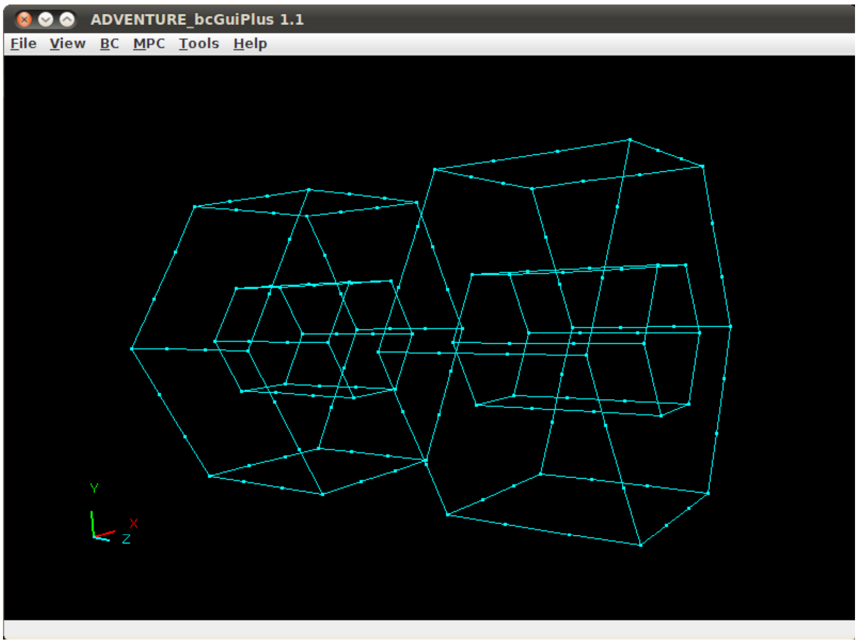


Fig. 7.3.5-1 Geometry of the example of multiple volume tetrahedral linear element mesh

Input files

model/single.msh	A mesh data file of single volume tetrahedral linear elements
model/multi_2.fgr	A mesh surface data file
model/multi_2.pcg	A surface patch group data file (Not to be used by the makefem3)
model/multi_2.pch	A surface patch data file (Not to be used by the makefem3)
model/multi_2.trn	A global index file
model/m_material.dat	A material properties data file
cnd/all.cnd	A boundary condition file(*It does not include all the boundary conditions)

Output files

result.adv

7.3.6 A Multiple Volume Tetrahedral Quadratic Element Mesh (samples/makefem3/ 2ji_multi_tetra/)

The screen that displays the example in BcGUI is shown in Fig. 7.3.6-1.

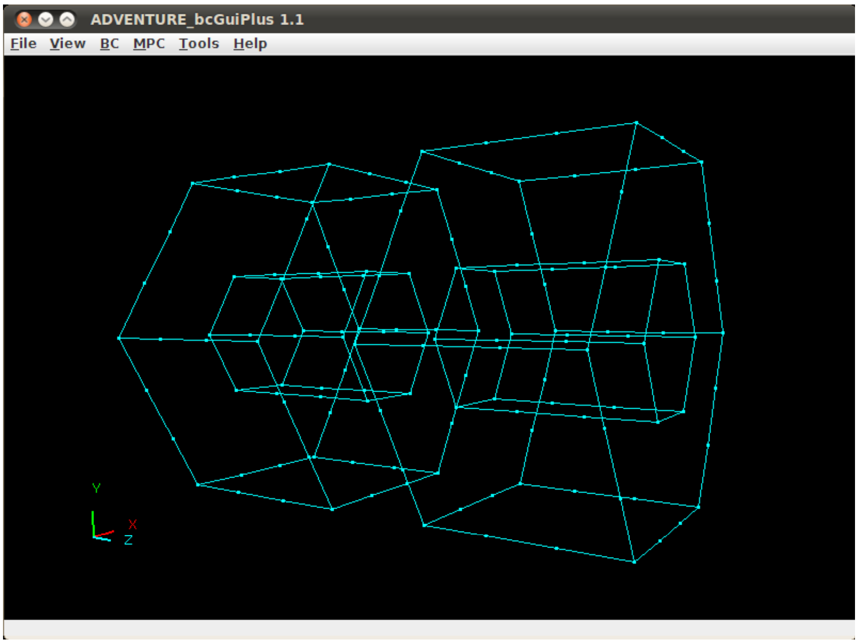


Fig. 7.3.6-1 Geometry of the example of multiple volume tetrahedral quadratic element mesh

Input files

model/single.msh	A mesh data file of single volume tetrahedral linear elements
model/multi_2.fgr	A mesh surface data file

model/multi_2.pcg A surface patch group data file (Not to be used by the makefem3)

model/multi_2.pch A surface patch data file (Not to be used by the makefem3)

model/multi_2.trn A global index file

model/m_material.dat A material properties data file

cnd/all.cnd A boundary condition file(*It does not include all the boundary conditions)

Output files

result.adv

7.3.7 A Multiple Volume Hexahedral Linear Element Mesh (samples/makefem3/1ji_multi_hexa/)

The screen that displays the example in BcGUI is shown in Fig. 7.3.7-1.

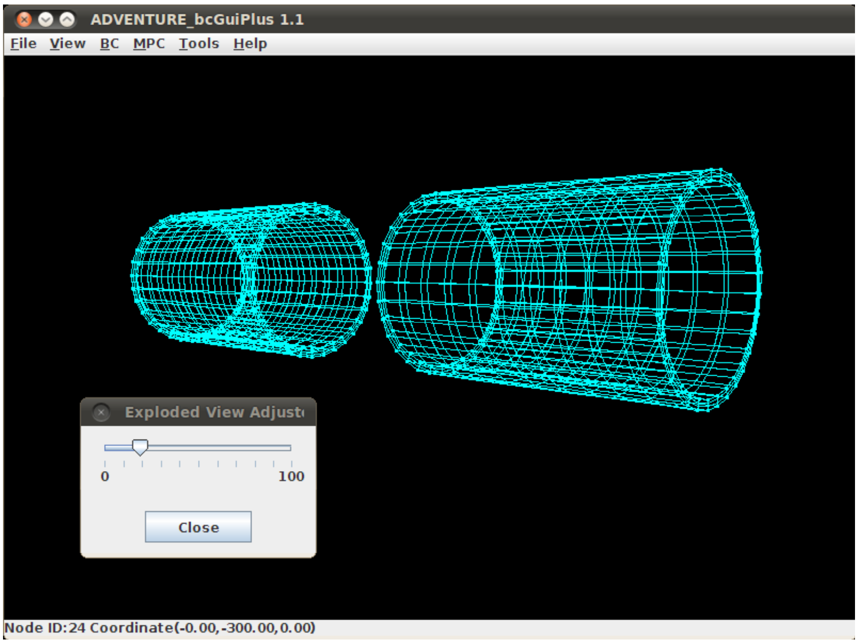


Fig. 7.3.7-1 Geometry of the example of multiple volume hexahedral linear element mesh

Input files

model/testCylinder_1ji_multi.msh A mesh data file of single volume hexahedral linear elements

model/testCylinder_1ji_multi_4.fgr A mesh surface data file

model/testCylinder_1ji_multi_4.pcg A surface patch group data file

(Not to be used by the makefem3)

model/testCylinder_1ji_multi_4.pch	A surface patch data file (Not to be used by the makefem3)
model/testCylinder_1ji_multi_4.trn	A global index file
model/m_material.dat	A material properties data file
cnd/all.cnd	A boundary condition file(*It does not include all the boundary conditions)

Output files

result.adv

7.3.8 A Multiple Volume Hexahedral Quadratic Element Mesh (samples/makefem3/2ji_multi_hexa/)

The screen that displays the example in BcGUI is shown in Fig. 7.3.8-1.

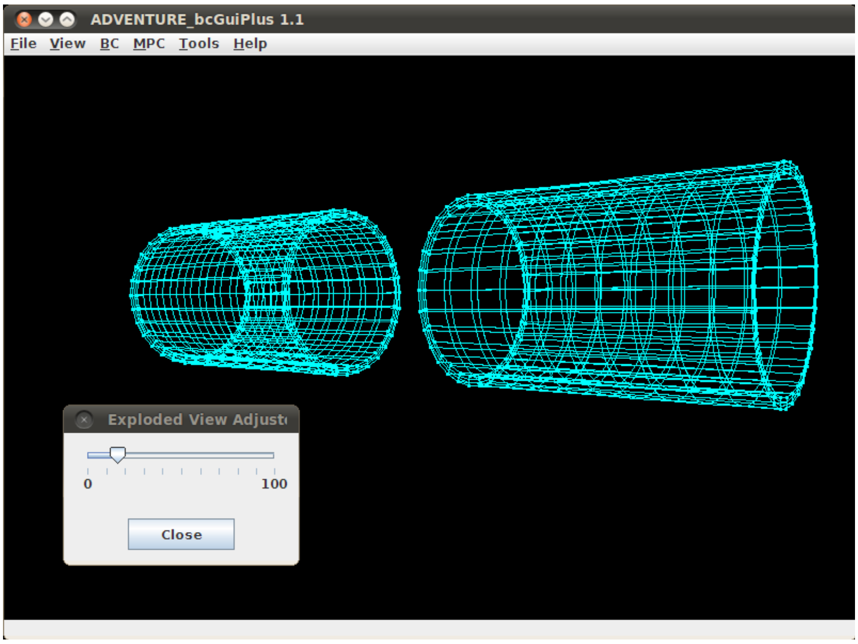


Fig. 7.3.8-1 Geometry of the example of multiple volume hexahedral quadratic element mesh

Input files

model/testCylinder_2ji_multi.msh	A mesh data file of multiple volume hexahedral quadratic elements
model/testCylinder_2ji_multi_4.fgr	A mesh surface data file
model/testCylinder_2ji_multi_4.pcg	A surface patch group data file (Not to be used by the makefem3)

model/testCylinder_2ji_multi_4.pch	A surface patch data file (Not to be used by the makefem3)
model/testCylinder_2ji_multi_4.trn	A global index file
model/m_material.dat	A material properties data file
cnd/all.cnd	A boundary condition file(*It does not include all the boundary conditions)

Output files

result.adv

7.4 MpcMasterSlaveTool

Refer to Chapter 6 of Part III.

7.5 a2adv

Refer to Chapter 7 of Part IV.

7.6 MpcLocal2Global

Refer to Chapter 5 of Part V.

7.7 csv2adv

Example files

・Inputs

1-1.csv

1-2.csv

1-3.csv

・Outputs

test1.a

・Inputs

2-1.csv

2-2.csv

2-3.csv

・Outputs

None

・Inputs

3-1.csv

3-2.csv

3-3.csv

・Outputs

test3.a

Execution examples

・In the case of one input file

```
% csv2adv.pl result.a bc_timehistory.csv
```

・In the case of three input files

```
% csv2adv.pl result.a 1-1.csv 1-2.csv 1-3.csv
```

7.8 Thermal

7.8.1 Temperature Specified Boundary Conditions (samples/Thermal/test_temp_only)

The screen that displays the example in BcGUI is shown in Fig. 7.8.1-1. The surface groups in red indicate that temperature boundary conditions are applied.

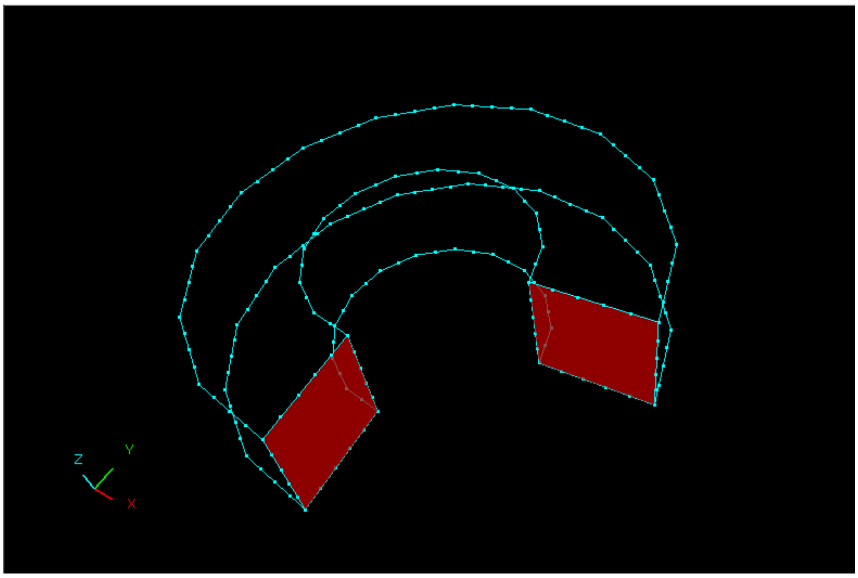


Fig. 7.8.1-1 An example data with temperature specified boundary conditions

Input files

requestModelc.msh	:A mesh data file
requestModelc_3.fgr	:A mesh surface data file
requestModelc_3.cnd	:A boundary condition file
requestModelc_3.dat	:A material properties data file
requestModelc_3.trn	:A global index file

Output files

  requestModel.adv :An integrated input file

7.8.2 Heat Flux Specified Boundary conditions (samples/Thermal/test_heatflux)

The screen that displays the example in BcGUI is shown in Fig. 7.8.2-1. The red surface group indicates temperature specified boundary conditions and the yellow one indicates heat flux specified boundary conditions.

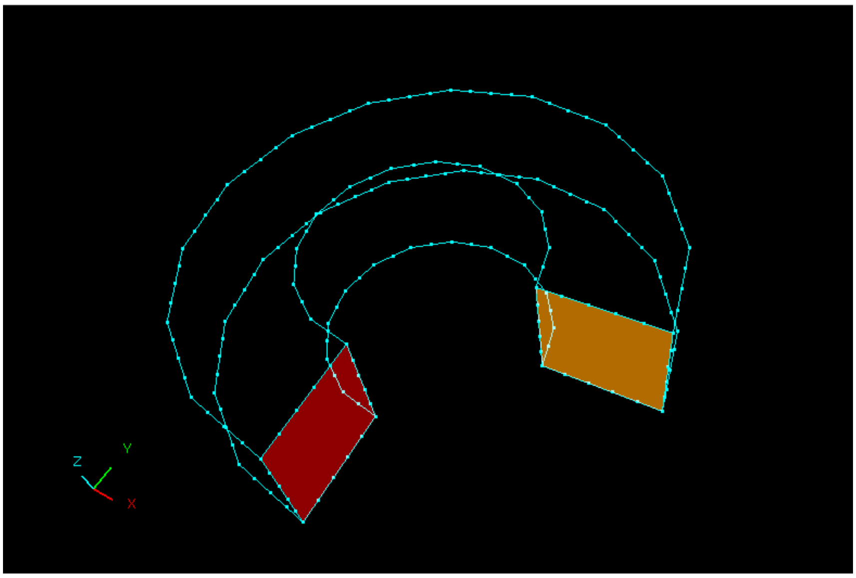


Fig. 7.8.2-1 An example data with heat flux specified boundary conditions

Input files

requestModelc.msh	:A mesh data file
requestModelc_3.fgr	:A mesh surface data file
requestModelc_3.cnd	:A boundary condition file
requestModelc_3.dat	:A material properties data file
requestModelc_3.trn	:A global index file

Output files

  requestModel.adv :An integrated input files

7.8.3 Heat Transfer Specified Boundary Conditions (samples/Thermal/test_heattransfer)

The screen that displays the example in BcGUI is shown in Fig. 7.8.3-1. The red surface group indicates temperature specified boundary conditions and the light pink one indicates heat transfer specified boundary conditions.

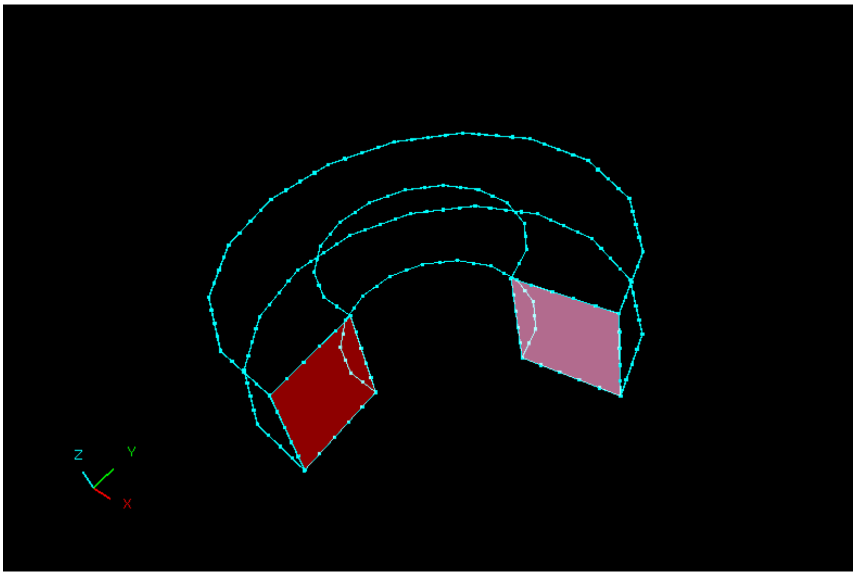


Fig. 7.8.3-1 An example data with heat transfer specified boundary conditions

Input files

requestModelc.msh	:A mesh data file
requestModelc_3.fgr	:A mesh surface data file
requestModelc_3.cnd	:A boundary condition file
requestModelc_3.dat	:A material properties data file
requestModelc_3.trn	:A global index file

Output files

requestModel.adv	:An Integrated input files

7.8.4 Thermal Radiation Specified Boundary Conditions (samples/Thermal/test_heatradiation)

The screen that displays the example in BcGUI is shown in Fig. 7.8.4-1. The red surface group indicates temperature specified boundary conditions and the pink one indicates thermal radiation specified boundary conditions.

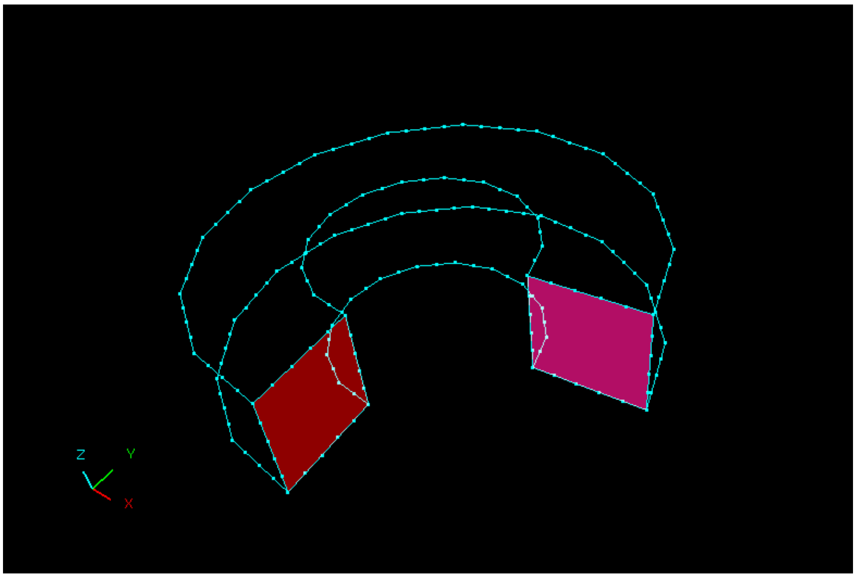


Fig. 7.8.1-1 An example data with thermal radiation specified boundary conditions

Input files

requestModelc.msh	:A mesh data file
requestModelc_3.fgr	:A mesh surface data file
requestModelc_3.cnd	:A boundary condition file
requestModelc_3.dat	:A material properties data file
requestModelc_3.trn	:A global index file

Output files

requestModel.adv	:An Integrated input files

Part II The Manual for the PcmMerge

The PcmMerge is a converter tool to generate a mesh of two volumes applicable to analyses with MPC conditions.

Version information: PcmMerge Ver.0.1b (Dec. 1, 2006)

1. What is the PcmMerge?

The PcmMerge is an auxiliary tool for generating from two surface patches a mesh file of two volumes that enables analyses with MPC conditions. The two given surface patches must have two corresponding surface groups either in contact, in the state of interference or slightly apart as a prerequisite. Figure 1-1 shows its data flow.

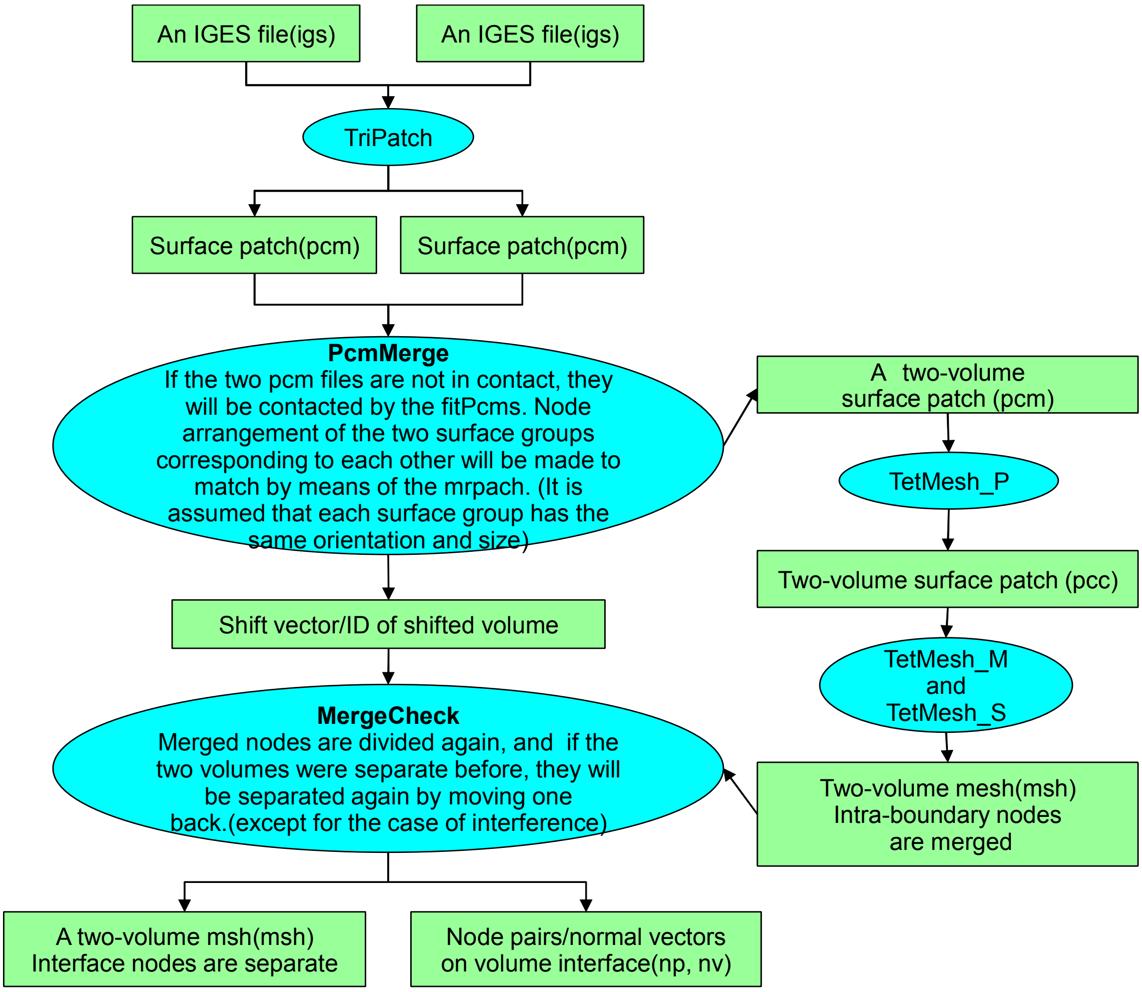


Fig. 1-1 The data flow with regard to the PcmMerge

The PcmMerge is made up of two tools, i.e. the PcmMerge and the MergeCheck. The former has the ability to merge two given surface patch files into one, and the latter has the ability to split the merged nodes between the two merged volumes to node pairs.

2. The Operating Environment and Compilation

In order to make a series of operations in Figure 1-1, it is necessary, besides this converter, for the tool mrpach of the ADVENTURE_TriPatch and the ADVENTURE_TetMesh to be installed. For compilation of this converter, the JDK (Java Development Toolkit, 1.6 or later) and the GNU Make is required.

Operation verification was done with the following environments.

*Linux (Kernel 2.6.9-42.0.3.EL CentOS 4.4 (Final)) + JDK 1.6.0_16 + GNU Make 3.80
*Linux(Kernel 2.6.32-54-generic Ubuntu x86_64) + OpenJDK 1.6.0_27 + GNU Make 3.81
*Microsoft Windows 7 Home Premium SP 1, 64bit + Cygwin 1.7.27 + JDK 1.7.0_45 + GNU Make 4.0
*Microsoft Windows Vista Home Premium SP2 + Cygwin 1.7.27 x86_64 + JDK 1.7.0_45 64-Bit
*Microsoft Windows 8 Pro + Cygwin 1.7.27 x86_64 + JDK 1.7.0_45 64-Bit + GNU Make 4.0

After unpacking the archive file, go to the src/pcmMerge directory first and edit the Makefile's following line:

```
PREFIX          =/home/user/ADVENTURE/
```

Change the right-hand side of the line as appropriate to the directory where the ADVENTURE is installed. Use of the environment variable $(HOME) does not seem to work. Next, execute the following.

```
% make
```

On Cygwin, installing just the JDK does not set PATH to the compilation command javac. So, set the PATH as appropriate.

If compilation is successful, a class file is created per each source file (extension is java) in the same directory as the java source file. Next, execute the following.

```
% make install
```

Then, all the class files will be packaged into an archive file FitPcms.jar and will be placed in the current directory. Moreover, the FitPcms.jar will be copied to ${PREFIX}lib, the script PcmMerge and the script MergeCheck will be copied to ${PREFIX}bin. It should be noted that, when installing the tool in system area on Linux, a root privilege is needed. So, use the su or the sudo command.

When deleting compile results, execute the following.

```
% make clean
```

3. How to Execute the Commands

3.1 Execution of the PcmMerge

The PcmMerge is a shell script. The script executes a program written in Java and a tool mrpach of ADVENTURE_TriPatch. It is assumed that the PATH includes the directories that contain the present script and the mrpach.

```
% PcmMerge [Options] pcmFileA pcmFileB [-o outPcmFile] [-g outPcgFile] [-m moveInfoFile]
```

Parameters:

pcmFileA	(Input) :One of the extracted surface mesh data file (extension needed)
pcmFileB	(Input) :The other extracted surface mesh data file (extension needed)
-o outPcmFile	(Output):Merged extracted surface mesh data file (extension needed)
-g outPcgFile	(Output):Merged surface patch group data file (extension needed)
-m moveInfoFile	(Output):Volume movement information file(extension needed)

[Options] are as follows:

-h or --help	:The option to show the usage of the program
-version	:The option to show the version of the program
-a or --angle	:The Option to specify in degrees a dihedral included angle by which to group the patch surface
-t or --tolerance	:The tolerance of the gap between the corresponding two surface groups

The following is detailed description of each parameter.

pcmFileA (Input)

One of the surface patch data files. This surface patch is the one that will not be moved by the process of this script. Either the pcm format or the pch format can be used.

pcmFileB (Input)

The other surface patch data file. This is the one that will be moved so that the two patches will be just in contact when the two patches are either separate or in the state of interference. Either the pcm format or the pch format can be used.

For the pcm format of the extracted surface mesh data file, see Section 6.2, "Surface Patch Data File" in Chapter 6, in the ADVENTURE_TriPatch manual. For the pch format, see "A.1.4 Surface Mesh Extraction Data File (extension is pch)" of "A.1 Various File Formats" of this manual.

-t or --tolerance (Input)

When there is a gap between the corresponding two surface patches, the two bodies will be regarded as in contact if the distance is less than or equal to the specified tolerance, and neither of the surface patches will be moved. The tolerance defaults to 1.e-5.

-a or --angle (Input)

If the included angle between two neighboring element faces is equal to or greater than the user-designated angle, the two faces are considered to belong to different surface groups. The default value is 45 degrees.

The following parameters are omissible and designate the output files of this tool.

-o outPcmFile

It is a surface patch data file made by merging the pcmFileA and the pcmFileB. Nodes on the boundary between the two volumes are duplicate. The default file name is out.pcm.

-g outPcgFile

It is a surface patch group data file made from the surface patch data file that is made by merging the pcmFileA and the pcmFileB. The default file name is out.pcg

-m moveInfoFile

In the moveInfo.dat, the type of positional relationship between the two volumes (contact, separation, or interference), the volume ID of the surface patch that will be moved, and the movement vector will be described. The default file name is moveInfo.dat.s

Other output files include modelA.pcg, modelA.pcm, modelB.pcg, modelB.pcm and out.wrl, but these are not needed for subsequent processing.

3.2 Execution Examples of the PcmMerge

If, for example, the following command is executed,

```
% PcmMerge -a20. -t1.e-4 body1.pcm body2.pcm
```

the following files will be newly created.

./out.pcm

./out.pcg

./moveInfo.dat

3.3 Execution of the MergeCheck

The MergeCheck is a shell script. The script executes a program written in Java. It is assumed that the PATH includes the directory that contains the script file.

```
% MergeCheck [Options] mshFile moveInfoFile outputFile
```

Parameters:

mshFile (Input) :A mesh data file

moveInfoFile (Input): Volume movement information file

outputFile (Output): An arbitrary output file named by the user. It may not include extension.

[Options] are as follows:

-h or --help :The option to show the usage of the program

-v or --version :The option to show the version of the program

The following is detailed description of each parameter.

mshFile (Input)

It is a tetrahedral mesh data file. Either the linear element or the quadratic element can be used.

moveInfoFile (Input)

A file generated by the execution of the PcmMerge is used. The type of positional relationship between the two volumes (contact, separation, or interference), the volume ID of the surface patches that were moved, and the movement vector are described.

outputFile (Output)

Three files generated by the program. ouputFile.msh, outputFile.np and outputFile.nv will be output.

outputFile.msh: it is a two-volume tetrahedral mesh data file with the nodes on the inter-volume boundary are split.

outputFile.np: it is a file that describes the numbers of node pairs on the corresponding two surface groups on each volume surface. Its format is shown in Section A.1.13.1.

outputFile.nv:it is a file that describes the normal vectors at each node on the corresponding two surface groups on each volume surface.

3.4 Execution Example of the MergeCheck

If, for example, the following command is executed,

```
% MergeCheck doubleNut.msh moveInfoFile.dat doubleNutOut
```

the following files will be newly created.

./doubleNutOut.msh
./doubleNutOut.np
./doubleNutOut.nv

4. Examples

4.1 Two Nuts

The following files are available as examples.

PcmMerge input files

samples/PcmMerge/doubleNut/lowerNut-2mm.pch
samples/PcmMerge/doubleNut/upperNut-2mm.pch

The coordinate ranges of the lowerNut-2mm.pch and the upperNut-2mm.pch and are (-5, -4.33, 0) to (5, 4.33, 5) and (-5, -4.33, 6) to (5, 4.33, 11) respectively. Geometry is shown in Fig. 4.1-1 and Fig. 4.1-2 respectively.

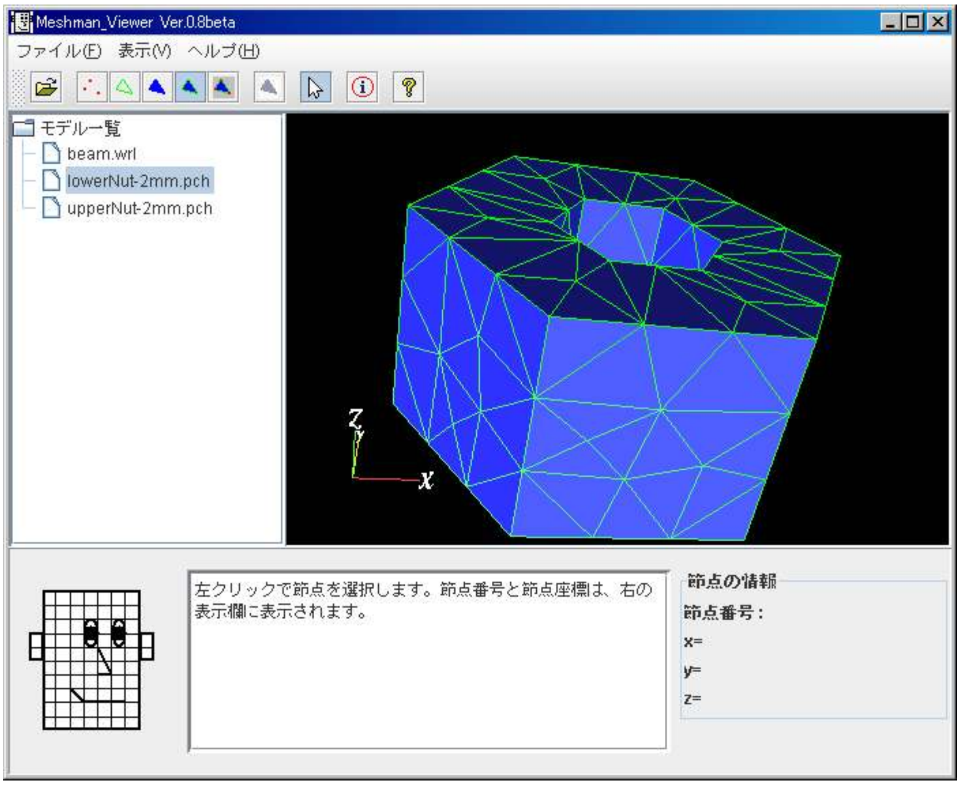


Fig. 4.1-1 Geometry of the lowerNut-2mm.pch

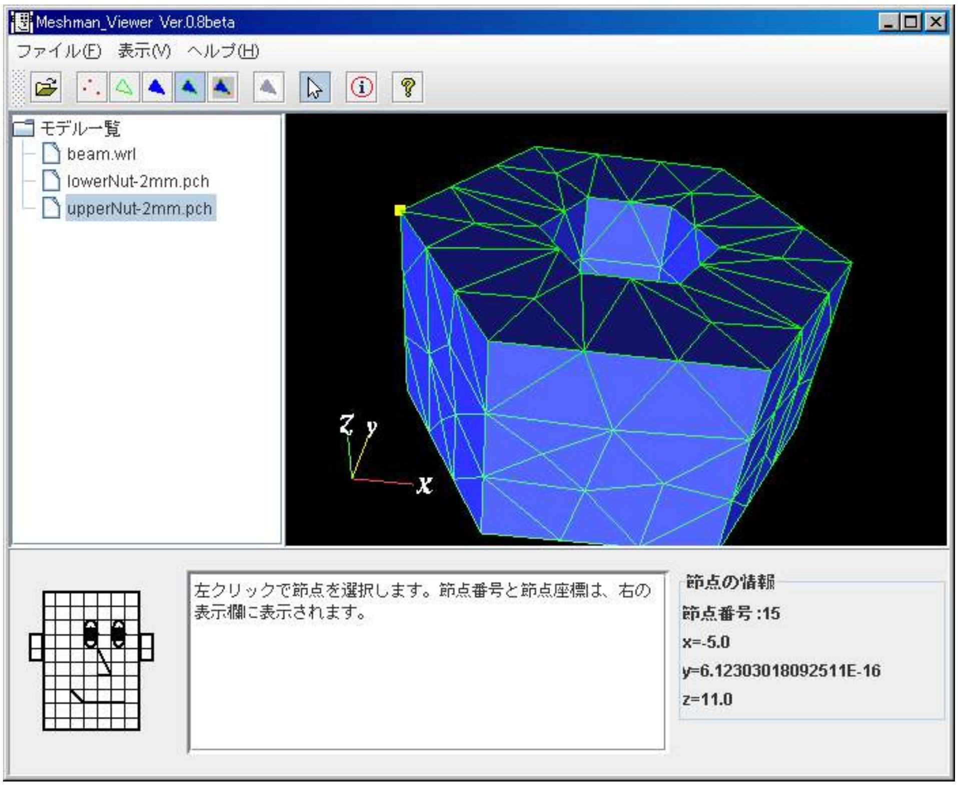


Fig. 4.1-2 Geometry of the upperNut-2mm.pch

As can be seen from their names, the lowerNut is under the upperNut, and they are in contact with each other. Specifying as pcmFileA the lowerNut-2mm.pch, run the following command PcmMerge.

```
% PcmMerge lowerNut-2mm.pch upperNut-2mm.pch
```

Then the following two PcmMerge output files will be output.

samples/PcmMerge/doubleNut/out.pcm

samples/PcmMerge/doubleNut/moveInfo.dat

Fig. 4.1-3 shows the geometry of the out.pcm. It has a shape made by merging the upperNut and the lowerNut.

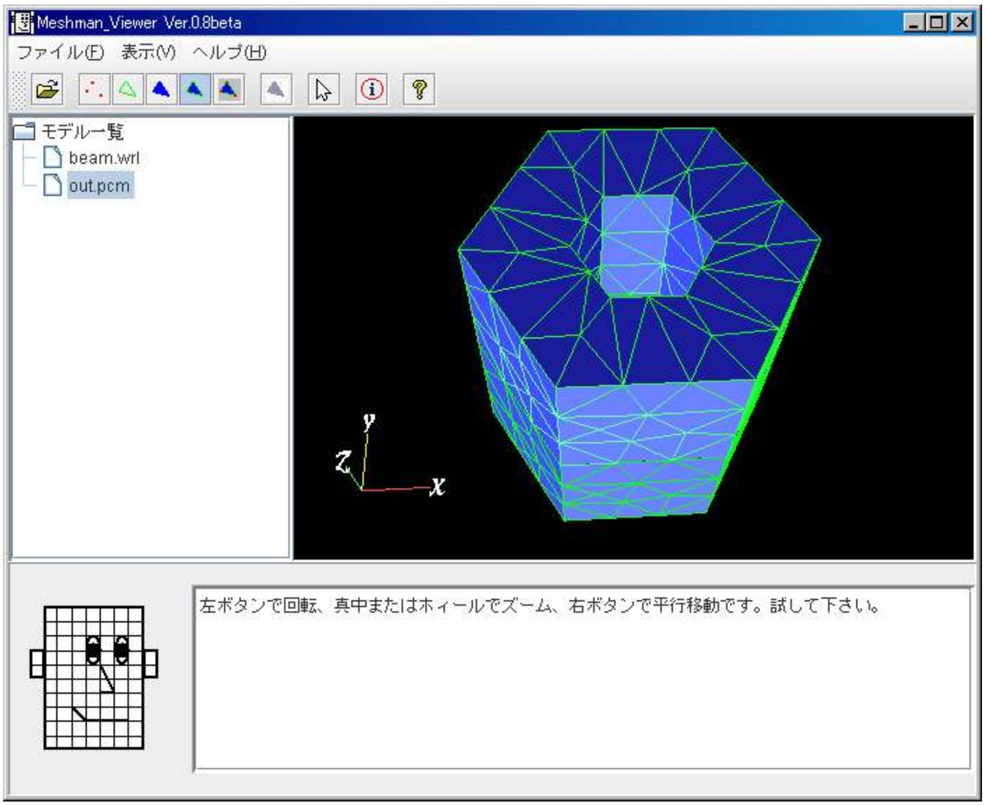


Fig. 4.1-3 The shape made by merging the lowerNut and the upperNut

modelA.pcm, modelA.pcg, modelB.pcm, modelB.pcg, out.wrl and out.pcg will also be generated as a by-product although not listed up above.

Then modify the surface patch by the advtmesh9p command.

```
% advtmesh9p out -base1. -p
```

The following three files will be created as output files of the AdvTetMesh_P.

samples/PcmMerge/doubleNut/outc.pcc

samples/PcmMerge/doubleNut/outc.ptn

samples/PcmMerge/doubleNut/out_c.wrl

Then, generate linear tetrahedral elements by the advtmesh9m command.

```
% advtmesh9m outc -p
```

The following three files will be created as output files of the AdvTetMesh_M.

samples/PcmMerge/doubleNut/outc.msh

samples/PcmMerge/doubleNut/outc_e.wrl

samples/PcmMerge/doubleNut/outc_n.wrl

Then generate quadratic tetrahedral elements by the advtmesh9s command.

```
% advtmesh9s outc
```

The following file will be created as an output file of the AdvTetMesh_S.

samples/PcmMerge/doubleNut/outcs.msh

The MergeCheck command below will separate the merged nodes in the linear tetrahedral element mesh.

```
% MergeCheck outc.msh moveInfo.dat doubleNutOutLinear
```

Then three output files of the MergeCheck will be generated.

samples/PcmMerge/doubleNut/doubleNutOutLinear.msh

samples/PcmMerge/doubleNut/doubleNutOutLinear.np

samples/PcmMerge/doubleNut/doubleNutOutLinear.nv

They are a two volume linear tetrahedral element mesh data file with inter-volume boundary nodes split, a file of the list of node pairs on the corresponding boundary surfaces, and a file of the normal vectors at each node on the boundary surfaces respectively.

As shown below, the MergeCheck command will separate the merged nodes in the quadratic tetrahedral element mesh.

```
% MergeCheck outcs.msh moveInfo.dat doubleNutOutQuadratic
```

Then three output files of the MergeCheck are generated.

samples/PcmMerge/doubleNut/ doubleNutOutQuadratic.msh

samples/PcmMerge/doubleNut/ doubleNutOutQuadratic.np

samples/PcmMerge/doubleNut/ doubleNutOutQuadratic.nv

They are a two volume quadratic tetrahedral element mesh data file with inter-volume boundary nodes split, a file of the list of node pairs on the corresponding boundary surfaces, and a file of the normal vectors at each node on the boundary surfaces respectively.

4.2 Bricks on A Disk

The following files are available as examples.

PcmMerge Input files

samples/PcmMerge/brickOnDisk/body1.pcm

samples/PcmMerge/brickOnDisk/contact/body2Contact.pcm

samples/PcmMerge/brickOnDisk/interfere/body2Interfere.pcm

samples/PcmMerge/brickOnDisk/separate/body2Separate.pcm

The coordinate range of the body1.pcm, the body2Contact.pcm, the body2Interfere.pcm, and the body2Separate.pcm are (-4, -4, 0) to (4, 4, 3), (-1, -2, 3) to (1, 2, 6), (-1, -2, 2.9) to (1, 2, 6), and (-1, -2, 3.5) to (1, 2, 6) respectively. Geometry is shown in Fig. 4.2-1 through Fig. 4.2-24.

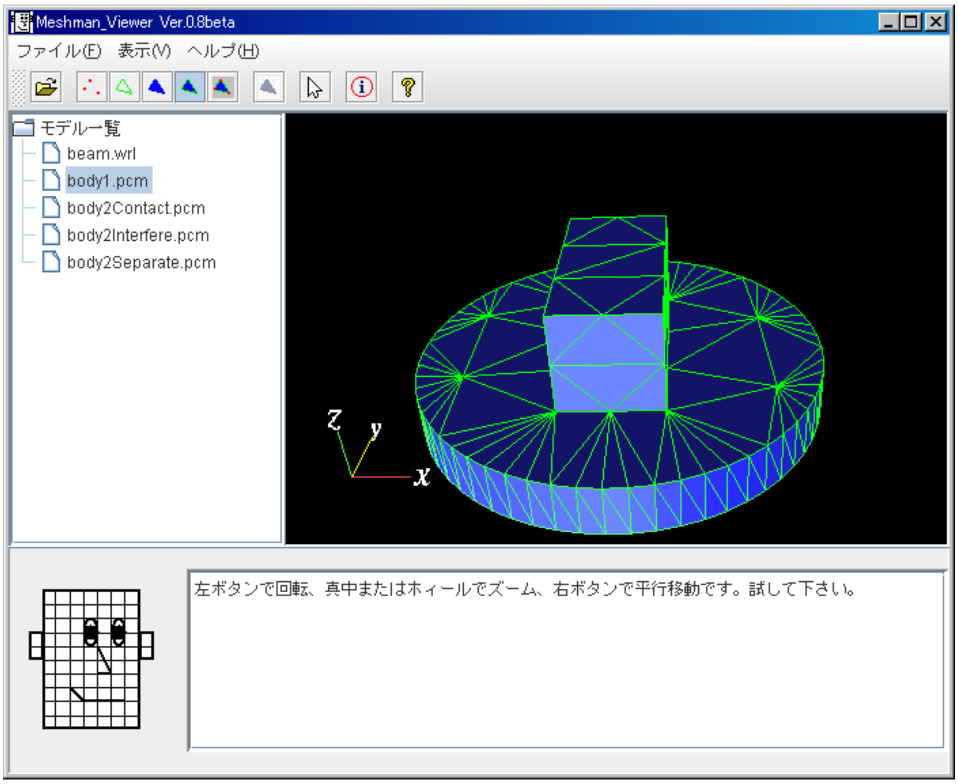


Fig. 4-2-1 Geometry of the body1.pcm

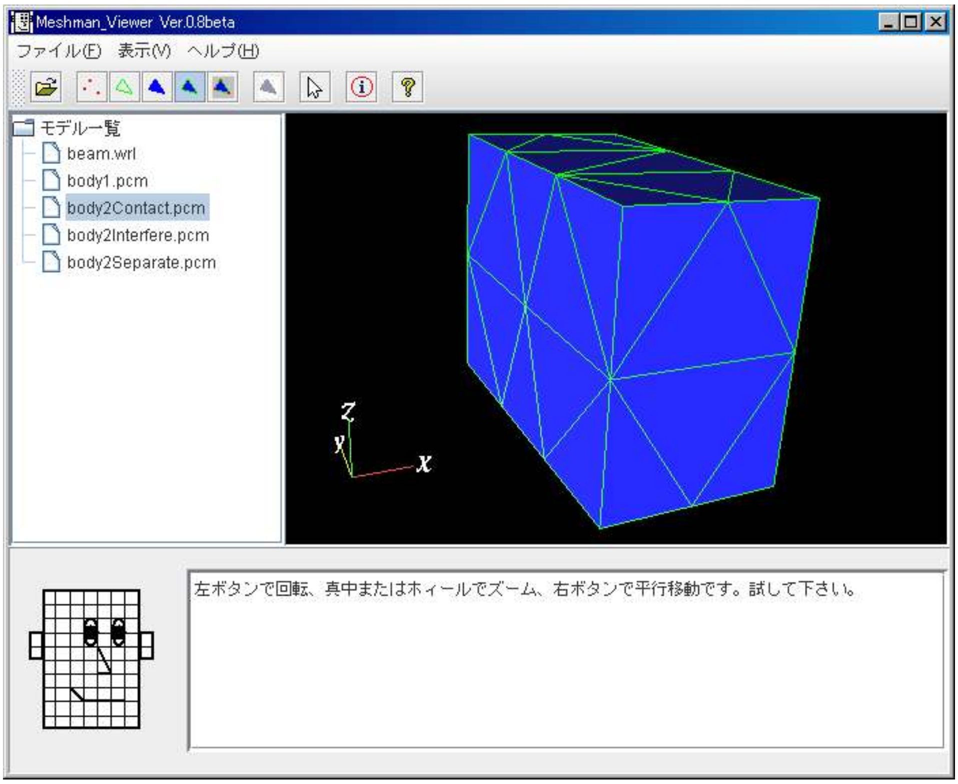


Fig. 4-2-2 Geometry of the body2Contact.pcm

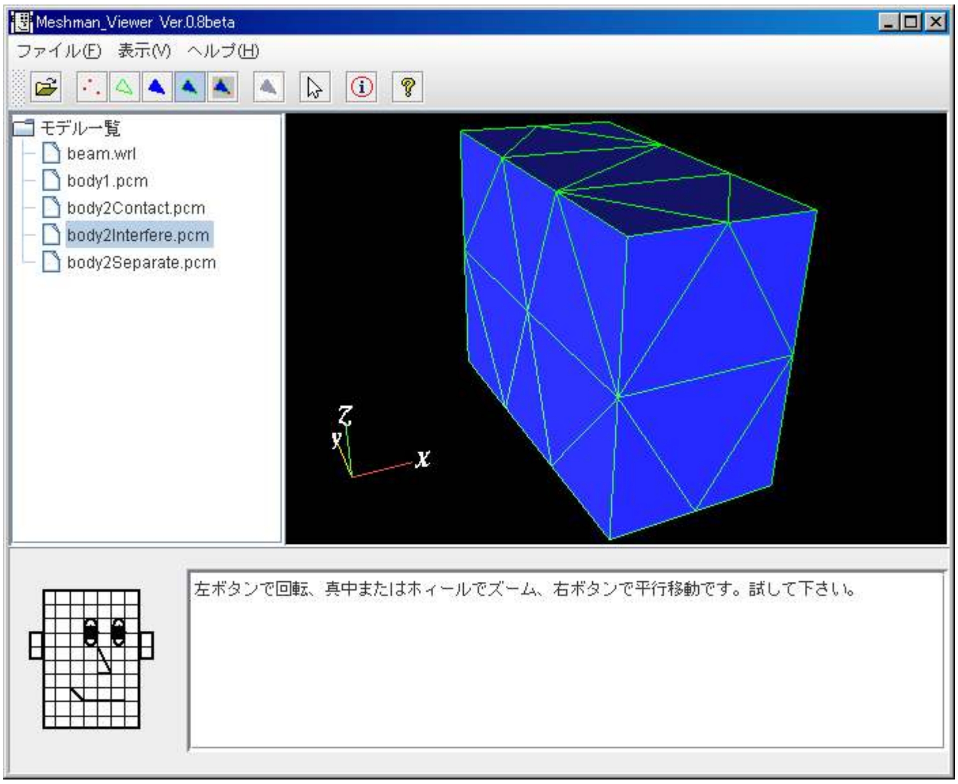


Fig. 4-2-3 Geometry of the body2Interfere.pcm

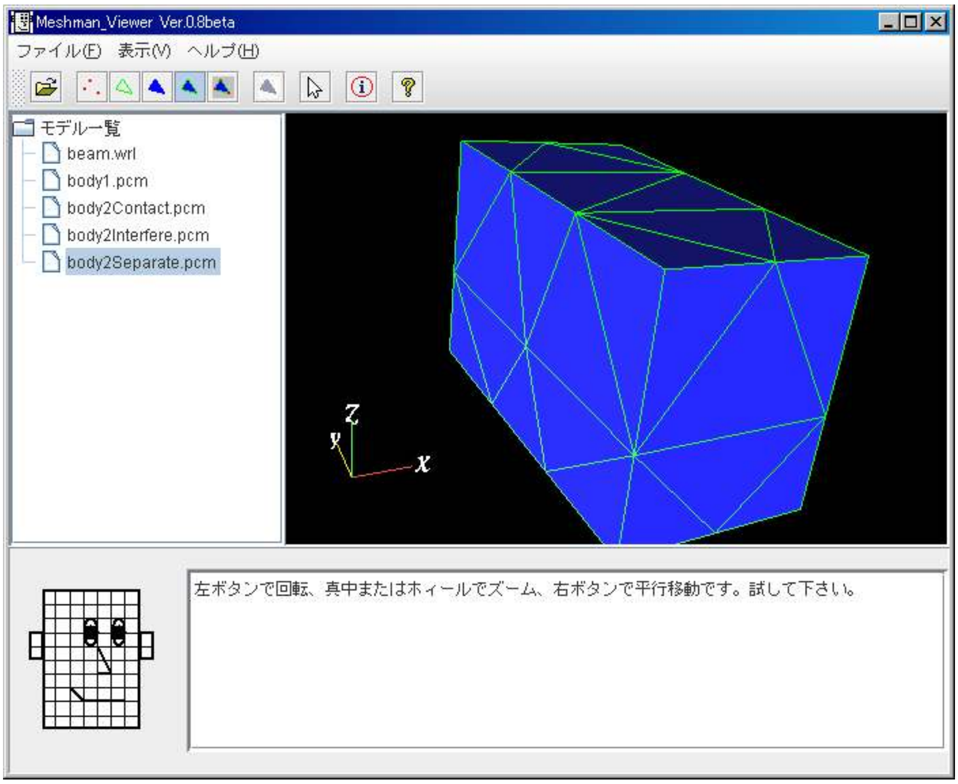


Fig. 4-2-4 Geometry of the body2Separate.pcm

The body1.pcm and the body2Contact.pcm are in contact. If the next command is executed in the folder samples/PcmMerge/brickOnDisk/contact/,

```
% PcmMerge ../body1.pcm body2Contact.pcm
```

the following two files and six by-product files will be output.

samples/PcmMerge/brickOnDisk/contact/out.pcm

samples/PcmMerge/brickOnDisk/contact/moveInfo.dat

Fig. 4.2-5 shows the geometry of the out.pcm. It has a shape made by merging the body1 and the body2Contact.

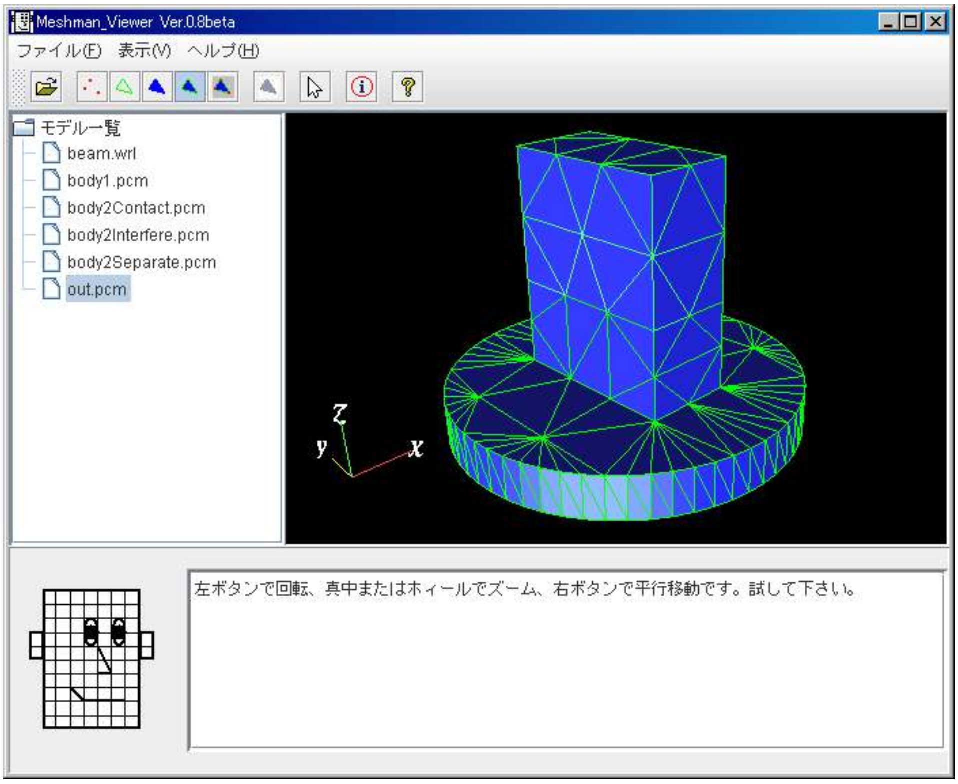


Fig. 4-2-5 The shape of out.pcm in the case where bodies are initially in contact

The body1.pcm and the body2Interfere.pcm have interference in the range of z equal to 2.9 to 3. If the following command is run in the folder samples/brickOnDisk/interfere/,

```
% PcmMerge ../body1.pcm body2Interfere.pcm
```

the following two files and six by-product files will be output.

samples/PcmMerge/brickOnDisk/interfere/out.pcm

samples/PcmMerge/brickOnDisk/interfere/moveInfo.dat

The shape of the out.pcm looks very similar to that shown in Fig. 4.2-5, but there is no interference.

The body1.pcm and the body2Separate.pcm have a gap in the range of z equal to 3 to 3.5. If the following command is run in the folder samples/PcmMerge/brickOnDisk/separate/,

```
% PcmMerge ../body1.pcm body2Separate.pcm
```

the following two files and six by-product files will be output.

samples/PcmMerge/brickOnDisk/separate/out.pcm

samples/PcmMerge/brickOnDisk/separate/moveInfo.dat

Since the out.pcm's gap is closed, its geometry is shorter than that in Fig. 4.2-5 as shown in Fig. 4.2-6.

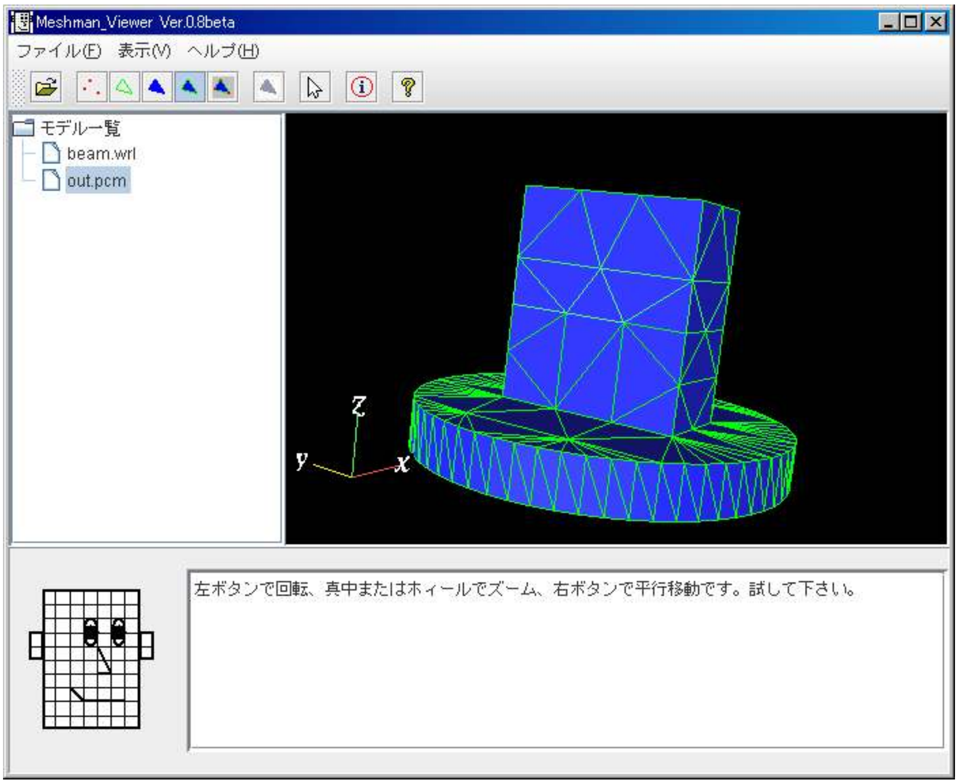


Fig. 4-2-6 The shape of the out.pcm in the case of initially separate bodies

Tetrahedra will be generated in the folders contact, interfere, and separate by means of the following command.

```
% advtmesh9p out -base1. -p
% advtmesh9m outc -p
```

If quadratic elements are needed,

```
% advtmesh9s outc
```

must be executed. In each folder, the following files will be created.

outc.pcc

outc.ptn

out_c.wrl

outc.msh

outc_e.wrl

outc_n.wrl

outcs.msh

The MergeCheck command will be executed in the folders contact, interfere, and separate and will divide the merged nodes in the quadratic tetrahedral element mesh.

```
% MergeCheck outcs.msh moveInfo.dat brickOnDiskOutQuadratic
```

Then three output files of the MergeCheck will be generated.

brickOnDiskOutQuadratic.msh

brickOnDiskOutQuadratic.np

brickOnDiskOutQuadratic.nv

Fig. 4.2-7 shows the brickOnDiskOutQuadratic.msh in the case of contact. Two volumes are in contact.

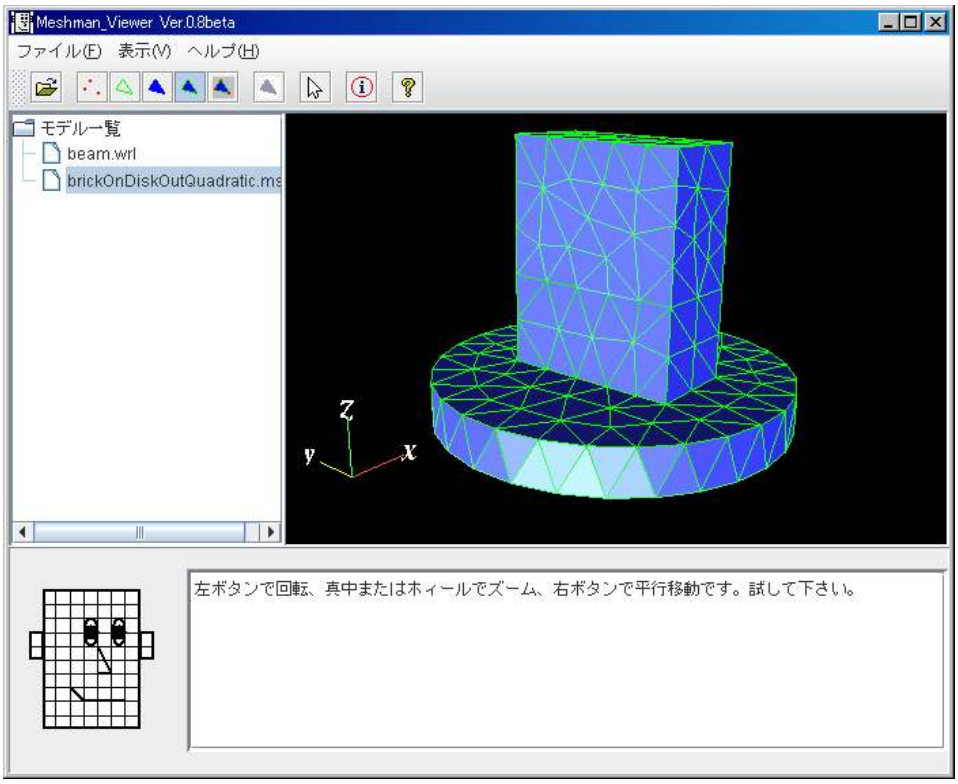


Fig. 4.2-7 The shape of brickOnDiskOutQuadratic.msh in the case of initially contacted bodies

Fig. 4.2-8 shows the brickOnDiskOutQuadratic.msh in the case of interference. The two volumes are in contact.

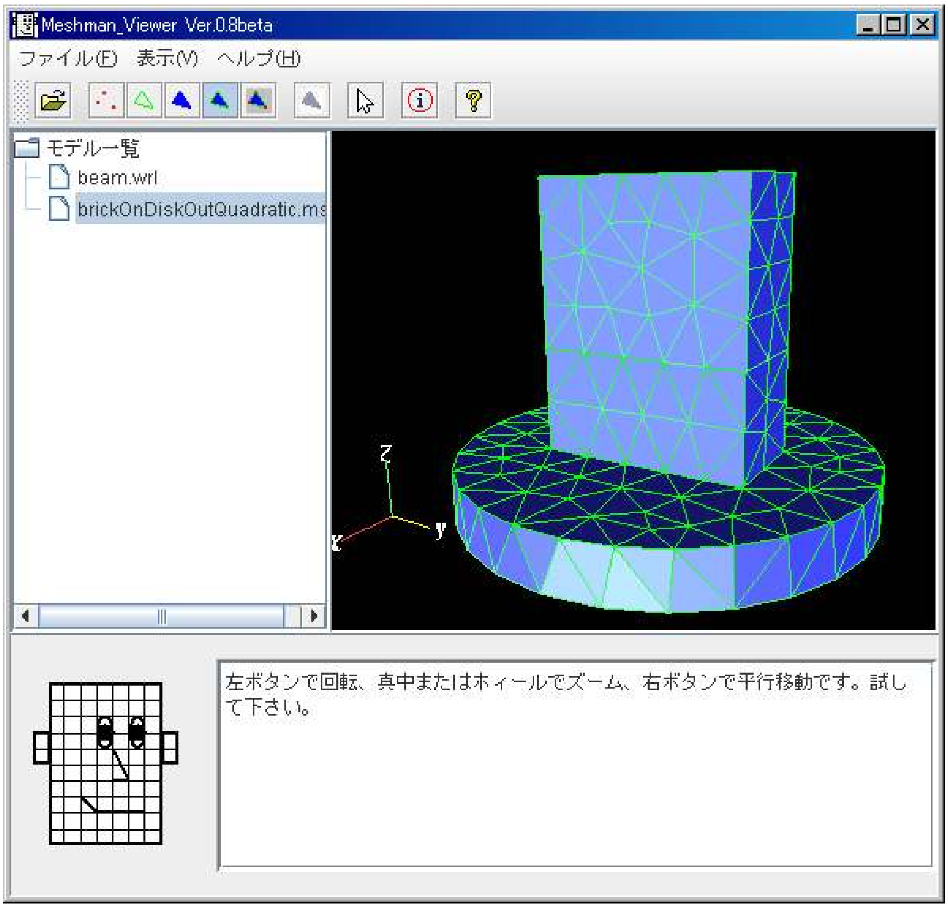


Fig. 4.2-8 The shape of the brickOnDiskOutQuadratic.msh in the case of initially interfered bodies

Fig. 4.2-9 shows the brickOnDiskOutQuadratic.msh in the case of separate. The two volumes are separated back again.

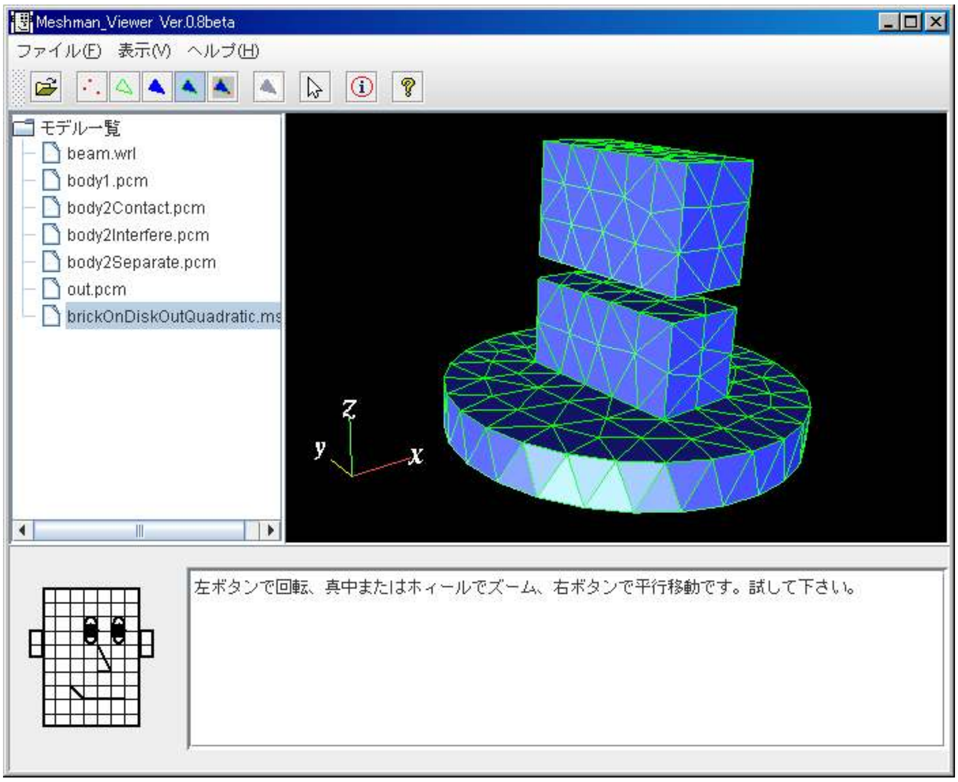


Fig. 4.2-9 The shape of the brickOnDiskOutQuadratic.msh in the case of initially separate bodies.

Part III The Manual of the MpcMasterSlaveTool

1. Summary

The MPC_MasterSlave is a tool to write a file of MPC, multi-point constraints between two surface groups of different shapes and sizes.

2. Functions

It has the function of combining two meshes (parts), and generating MPC conditions automatically for the two specified sides of the synthesized multiple volume mesh. At that time, if the two specified sides are in contact with each other, their shape, size, and mesh need not be the same.

Supported type of elements:

Linear tetrahedral element, and quadratic tetrahedral element

Supported type of the MPC:

Parts are fixed to each other

The positions where MPC conditions are applicable

A pair of two surface groups in contact of a mesh with multiple volumes

3. Operating Environments

OS	UNIX, Linux
Compiler	GNU C++ 4.1.2 was used for an operation test
Libraries	Not applicable

4. How to Install

Please create the executable files in the src folder with the following command.

```
% make
```

Edit the fourth line of the src/Makefile as appropriate.

```
PREFIX   = $(HOME)/ADVENTURE
```

And then, type the next command to install the executables.

```
% make install
```

If the tool is installed in a system area, become the root by the su command in advance or use the sudo command. It should be noted that the executable file and the intermediate files that were created in the src folder at the time of compilation can be deleted by the following command.

```
% make clean
```

5. How to Use the Commands

To combine mesh data files, type the following command.

```
% MPC_mshmrg.pl xxx.msh yyy.msh zzz.msh
```

where

xxx.msh is an input mesh data file 1.

yyy.msh is an input mesh data file 2.

zzz.msh is an output mesh data file.

The following command will create an MPC condition.

```
% MPC_assem2 aaa.msh bbb.fgr ccc.cmb > ddd.mpc
```

where

aaa.msh is an input mesh data file.

bbb.fgr is an input file. It should be created by the msh2pch from the aaa.msh.

ccc.cmb is an input file. It is a configuration file to set surface group pairs defined in the Appendix Section A.1.12.

ddd.mpc is an output MPC definition file.

Then write "LinearConstraint <number of lines>" at the beginning of the ddd.mpc after examining the number of lines of the ddd.mpc by a text editor.

6. Examples

Files in the samples folder is a result of the process according to the data flow in Fig. 5.1-2, Chapter 5, Part I.

6.1 2box

This is an example of two cubes of different sizes and in contact.

Commands and a brief description will be given below.

Combining mesh data files

Make MPC_merged.msh using the leftc.msh and the rightc.msh which are the mesh data files prepared in advance. Here, type the following command.

```
% MPC_mshmrg.pl leftc.msh rightc.msh
```

Then, a merged mesh data file MPC_merged.msh will be output. The configuration of the MPC_merged.msh is shown in Fig. 6.1-1. A small cube. in Fig. 6.1-1 comes from the leftc.msh and the large cube comes from the rightc.msh.

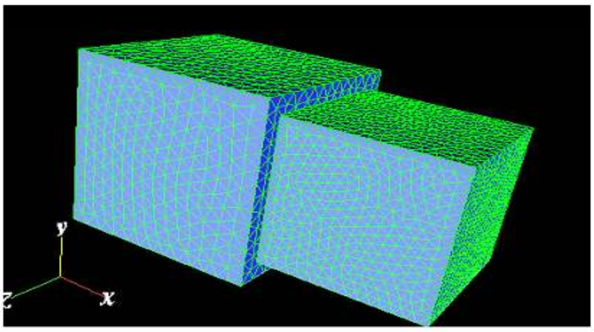


Fig. 6.1-1 The configuration of the MPC_merged.msh for 2box

Extraction of the surface of the mesh

By means of the msh2pch, the surface of the mesh will be extracted from MPC_merged.msh and grouped, to convert to the input format of the GUI. Here, let the name of the mesh data file MPC_merged.msh and let the dihedral angle 60, i.e. 180/3, and type the following command.

```
% msh2pch MPC_merged.msh 3
```

The files below will be output.

MPC_merged_3.fgr: A mesh surface data file

MPC_merged_3.pch: An extracted surface mesh data file

MPC_merged_3.pcg: A surface patch group data file

MPC_merged_3.trn: A global index file

Creation of MPCs

By means of the MPC_assem2, MPCs are created that combine sides with different sets of nodal coordinates based on a mesh data file, a mesh surface data file, and the cmb file. At this time, a pair of surface group numbers 5 and 8 should be written in the combi.cmb file. Here, enter the following command.

```
% MPC_assem2 MPC_merged.msh MPC_merged_3.fgr combi.cmb > MPC_merged_3.mpc
```

Write "LinearConstraint number of lines" at the beginning of the MPC_merged_3.mpc after examining its number of lines using a text editor.

Creation of an adv file by the a2adv

By means of the adv.pl, an integrated analysis model file with the ADVENTURE_IO format will be output based on the MPC file. Here, type the following command.

```
% a2adv.pl MPC_merged_3.mpc LinearConstraint.adv
```

LinearConstraint.adv will be output.

Settings of boundary conditions (loads and constraints)

Boundary conditions are to be set by means of the BcGUI 2.0. Select a node or a surface group to add a boundary condition. Then, selecting from the menu bar BC->BC(Solid)->Add SurfaceTraction (for a load condition) or BC->BC(Solid)->Add Displacement (for constrains or forced displacements) will show a dialog for setting boundary conditions. In this case, select the surface group number 11 as shown in Fig. 6.1-2, and select BC->BC(Solid)->Add SurfaceTraction from the menu bar. Then select the check button to the left of the "X" in the dialog, type "10" in the text box to its right, and click the "OK" button (on the surface group number 11, an load with an intensity of 10 is applied in the X direction). Similarly, select the surface group 2 as shown in Fig. 6.1-3, select BC->BC(Solid)->Add Displacement from the menu bar, select the check buttons to the left of the "X", "Y", and "Z", and enter "0"s in all the text boxes to their right (displacements of the surface group number 2 are constraint in all directions).

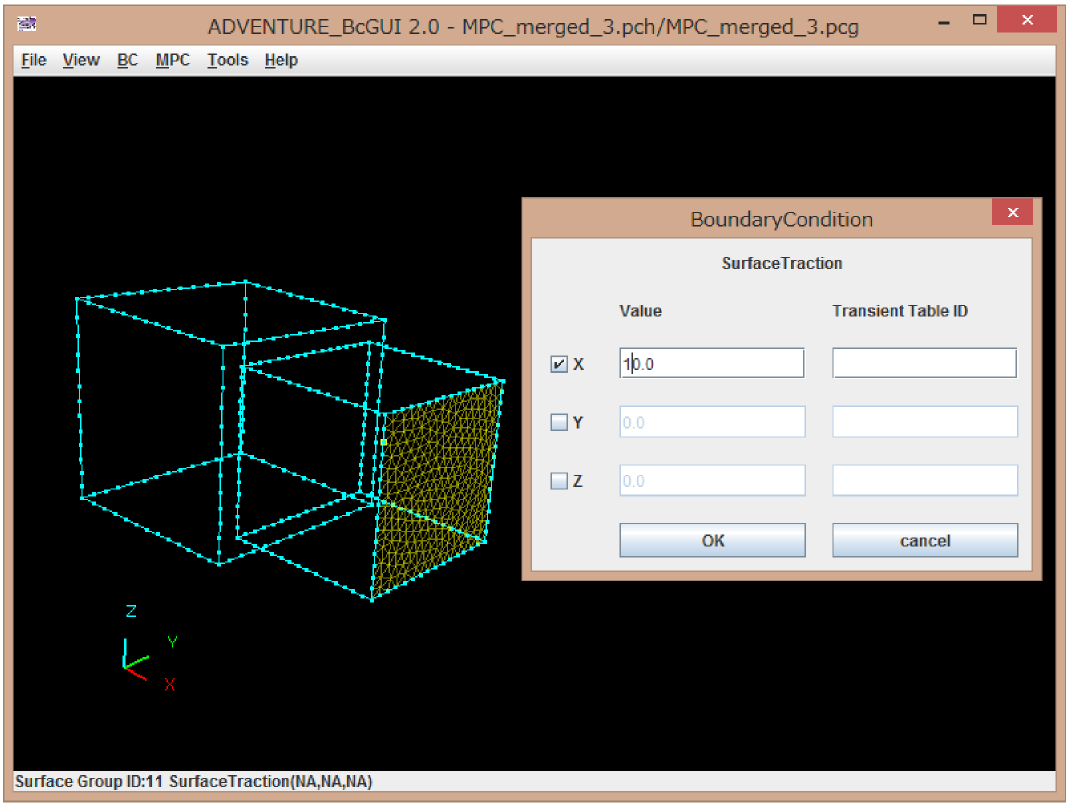


Fig. 6.1-2 Selection of the surface group #11 and setting of load conditions

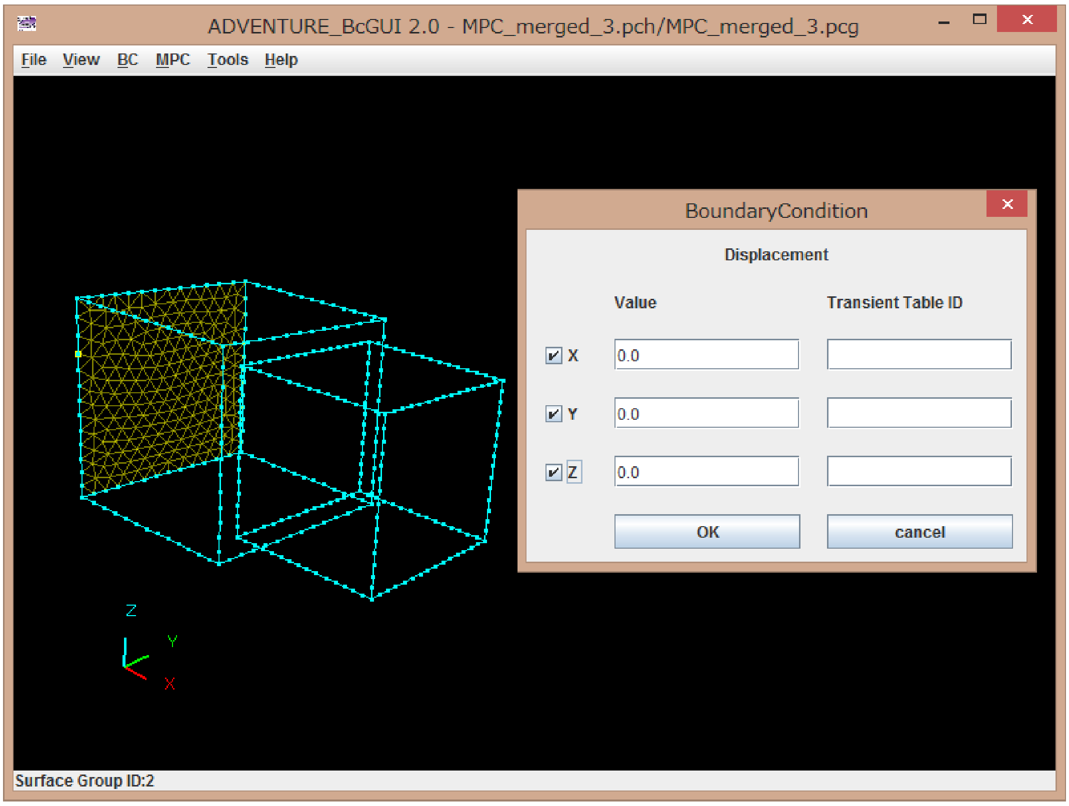


Fig. 6.1-3 Selection of the surface group #2 and setting of displacement constraints

Setting of material properties

Create a material properties data file using a text editor. Here, name it, MPC_merged_3.mat, and let the Young's modulus 21000.0, and the Poisson's ratio 0.3 and type as follows.

```
YoungModulus 21000.0
PoissonRatio 0.3
```

Creation of an adv file by the makefem3

By means of the makefem3, boundary conditions and material properties will be attached to the mesh, and an integrated analysis model file with the ADVENTURE_IO format will be created. Here, type the following command.

```
% makefem3 MPC_merged.msh MPC_merged_3.fgr MPC_merged_3.cnd MPC_merged_3.mat
MPC_merged_3.adv -t MPC_merged_3.trn
```

Combining adv files

By means of the advcat, the adv files made by the a2adv and the makefem3 will be combined. Here, the following command will be entered.

```
% advcat MPC_merged_3.adv LinearConstraint.adv 2box.adv
```

test_merged.adv will be output.

6.2 bigplate

This is an example of nine rectangular pillars between two big plates.

Commands and a brief description will be given below.

Combining mesh data files

```
% MPC_mshmrg.pl platecs.msh columncs.msh
```

The configuration of the output MPC_merged.msh is shown in Fig. 6.2-1. The two large plates in Fig. 6.2-1 come from the platecs.msh, and the rectangular nine pillars come from columncs.msh.

Extraction of the surface of the mesh

```
% msh2pch MPC_merged.msh 3
```

Creation of MPCs

Total of 18 pairs of surface groups described in the combi.cmb consist of 2 and 27, 2 and 51, 2 and

57, 2 and 21, 2 and 45, 2 and 63, 2 and 15, 2 and 33, 2 and 39, 9 and 25, 9 and 49, 9 and 55, 9 and 19, 9 and 42, 9 and 61, 9 and 12, 9 and 31, and 9 and37.

```
% MPC_assem2 MPC_merged.msh MPC_merged_3.fgr combi.cmb > MPC_merged_3.mpc
```

Write "LinearConstraint number of lines" at the beginning of the MPC_merged_3.mpc after examining its number of lines.

Creation of an adv file by the a2adv

```
% a2adv.pl MPC_merged_3.mpc LinearConstraint.adv
```

Settings of boundary conditions (forced displacements and constraints)

```
% BcGUI MPC_merged_3.pch MPC_merged_3.pcg -ocnd MPC_merged_3.cnd
```

Boundary conditions will be set as shown in Fig. 6.2-2 and Fig. 6.2-3.

Setting of material properties

MPC_merged_3.mat will be created with the Young's modulus equal to 21000.0 and the Poisson's ration equal to 0.3.

Creation of an adv file by the makefem3

```
% makefem3 MPC_merged.msh MPC_merged_3.fgr MPC_merged_3.cnd MPC_merged_3.mat
MPC_merged_3.adv -t MPC_merged_3.trn
```

Combining adv files

```
% advcat MPC_merged_3.adv LinearConstraint.adv bigplate.adv
```

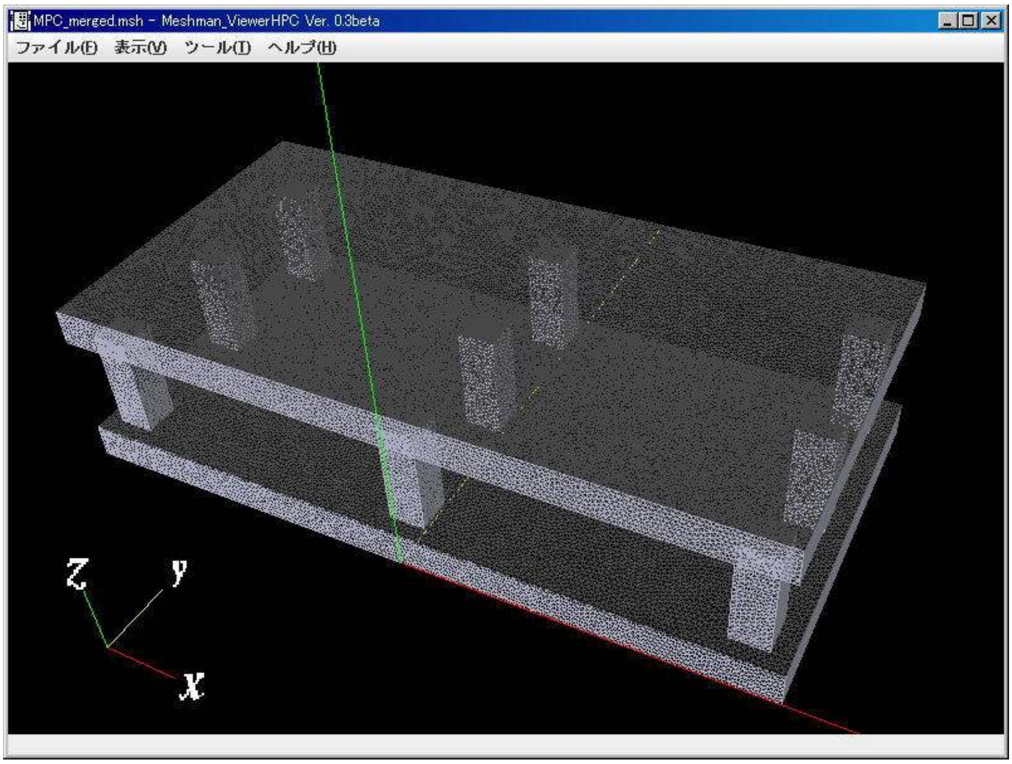


Fig. 6.2-1 The configuration of the MPC_merged.msh for bigplate

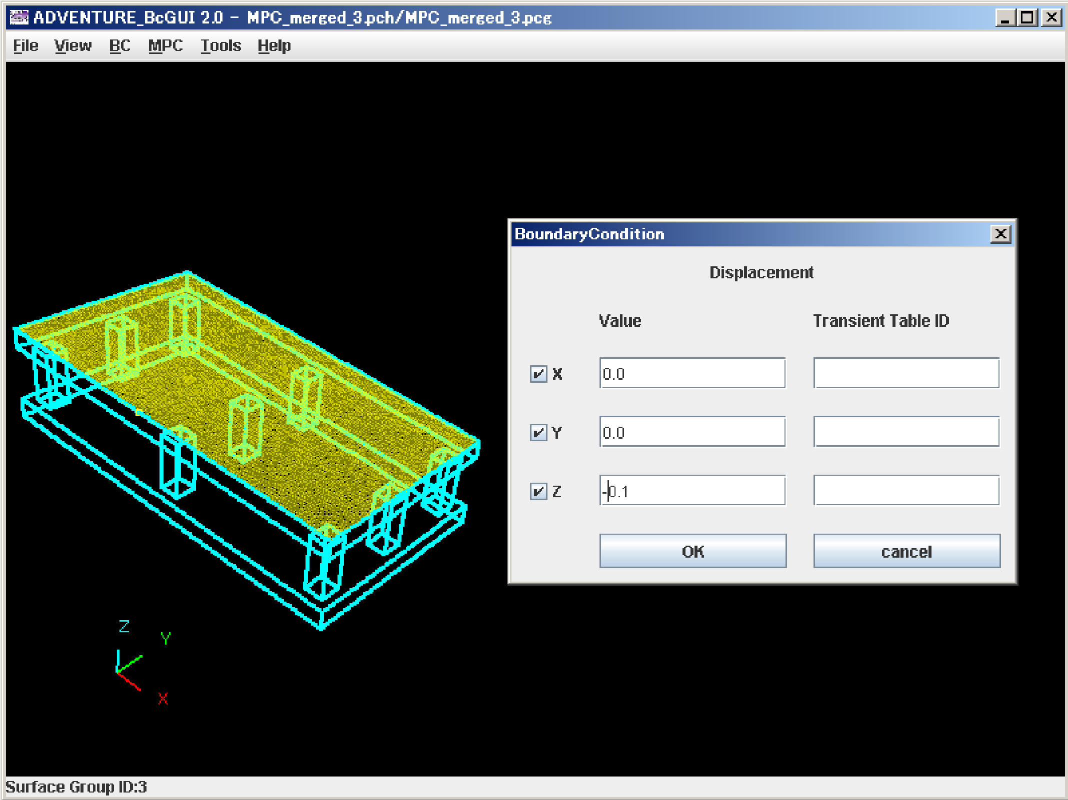


Fig. 6.2-2 Selection of the surface group #3 and setting of forced displacements

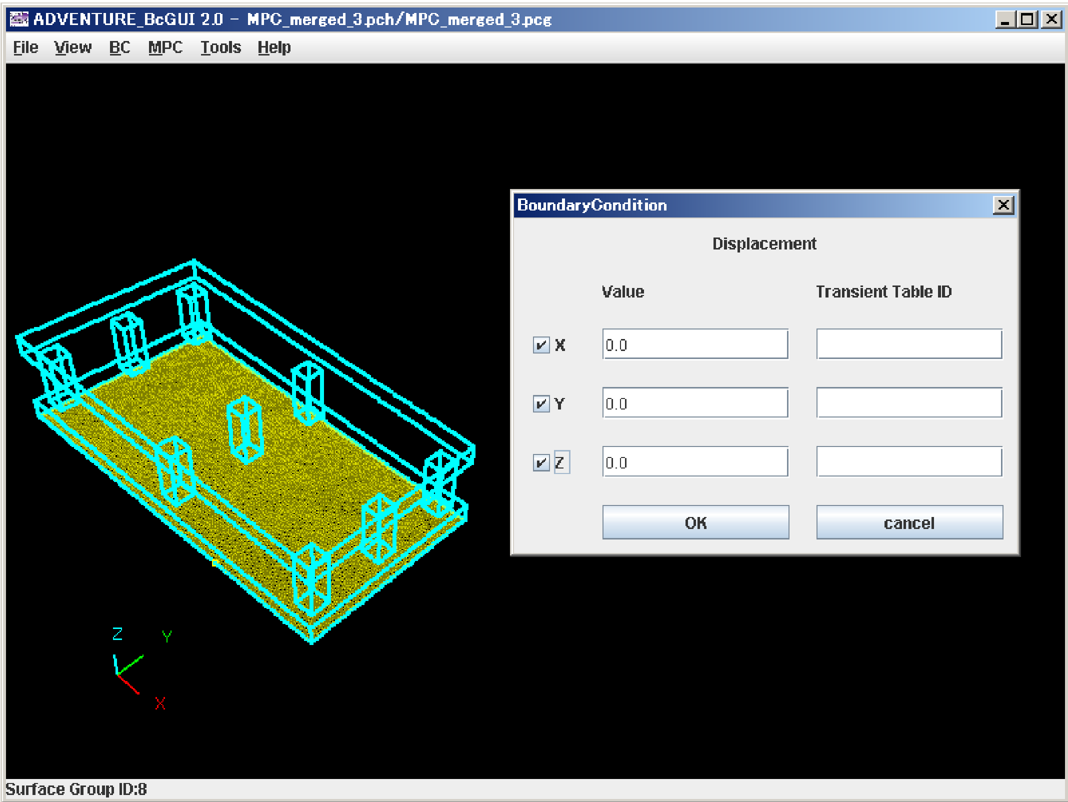


Fig. 6.2-3 Selection of the surface group #8 and setting of displacement constraints

6.3 plate

This is an example of nine rectangular pillars between two plates.

Commands and a brief description will be given below.

Combining mesh data files

```
% MPC_mshmrg.pl platecs.msh columncs.msh
```

The configuration of the output MPC_merged.msh is shown in Fig. 6.3-1. The two large plates in Fig. 6.3-1 come from the platecs.msh, and the rectangular nine pillars come from columncs.msh.

Extraction of the surface of the mesh

```
% msh2pch MPC_merged.msh 3
```

Creation of MPCs

Total of 18 pairs of surface groups described in the combi.cmb consist of 2 and 29, 2 and 53, 2 and 59, 2 and 23, 2 and 47, 2 and 65, 2 and 17, 2 and 35, 2 and 41, 9 and 27, 9 and 50, 9 and 55, 9 and 20, 9 and 44, 9 and 60, 9 and 15, 9 and 32, and 9 and38.

```
% MPC_assem2 MPC_merged.msh MPC_merged_3.fgr combi.cmb > MPC_merged_3.mpc
```

Write "LinearConstraint number of lines" at the beginning of the MPC_merged_3.mpc after examining its number of lines.

Creation of an adv file by the a2adv

```
% a2adv.pl MPC_merged_3.mpc LinearConstraint.adv
```

Settings of boundary conditions (forced displacements and constraints)

```
% BcGUI MPC_merged_3.pch MPC_merged_3.pcg -ocnd MPC_merged_3.cnd
```

Boundary conditions will be set as shown in Fig. 6.3-2 and Fig. 6.3-3.

Setting of material properties

MPC_merged_3.mat will be created with the Young's modulus equal to 21000.0 and the Poisson's ration equal to 0.3.

Creation of an adv file by the makefem3

```
% makefem3 MPC_merged.msh MPC_merged_3.fgr MPC_merged_3.cnd MPC_merged_3.mat
MPC_merged_3.adv -t MPC_merged_3.trn
```

Combining adv files

```
% advcat MPC_merged_3.adv LinearConstraint.adv plate.adv
```

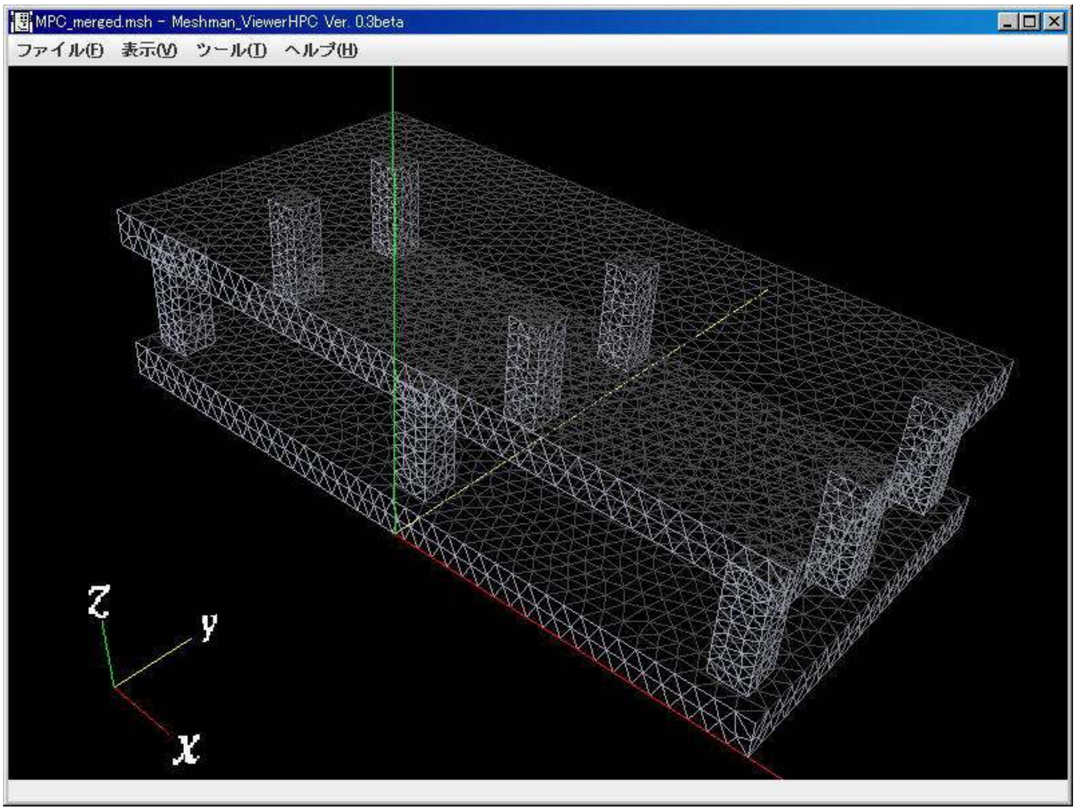


Fig. 6.3-1 The configuration of the MPC_merged.msh for plate

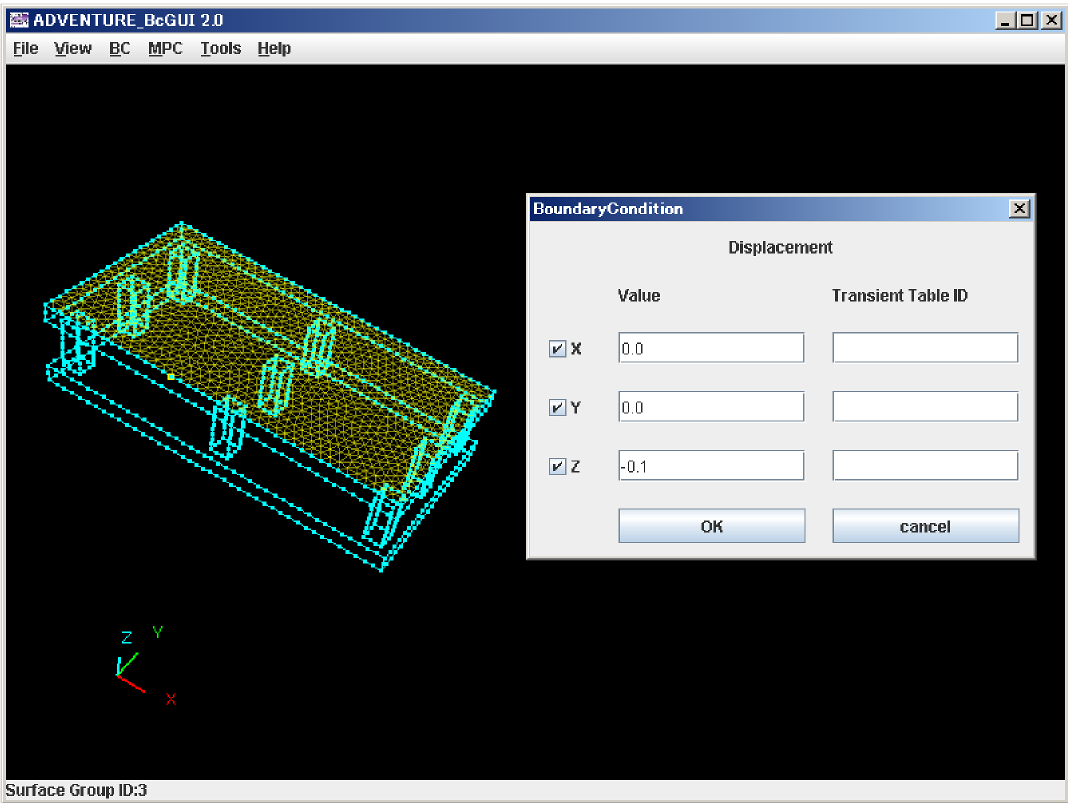


Fig. 6.3-2 Selection of the surface group #3 and setting of forced displacements

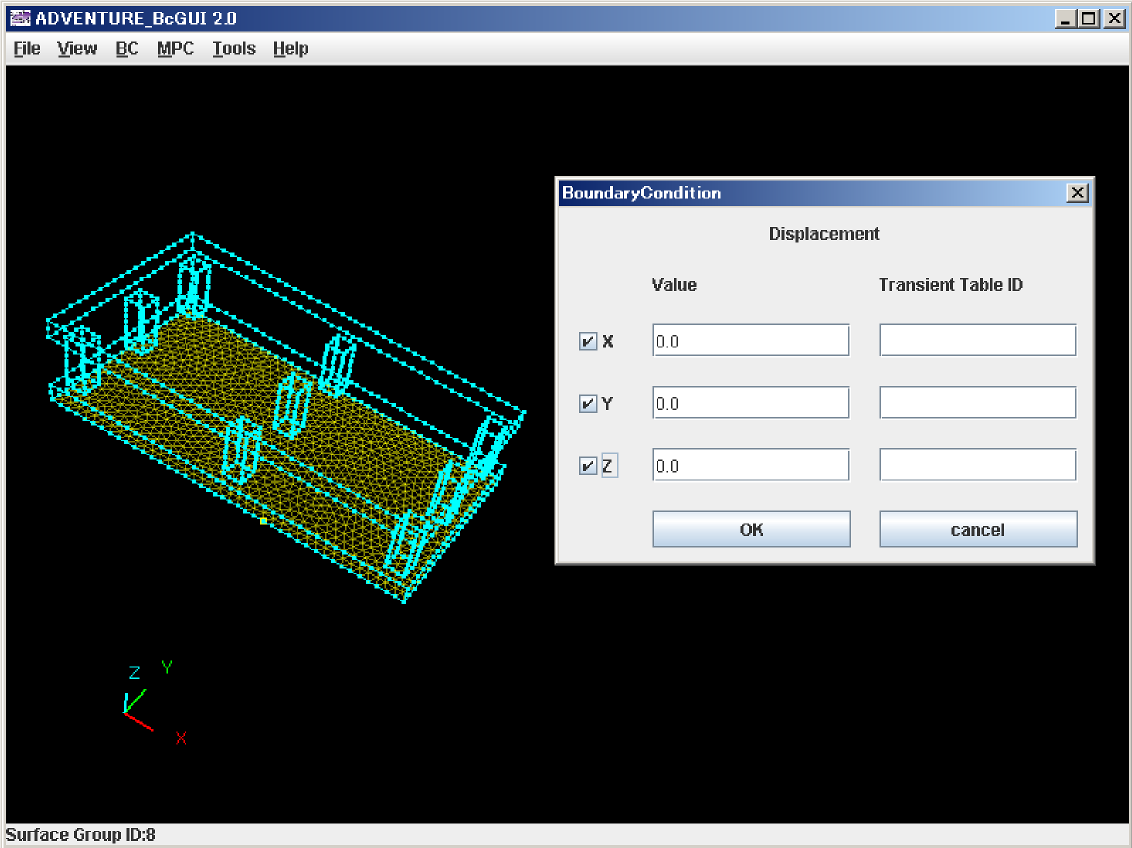


Fig. 6.3-3 Selection of the surface group #8 and setting of displacement constraints

6.4 qring

This is an example of a quarter of a small ring in contact with and inside a larger ring.

Commands and a brief description will be given below.

Combining mesh data files

```
% MPC_mshmrg.pl qring.msh qouter.msh
```

The configuration of the output MPC_merged.msh is shown in Fig. 6.4-1. A larger ring in Fig. 6.4-1 comes from the qouter.msh , and the smaller ring inside comes from the qring.msh.

Extraction of the surface of the mesh

```
% msh2pch MPC_merged.msh 3
```

Creation of MPCs

The number of pairs of surface groups to be specified in the combi.cmb is 1 and that is 11 and 0.

```
% MPC_assem2 MPC_merged.msh MPC_merged_3.fgr combi.cmb > MPC_merged_3.mpc
```

Write "LinearConstraint number of lines" at the beginning of the MPC_merged_3.mpc after examining its number of lines.

Creation of an adv file by the a2adv

```
% a2adv.pl MPC_merged_3.mpc LinearConstraint.adv
```

Settings of boundary conditions (loads, symmetric conditions, and constraints)

```
% BcGUI MPC_merged_3.pch MPC_merged_3.pcg -ocnd MPC_merged_3.cnd
```

Boundary conditions will be set as shown in Fig. 6.4-2 through Fig. 6.4-8.

Setting of material properties

MPC_merged_3.mat will be created with the Young's modulus equal to 21000.0 and the Poisson's ration equal to 0.3.

Creation of an adv file by the makefem3

```
% makefem3 MPC_merged.msh MPC_merged_3.fgr MPC_merged_3.cnd MPC_merged_3.mat
MPC_merged_3.adv -t MPC_merged_3.trn
```

Combining adv files

```
% advcat MPC_merged_3.adv LinearConstraint.adv qring.adv
```

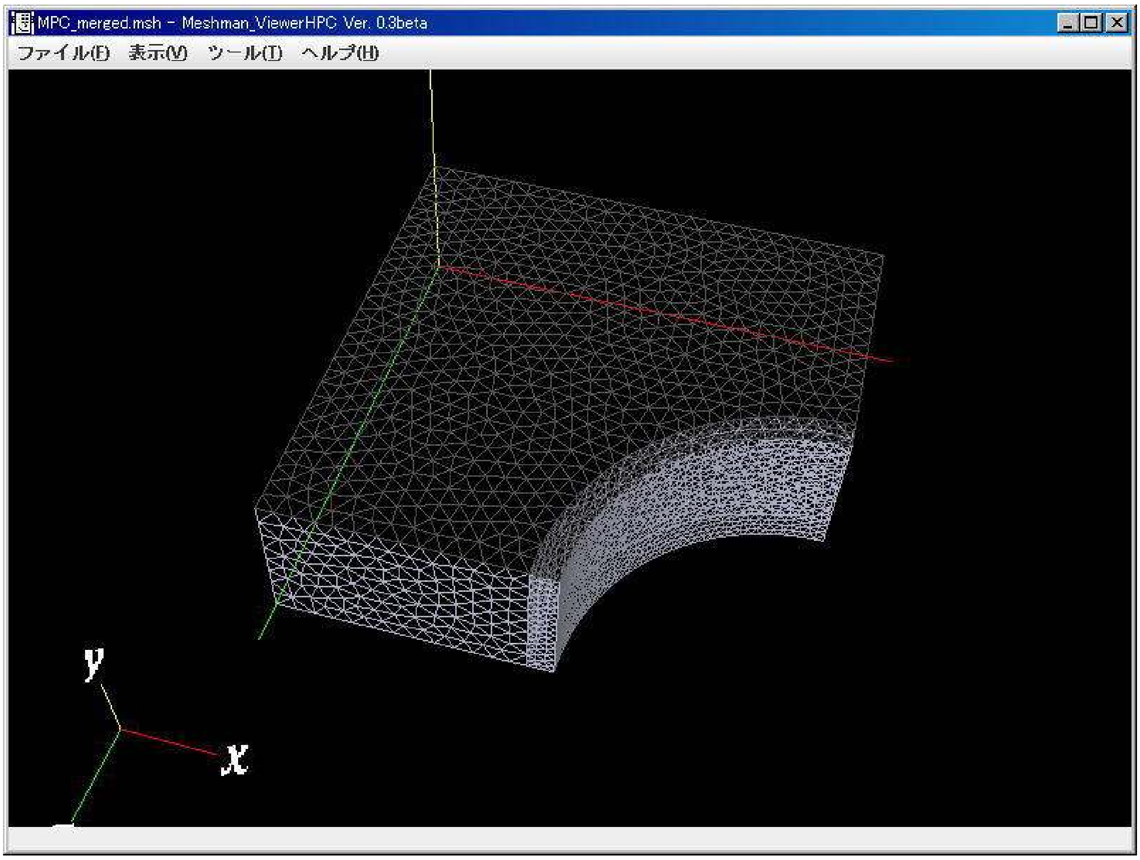


Fig. 6.4-1 The configuration of the MPC_merged.msh for qring

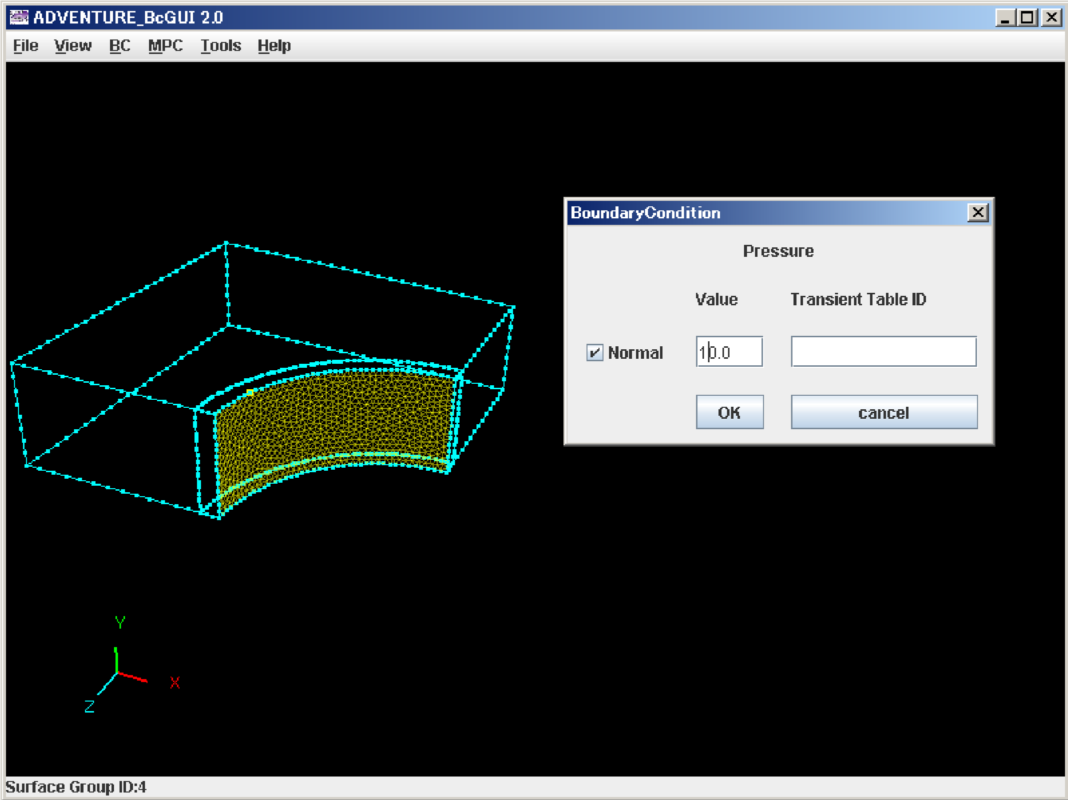


Fig. 6.4-2 Selection of the surface group #4 and setting of pressure conditions

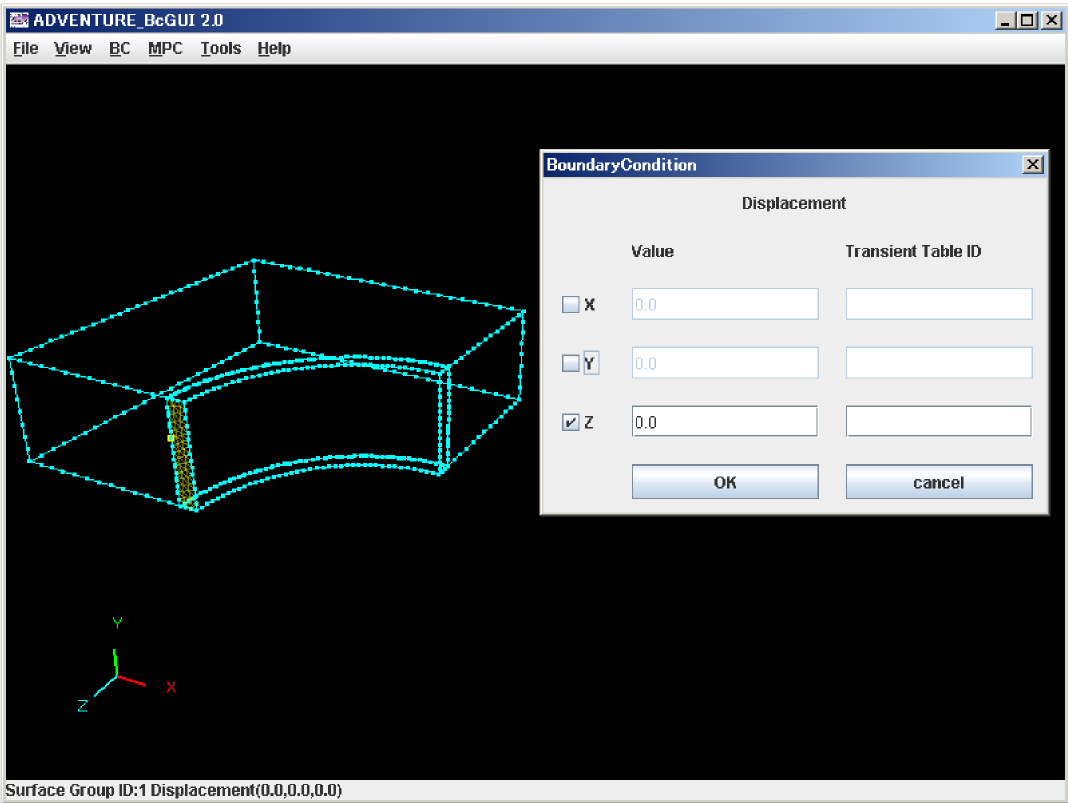


Fig. 6.4-3 Selection of the surface group #1 and setting of symmetric conditions

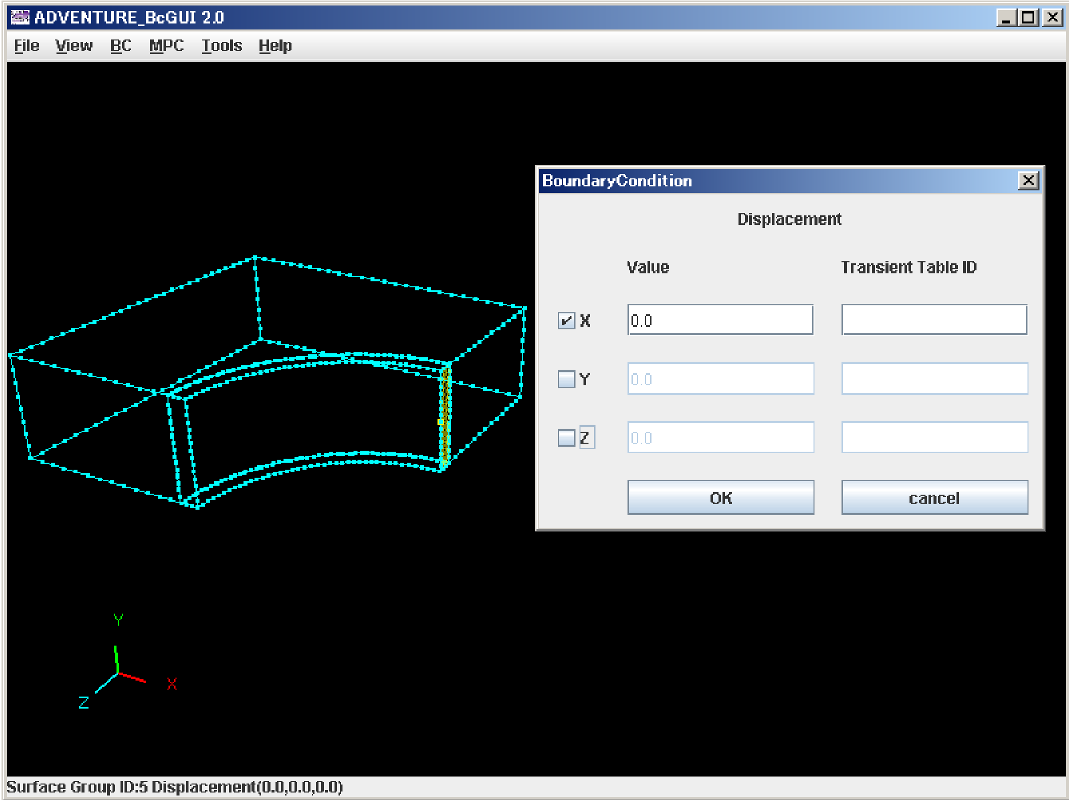


Fig. 6.4-4 Selection of the surface group #5 and setting of symmetric conditions

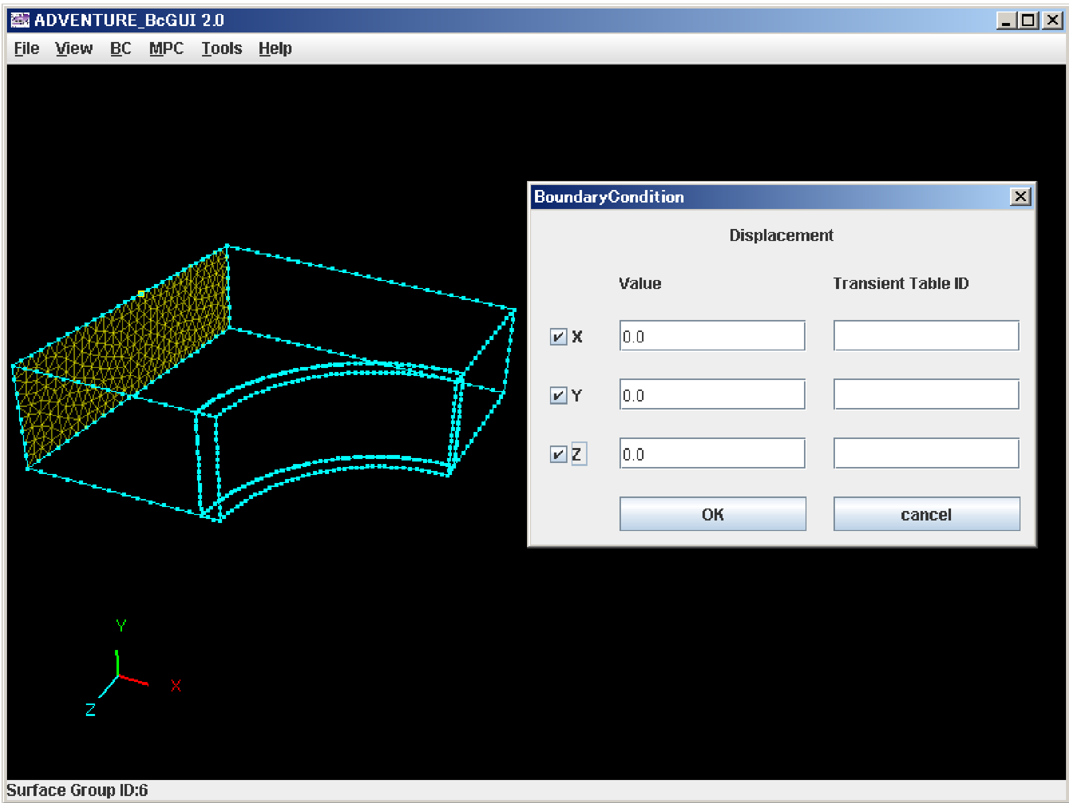


Fig. 6.4-5 Selection of the surface group #6 and setting of displacement constraints

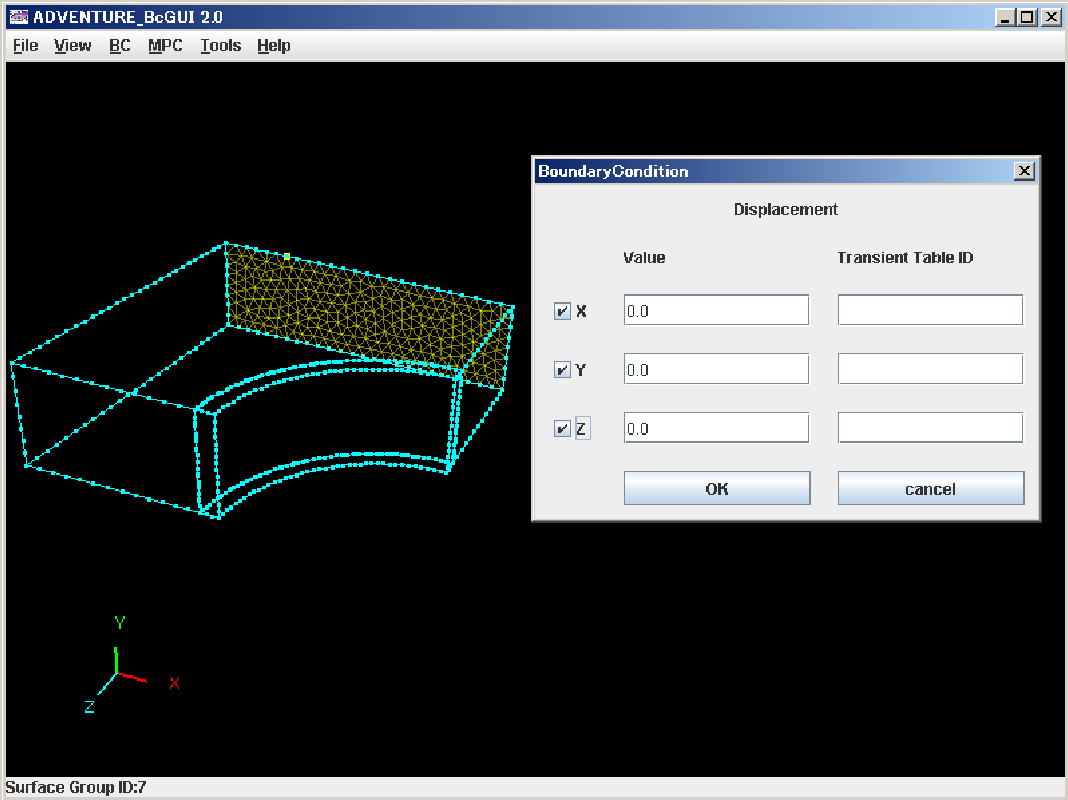


Fig. 6.4-6 Selection of the surface group #7 and setting of displacement constraints

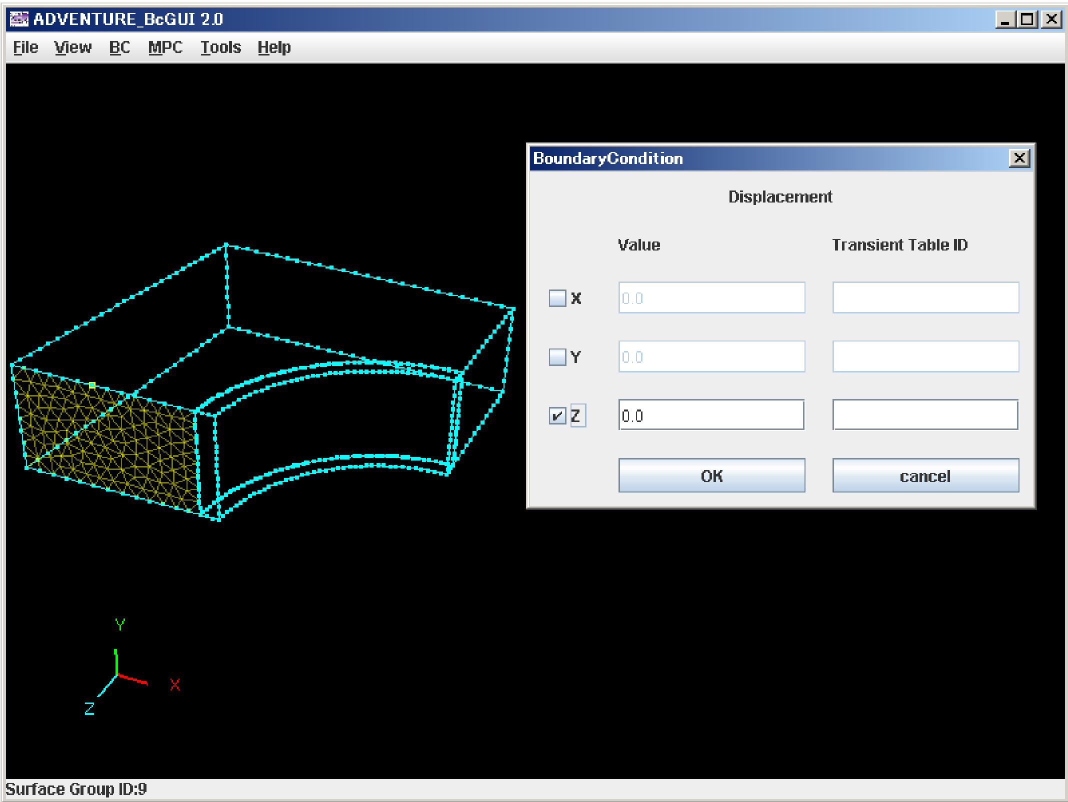


Fig. 6.4-7 Selection of the surface group #9 and setting of symmetric conditions

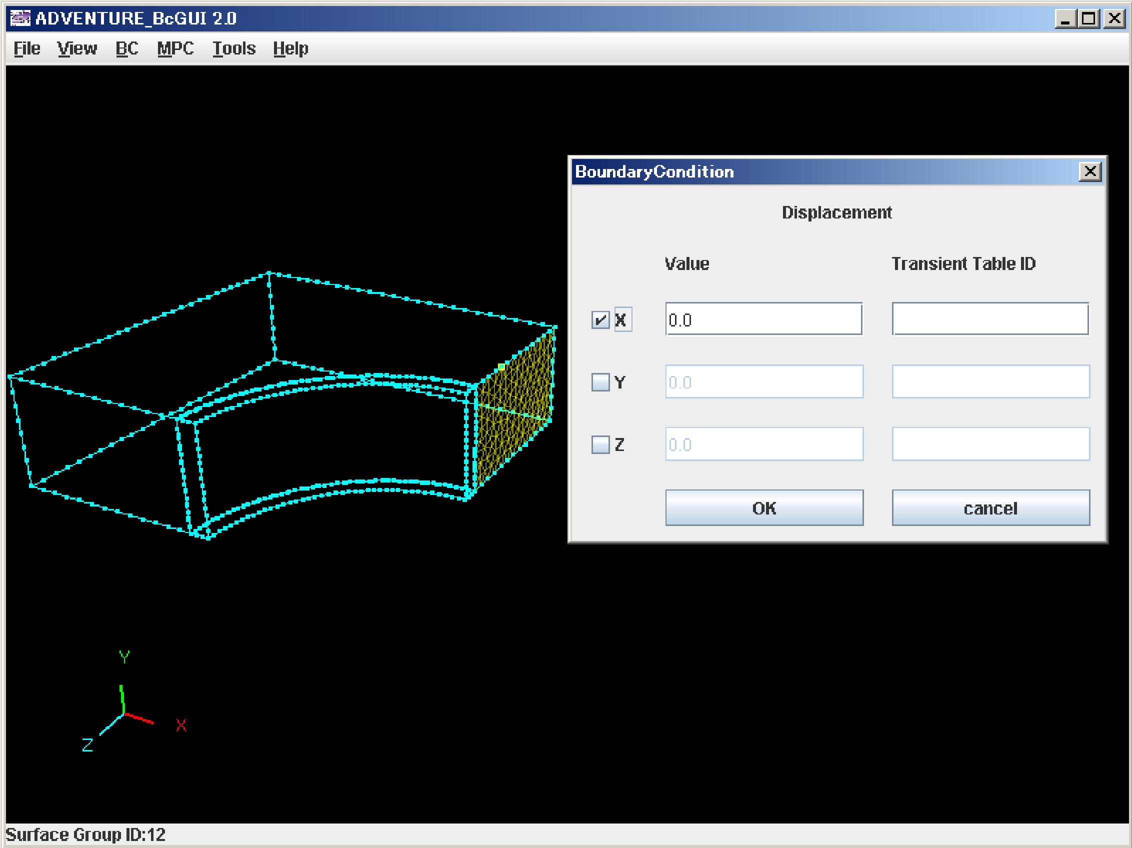


Fig. 6.4-8 Selection of the surface group #12 and setting of symmetric conditions

6.5 Tassem

This is an example of a T-shaped wood and two reinforcing members in contact with the wood.

Commands and a brief description will be given below.

Combining mesh data files

```
% MPC_mshmrg.pl Tpole.msh Rstrengthen.msh Lstrengthen.msh
```

The configuration of the output MPC_merged.msh is shown in Fig. 6.5-1. The T-shape in Fig. 6.5-1 comes from the Tpole.msh, and the front side stiffener comes from the Rstrengthen.msh, and the stiffener in the back comes from the Lstrengthen.msh.

Extraction of the surface of the mesh

```
% msh2pch MPC_merged.msh 3
```

Creation of MPCs

Total of 4 pairs of surface groups described in the combi.cmb consist of 7 and 11, 5 and 29, 8 and 17, and 3 and 21.

```
% MPC_assem2 MPC_merged.msh MPC_merged_3.fgr combi.cmb > MPC_merged_3.mpc
```

Write "LinearConstraint number of lines" at the beginning of the MPC_merged_3.mpc after examining its number of lines.

Creation of an adv file by the a2adv

```
% a2adv.pl MPC_merged_3.mpc LinearConstraint.adv
```

Settings of boundary conditions (loads and constraints)

```
% BcGUI MPC_merged_3.pch MPC_merged_3.pcg -ocnd MPC_merged_3.cnd
```

Boundary conditions will be set as shown in Fig. 6.5-2 and Fig. 6.5-3.

Setting of material properties

MPC_merged_3.mat will be created with the Young's modulus equal to 21000.0 and the Poisson's ration equal to 0.3.

Creation of an adv file by the makefem3

```
% makefem3 MPC_merged.msh MPC_merged_3.fgr MPC_merged_3.cnd MPC_merged_3.mat
MPC_merged_3.adv -t MPC_merged_3.trn
```

Combining adv files

```
% advcat MPC_merged_3.adv LinearConstraint.adv Tassem.adv
```

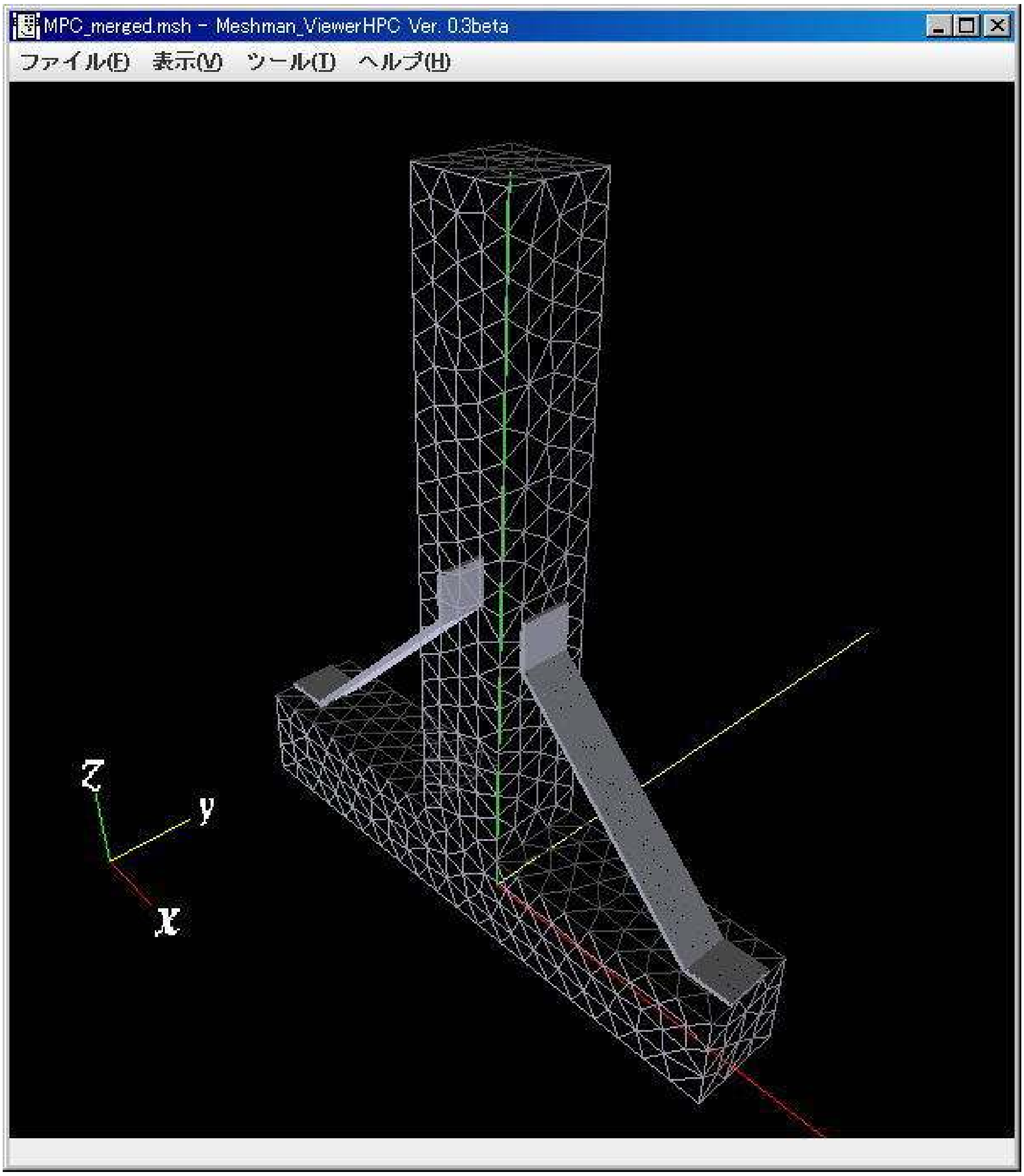


Fig. 6.5-1 The configuration of the MPC_merged.msh for Tassem

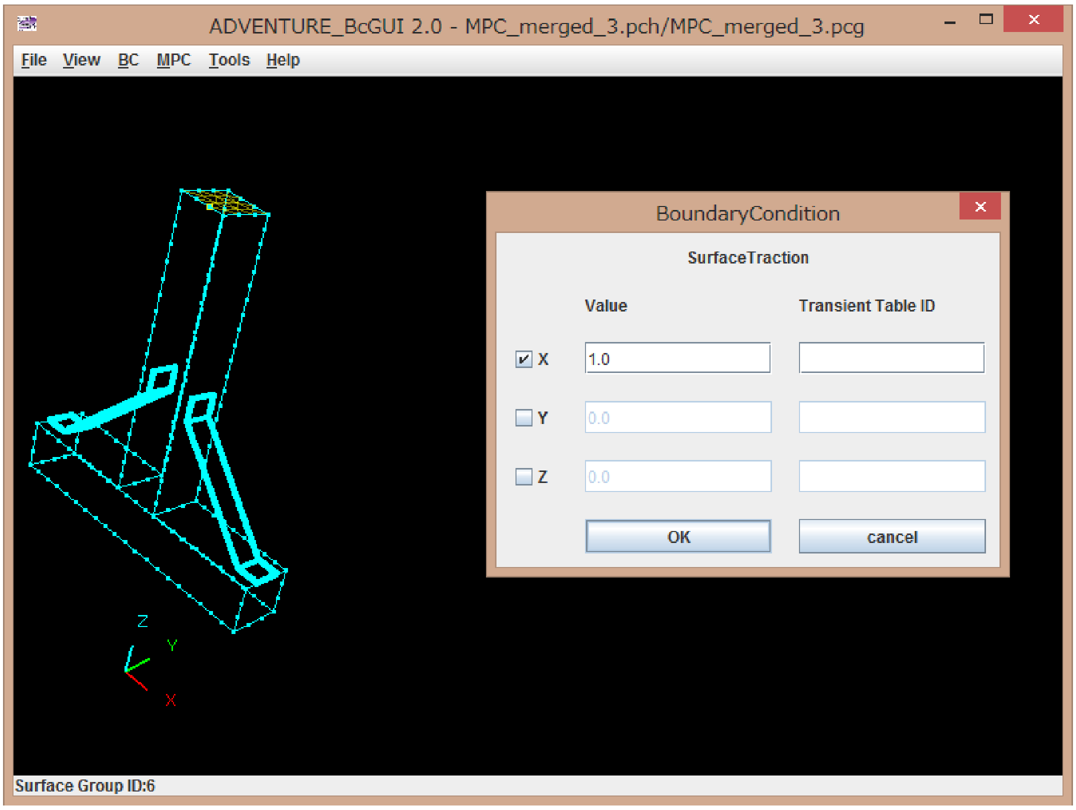


Fig. 6.5-2 Selection of the surface group #6 and setting of load conditions

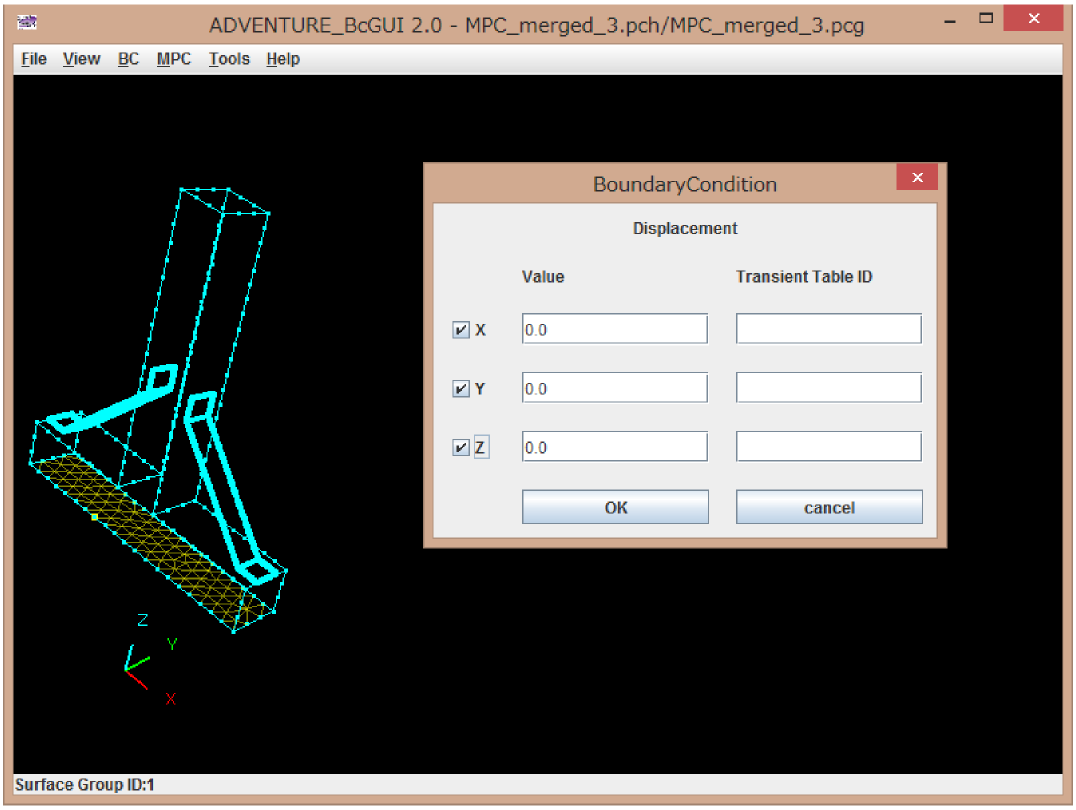


Fig. 6.5-3 Selection of the surface group #1 and setting of displacement constraints

Part IV The Manual for the a2adv

1. Summary

The a2adv.pl is an analysis model file converter from an ASCII file to the format adapted to the ADVENTURE IO.

2. Functions

It has the ability to convert an analysis model of text file format to an ADVENTURE IO compatible format.

3. The Operating Environment

OS	UNIX, Linux
Compiler	C
Library	ADVENTURE_IO
Executable	Perl

4. How to Install

4.1. Compilation

Edit the second line in src/Makefile appropriately,

```
ADVIO_DIR   = $(HOME)/ADVENTURE
```

then type the following command to compile.

```
% make addadv
```

4.2. How to Install

Copy the addadv and the a2adv.pl to an appropriate directory (place all of them in the same directory).

5. How to Use

5.1. How to Execute the Command

```
% a2adv.pl [option]  <input_file> [ <input_file> ... ] <output_file>
```

<input_file>: The names of input files of ASCII format. Multiple files can be designated. In the case of "-", input is read from the standard input. However, reading 'fem' files and 'msh' files from the standard input is impossible. This command supports only the a2adv format.

<output_file>: An ADVENTURE IO format file name to be output.

Option

-add: Instead of creating a new output file, data will be appended to an existing file, if any.

5.2. Input Files Format

・a2adv Input file format

Standard input files are written in the a2adv input file format. For the a2adv input file's format, description is given in Section A.1 in the appendix, using the cube.dat as an example.

・Other file formats

Besides the a2adv input file format, the each of the following files can be input.

(a) An integrated analysis model file (fem file)

(b) A mesh data file (msh file)

The msh format of the item (b) is that obtained by removing the boundary conditions from (a).

File (a) or (b) may or may not be used. But if used, only one of them can be used, and be sure to specify it as the first command line argument.

However, in the case of using (a), boundary conditions and material properties are not converted, and only the mesh is converted as in (b).

5.3. The Mode for Appending to an Output file

The default behavior is to create a new output file. Therefore, an exiting file of the same name will be overwritten.

If "-add" option is used, a boundary condition file and a mesh file (including the contents of "Element" and "Node") will be appended to the existing file.

6. Execution Examples

Various execution commands by means of the a2adv.pl will be introduced. File names to be used here are of the following file types.

Input files (located in "samples" directory)

cube.dat: The a2adv input file (mesh and analysis conditions are mixed)

cube.fem: The fem format integrated analysis model file cube.msh: A mesh data file

TypeV.mpc: An MPC description file

conditions1.dat: A boundary condition file (material properties only)

conditions2.dat: A boundary condition file (boundary conditions and material properties)

Output files

cube1.adv through cube5.adv, TypeV.adv: ADVENTURE Format files

Some of the example commands are illustrated below.

An a2adv input file format analysis model (mesh and boundary conditions are integrated) will be converted.

```
% a2adv.pl  cube.dat  cube1.adv
```

An analysis model file and a boundary condition file will be converted.

```
% a2adv.pl  cube.fem  conditions1.dat cube2.adv
```

A mesh data file and a boundary condition file will be converted.

```
% a2adv.pl  cube.msh  conditions2.dat cube3.adv
```

An MPC description file will be converted.

```
% a2adv.pl  TypeV.mpc TypeV.adv
```

After only a mesh is converted, analysis conditions will be added.

```
% a2adv.pl  cube.msh  cube4.adv                (<- cube4.adv will be newly created)
% a2adv.pl -add conditions2.dat  cube4.adv     (<- added to cube4.adv)
```

After only an analysis model is converted, analysis conditions will be added.

```
% a2adv.pl  cube.fem  cube5.adv                (<- cube5.adv will be newly created)
% a2adv.pl -add conditions1.dat  cube5.adv     (<- added to cube5.adv)
```

7. Examples

7.1. cube.dat

A mesh of linear hexahedral elements, and shape is cube (Fig. 7.1-1). The total number of elements is 64 and the total number of nodes is 125. The number of forced displacement boundary condition data is 75, and the plane of z = 0 (bottom plane) is fully constrained. The number of nodal load boundary condition data is 25 and a load with an intensity of -0.016 is applied on the plane of z = 10 (top plane) in the z direction.

21,000. is specified as the Young's modulus and 0.4 is specified as the Poisson's ratio. The range of coordinate values is (0, 0, 0) to (10, 10, 10).

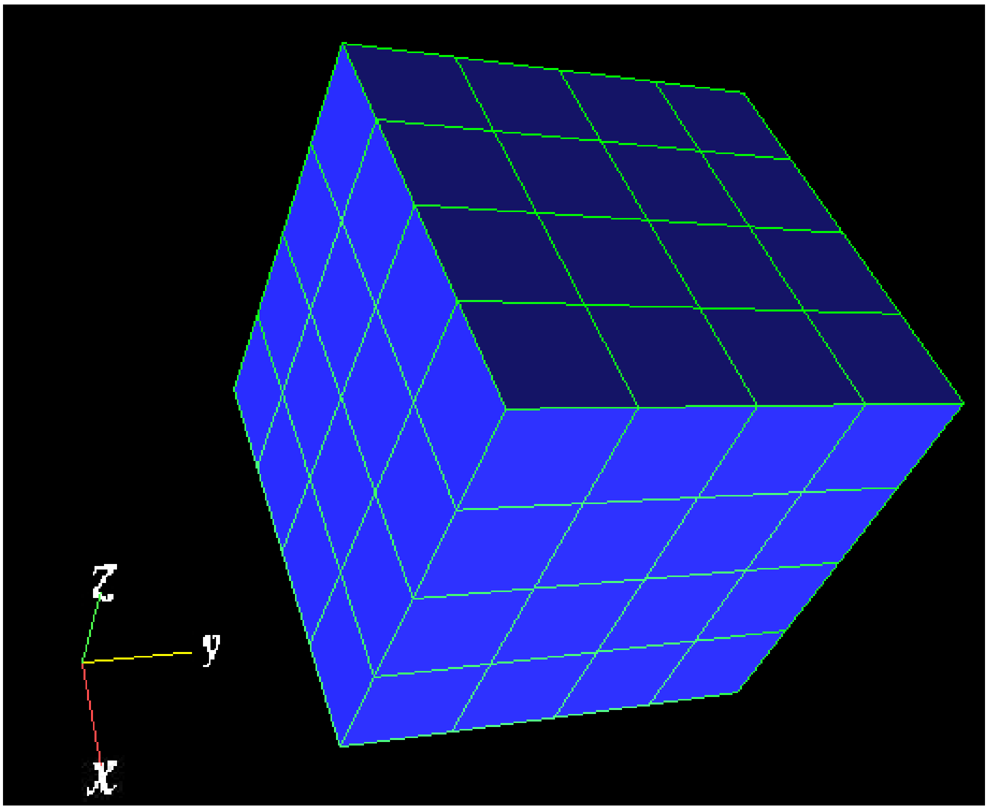


Fig. 7.1-1 The cube written in cube.dat, cube.fem, and cube.msh

7.2. cube.fem

A mesh of linear hexahedral elements, and shape is cube (Fig. 7.1-1). The total number of elements is 64 and the total number of nodes is 125. The number of forced displacement boundary condition data is 75, and the plane of z = 0 (bottom plane) is fully constrained. The number of nodal load boundary condition data is 25 and a load with an intensity of -0.016 is applied on the plane of z = 10 (top plane) in the z direction. The range of coordinate values is (0, 0, 0) to (10, 10, 10).

7.3. cube.msh

A mesh of linear hexahedral elements, and shape is cube (Fig. 7.1-1).

The total number of elements is 64 and the total number of nodes is 125. The range of coordinate values is (0, 0, 0) to (10, 10, 10).

7.4. TypeV.mpc

A general MPC condition is specified. The MPCRHS is the value of the right-hand side of each condition.

7.5. conditions1.dat

The Young's modulus is 21,000, and the Poisson's ratio is 0.4.

7.6. conditions2.dat

The number of forced displacement boundary condition data is 75, and the plane of z = 0 (bottom plane) is fully constrained. The number of nodal load boundary condition data is 25 and a load with an intensity of -0.016 is applied on the plane of z = 10 (top plane) in the z direction. The Young's modulus is defined as 21,000, and the Poisson's ratio is defined as 0.4.

7.7. conditions-temp.dat

This is the condition data of thermal stress analysis. Thermal expansion rate is 4.2941xE-6, and the reference temperature is 300.0. Temperature is given at each node.

7.8. multi-material.dat

The number of forced displacement boundary condition data is 75, and the plane of z = 0 (bottom plane) is fully constrained. The number of nodal load boundary condition data is 25 and a load with an intensity of -0.016 is applied on the plane of z = 10 (top plane) in the z direction.

The Young's modulus of the material ID 0 is defined as 21,000, and its Poisson's ratio is defined as 0.4. The Young's modulus of the material ID 1 is defined as 71,600, and its Poisson's ratio is defined as 0.345.

The material ID of each element is given.

7.9. unknownlabel.dat

A mesh of linear hexahedral elements, and shape is cube.

The total number of elements is 1 and the total number of nodes is 8. The range of coordinate values is (0, 0, 0) to (1, 1, 1).

This is an example in which a FEGA is defined in the data file because it is not defined in the program.

```
#################################################################
UnknownFEGA_A   1       # FEGA not defined in a2adv
#################################################################
```

```
fega_type=AllElementConstant
format=f8

  3.0

################################################################
UnknownFEGA_B   2       # FEGA not defined in a2adv
################################################################
fega_type=NodeVariable
format=f8

  3  9.0
 10  0.0
```

As is described in Section A.4., the fega_type and the format are defined in the data file. "UnknownFEGA_A" is a label, and the number of data is one. The fega_type specifies the type of the FEGA, and the format specifies sequence of numbers and the type. UnknownFEGA_B is similar.

7.10. unknownlabel.msh

A mesh of linear hexahedral elements, and shape is cube.

The total number of elements is 1 and the total number of nodes is 8. The range of coordinate values is (0, 0, 0) to (1, 1, 1).

Part V The Manual for the MpcLocal2Global

1. What is the MpcLocal2Global?

The MpcLocal2Global is a program for converting MPC condition data files with regard to nodes or surface groups based on surface patch data structure to those based on tetrahedral mesh data structure. Its data flow is shown in Fig.5.1-3 in Part I.

2. The Operating Environment and Compilation

For compilation, C compiler (and Make) is required.

Operation verification was done with the following environments.

- Linux(Kernel 2.6.9) + GCC 3.4.6
- Microsoft Windows2000 + Cygwin 1.5.18 + GCC 3.4.4
- Mac OS X 10.3.9 + GCC 3.3

Because there is no environment-specific programming in particular, most C compiler will do.

In order to compile, after unpacking the archive file, move to the MpcLocal2Global/src directory, and type the following command.

```
% make
```

If compilation is successful, an executable file called MpcLocal2Global (MPCLocal2Global.exe if on Windows) will be created.

For installation, continue to execute the following command in the same directory.

```
% make install
```

The executable file MpcLocal2Global (MPCLocal2Global.exe if on Windows) will be copied to ~/ADVENTURE/bin.

To install in a different directory, edit the src/Makefile and rewrite the following line.

```
ADV_HOME=${HOME}/ADVENTURE
```

To delete compilation results, execute the following.

```
% make clean
```

3. How to Execute

It is assumed that the PATH includes the directory that contains the executable file.

In the case of handling only primary nodes:

```
% MPCLocal2Global pchFile trnFile pcgFile cndFile
```

In the case of handling also quadratic nodes:

```
% MPCLocal2Global -s pchFile trnFile pcgFile cndFile mshFile fgrFile
```

In the case of showing help:

```
% MPCLocal2Global -h
```

In the case of showing the version:

```
% MPCLocal2Global --version
```

Parameters

pchFile	(Input) :extracted surface mesh data file name
trnFile	(Input) :global [nodal] index file name
pcgFile	(Input) :A surface patch group data file name
cndFile	(Input) :MPC conditions file name
mshFile	(Input) :mesh data file name
fgrFile	(Input) :mesh surface data file name

Options

-s	:the option in the case of handling quadratic nodes.
-h	:the option to show the usage of this program.
--version	:the option to show the version of this program.

The following is a detailed description of each file.

cndFile (Input)

This file has a format for describing analysis conditions extended to MPC conditions. It is capable of describing 7 types of MPC conditions. Of course, mixture of types is also possible. For details, refer to the Appendix Section A.1.10.

pchFile (Input)

This is an extracted surface mesh data file. This software handles only the surface of tetrahedral mesh. For details, refer to the Appendix Section A.1.4.

trnFile (Input)

This file is a data file that shows mapping from the node numbers of a pchFile to the node numbers of a mesh data file (extension msh). No mesh data file is used in this software, but it is assumed that a mesh data file exists. For details, refer to the Appendix Section A.1.6.

pcgFile (Input)

This is a surface patch group data file. For details, refer to the Appendix Section A.1.5.

fgrFile (Input)

This is a mesh surface data file です。For details, refer to the Appendix Section A.1.3.

mshFile (Input)

This is a mesh data file. For details, refer to the Appendix Section A.1.2.

mpcFile (Output)

This is a data file that describes MPC conditions by means of node numbers of a mesh data file. This file needs not to be designated as a parameter. The use as an input data file a2adv command. If used as an input data file to the a2adv command, this file can be converted into a data file of the ADVENTURE_IO format. For details, refer to the Appendix Section A.1.11.

4. Execution Examples

4.1 When Adding MPC Conditions to Primary Nodes Only

For example, the following command

```
 % MPCLocal2Global doubleNut.pch doubleNut.trn doubleNut.pcg doubleNutMpc.cnd
doubleNut.msh
```

will generate the following new file.

./doubleNutMpc.mpc

4.2 When Adding MPC Conditions to Quadratic Nodes Too

For example, the following comman

```
 % MPCLocal2Global -s doubleNut.pch doubleNut.trn doubleNut.pcg doubleNutMpc.cnd
doubleNut.msh doubleNut.fgr
```

will generate the following new file.

./doubleNutMpc.mpc

5. Examples

5.1 doubleNut.files

Input files

doubleNut.msh

Solid_2.0.pch

Solid_2.0.trn

Solid_2.0.pcg

MPC condition files

MpcTest.cnd:	Rigid beam V
MpcTest3.cnd:	Rigid beam V
MpcTest7.cnd:	Rigid beam V

Output files

MpcTest3.mpc:	Output from MptTest3.cnd
MpcTest4.mpc:	File manually modified from the MptTest3.mpc
MpcTest6.mpc:	File manually modified from the MptTest4.mpc
MpcTest7.mpc:	Output from MptTest7.cnd

Files not directly used by the program but required for solver execution

doubleNutOutQuadratic.msh:	3,330 elements and 6,025 nodes
Solid_2.0.fgr:	A mesh surface data file
Solid_2.0.cnd:	Normal boundary conditions
Solid_mp_multi.dat:	A material properties data file
Solid_2.0.adv:	An Integrated input file(except MPC conditions)

MPC condition files created in the BcGUI test

MpcI_N.cnd:	Rigid Beam I between nodes
MpcI_SG.cnd:	Rigid Beam I between surface groups
MpcII_N.cnd:	Rigid Beam II between nodes
MpcII_SG.cnd:	Rigid Beam II between surface groups
MpcIII_N.cnd:	Rigid Beam III between nodes
MpcIII_SG.cnd:	Rigid Beam III between surface groups
MpcIV_N.cnd:	Rigid Beam IV between nodes
MpcIV_SG.cnd:	Rigid Beam IV between surface groups
MpcV_N.cnd:	Rigid Beam V between nodes
MpcLM_N.cnd:	Arbitrary MPC between nodes

5.2 doubleBeam.files

Input files

Solid_2.0.pch

Solid_2.0.trn

Solid_2.0.pcg

An MPC condition file

MpcI_N.cnd: Rigid Beam I between nodes

Files not directly used by the program but required for solver execution

Solid.msh:	156 elements and 426 nodes
Solid_2.0.fgr:	A mesh surface data file
Solid_2.0.cnd:	Normal boundary conditions
Solid_mp_multi.dat:	A material properties data file
Solid_2.0.adv:	An Integrated input files(except MPC conditions)

5.3 Test1

Input files

testCase1.pch: If not $(x_{1,} y_1) \approx (x_{2,} y_2)$

testCase2.pch: If $(x_{1,} y_1) \approx (x_{2,} y_2)$ and $y_2 \approx y_3$

testCase3.pch: If $(x_{1,} y_1) \approx (x_{2,} y_2)$ and if not $y_2 \approx y_3$

testCase.trn

testCase_typeV.pcg

An MPC condition file

MpcSimplest.cnd

5.4 Test2

Input files

doubleNut_2.0.pch

doubleNut_2.0.trn

doubleNut_2.0.pcg

An MPC condition file

MpcTest7.cnd

5.5 doubleNut2010

Input files

Solid_2.0.pch

Solid_2.0.trn

Solid_2.0.pcg

Solid_2.0.msh

Solid_2.0.fgr

MPC condition files

Under doubleNut2010/cndFile,

TypeI.cnd:	Rigid Beam I
TypeII.cnd:	Rigid Beam II
TypeIII.cnd:	Rigid Beam III
TypeIV.cnd:	Rigid Beam IV
TypeV.cnd:	Rigid Beam V
SimpleBeam.cnd:	Simple Beam
LinearMPC.cnd:	Arbitrary MPC
Mix.cnd:	Mixture of Rigid Beam I to V, Simple Beam, arbitrary MPC

Output files

Under doubleNut2010/mpcFile,

TypeI.mpc:	Output from TypeI.cnd
TypeII.mpc:	Output from TypeII.cnd
TypeIII.mpc:	Output from TypeIII.cnd
TypeIV.mpc:	Output from TypeIV.cnd
TypeV.mpc:	Output from TypeV.cnd
SimpleBeam.mpc:	Output from SimpleBeam.cnd
LinearMPC.mpc:	Output from LinearMPC.cnd
Mix.mpc:	Output from Mix.cnd

Files not directly used by the program but required for solver execution

Solid_2.0.adv

6. Summary of MPC Conditions

6.1 Rigid Beams

Five types of Rigid Beams are defined.

Rigid Beam I: The relative displacement between two nodes is totally fixed.

Rigid Beam II: For displacing the two nodes that are initially separated so that they become in contact, constraints are applied to the relative displacement vector between two nodes in the vector direction defined by the two nodes. (If the two nodes have a same position, rigid beam II cannot be defined.

Rigid Beam III: Constraints are applied so that the distance between two nodes is kept unchanged.

Rigid Beam IV:Constraints are applied to the relative displacement between two nodes only in the normal direction of any surface group of user' choice.

Rigid Bem V:Constraints are applied so that a user-specified virtual node, and existing nodes which are more than three and are not on a straight line are connected by a Rigid Beam V to make a rigid body.

(1) Rigid Beam I

Constraint equation: $\vec{u}_1 = \vec{u}_2$

Here, $\vec{u}_1$ is a displacement of a node P_1 on a volume, and $\vec{u}_2$ is a displacement of another node P_2 on the other volume. If the components of the displacements are used, the following 3 equations are obtained.

$u_1 - u_2 = 0.$

$v_1 - v_2 = 0.$

$w_1 - w_2 = 0.$

Here, u_1 and u_2 are x components of the displacements of P_1 and P_2, respectively, and v1 and v2 are y components of P_1 and P_2, respectively, and w1 and w2 are z components of P_1 and P_2, respectively.

(2) Rigid Beam II

It is assumed that the positions of two nodes do not match.

Constraint equation: $(\vec{p}_1 - \vec{p}_2) \cdot \{(\vec{p}_1 + \vec{u}_1) - (\vec{p}_2 + \vec{u}_2)\} = 0$

Here, $\vec{p}_1$ is the position vector of a node P_1 on a volume, $\vec{u}_1$ is the displacement vector of the node P_1, and $\vec{p}_2$ is the position vector of another node P_2 on the other volume, and $\vec{u}_2$ is the displacement vector of the node P_2. If the components of each vector are used, the following equation is obtained.

$$(x_1 - x_2)(x_1 - x_2 + u_1 - u_2) + (y_1 - y_2)(y_1 - y_2 + v_1 - v_2) + (z_1 - z_2)(z_1 - z_2 + w_1 - w_2) = 0$$

By expanding the expression,

$$(x_1 - x_2)u_1 - (x_1 - x_2)u_2 + (y_1 - y_2)v1 - (y_1 - y_2)v_2 + (z_1 - z_2)w_1 - (z_1 - z_2)w_2 = -(x_1 - x_2)^2 - (y_1 - y_2)^2 - (z_1 - z_2)^2$$

and reorganizing the expression,

$$(x_2 - x_1)u_1 - (x_2 - x_1)u_2 + (y_2 - y_1)v_1 - (y_2 - y_1)v_2 + (z_2 - z_1)w_1 - (z_2 - z_1)w_2 = (x_2 - x_1)^2 + (y2 - y1)^2 + (z2 - z1)^2$$

Here, u_1 and u_2 are the displacement components in the x direction, v_1 and v_2 are the displacement components in the y direction, and w_1 and w_2 are the displacement components in the z direction.

(3) Rigid Beam III

Constrains are applied so that the distance between two nodes are kept unchanged.

Constraint equation: $(\vec{p}_2 - \vec{p}_1)\cdot(\vec{u}_2 - \vec{u}_1) = 0$

Here, $\vec{p}_1$ is the position vector of a node P_1 on a volume, and $\vec{u}_1$ is the displacement vector of the node P_1, and $\vec{p}_2$ is the position vector of another node P_2 on the other volume, and $\vec{u}_2$ is the displacement vector of the node P_2. The center dot in the equation means inner product. If the components of each vector are used, the following equation is obtained.

$(x_2 - x_1)(u_2 - u_1) + (y_2 - y_1)(v_2 - v_1) +- (z_2 - z_1)(w_2 - w_1) = 0$

Here, u_1 and u_2 are the displacement components in the x direction, v_1 and v_2 are the displacement components in the y direction, and w_1 and w_2 are the displacement components in the z direction. Therefore,

$-(x_2 - x_1) u_1 + (x_2 - x_1)u_2 - (y_2 - y_1) v_1 + (y_2 - y_1) v_2 - (z_2 - z_1) w_1 + (z_2 - z_1)w_2 = 0$

So,

$(x_2 - x_1) u_1 - (x_2 - x_1) u_2 + (y_2 - y_1) v_1 - (y_2 - y_1) v_2 + (z_2 - z_1) w_1 - (z_2 - z_1) w_2 = 0$

(4) Rigid Beam IV

Constraint equation: $\vec{n}\cdot\vec{u}_1 = \vec{n}\cdot\vec{u}_2$

That is $\vec{n}\cdot(\vec{u}_2 - \vec{u}_1) = 0$

Here, $\vec{n}$ is a normal vector to an arbitrary surface group of user's choice, $\vec{u}_1$ is the displacement vector of a node P_1 on a volume, $\vec{u}_2$ is the displacement vector of another node P_2 on the other volume. The symbol center dot means inner product. If the components of each vector are used, the following equation is obtained.

$n_x (u_2 - u_1) + n_y (v_2 - v_1) + n_z (w_2 - w_1) = 0$

Therefore,

$-n_x u_1 + n_x u_2 - n_y v_1 + n_y v_2 - n_z w_1 + n_z w_2 = 0$

Here, u_1 and u_2 are the displacement components in the x direction, v_1 and v_2 are the displacement components in the y direction, and w_1 and w_2 are the displacement components in the z direction, and n_x, n_y, and n_z are x, y, and z components of the normal vector to the surface group of user's choice respectively.

The normal vector of the selected surface groups is calculated by normalizing the normal vector for each triangle in the selected surface group and taking a mean value of each component of the vectors. The normal vector of each triangle is to be defined outwardly from the body.

(5) Rigid Beam V

Constraint equation:

$$
{}_i u = \begin{bmatrix} 1 & 0 & 0 & 0 & -{}_i l_z & {}_i l_z \\ 0 & 1 & 0 & {}_i l_z & 0 & -{}_i l_z \\ 0 & 0 & 1 & -{}_i l_z & {}_i l_z & 0 \end{bmatrix} B \begin{bmatrix} {}_1 u_x \\ {}_1 u_y \\ {}_1 u_z \\ {}_2 u_x \\ {}_2 u_y \\ {}_2 u_z \end{bmatrix} \qquad (5\text{-}1)
$$

where

if $(x_1, y_1) \approx (x_2, y_2)$ and $y_2 \approx y_3$ hold, ${}_3 u_y$ will be used instead of ${}_3 u_z$, and if $(x_1, y_1) \approx (x_2, y_2)$ holds and $y_2 \approx y_3$ does not hold, ${}_3 u_x$ will be used instead of ${}_3 u_z$.

Here,

${}_i u$ is the displacement vector at a general node (a node other than virtual nodes) i, and when $\bar{x}$ is the position vector of a virtual node, and ${}_i x$ is the position vector of a general node,

${}_i l = \bar{x} - {}_i x$,

$B = A^{-1}$,

$$
A = \begin{bmatrix} 1 & 0 & 0 & 0 & -{}_1 l_z & {}_1 l_y \\ 0 & 1 & 0 & {}_1 l_z & 0 & -{}_1 l_x \\ 0 & 0 & 1 & -{}_1 l_y & {}_1 l_x & 0 \\ 1 & 0 & 0 & 0 & -{}_2 l_z & {}_2 l_y \\ 0 & 1 & 0 & {}_2 l_z & 0 & -{}_2 l_x \\ 0 & 0 & 1 & -{}_3 l_y & {}_3 l_x & 0 \end{bmatrix}
$$

will hold,

where

if $(x_1, y_1) \approx (x_2, y_2)$ and $y_2 \approx y_3$ hold,

$$
A = \begin{bmatrix} 1 & 0 & 0 & 0 & -{}_1 l_z & {}_1 l_y \\ 0 & 1 & 0 & {}_1 l_z & 0 & -{}_1 l_x \\ 0 & 0 & 1 & -{}_1 l_y & {}_1 l_x & 0 \\ 1 & 0 & 0 & 0 & -{}_2 l_z & {}_2 l_y \\ 0 & 1 & 0 & {}_2 l_z & 0 & -{}_2 l_x \\ 0 & 1 & 0 & {}_3 l_z & 0 & -{}_2 l_x \end{bmatrix}
$$

and if $(x_1, y_1) \approx (x_2, y_2)$ holds and $y_2 \approx y_3$ does not hold,

$$A=\begin{bmatrix} 1 & 0 & 0 & 0 & -{}_1l_z & {}_1l_y \\ 0 & 1 & 0 & {}_1l_z & 0 & -{}_1l_x \\ 0 & 0 & 1 & -{}_1l_y & {}_1l_x & 0 \\ 1 & 0 & 0 & 0 & -{}_2l_z & {}_2l_y \\ 0 & 1 & 0 & {}_2l_z & 0 & -{}_2l_x \\ 1 & 0 & 0 & 0 & -{}_3l_z & {}_3l_y \end{bmatrix}$$

6.2 Simple Beam

The Simple Beam will be used when applying a forced internal force between two nodes. The constraint equations are as follows,

Constraint equations: $\vec{L}_2 = L\dfrac{\vec{p}_2-\vec{p}_1}{|\vec{p}_2-\vec{p}_1|}$

$$\vec{L}_1 = L\frac{\vec{p}_1-\vec{p}_2}{|\vec{p}_1-\vec{p}_2|}$$

Here, $\vec{p}_1$ is the position vector of a node P_1 on a volume, $\vec{L}_1$ is the load vector applied to the node P_1, $\vec{p}_2$ is the position vector of another node P_2 on the other volume, and $\vec{L}_2$ is the load vector applied to the node P_2. L is the magnitude of the load, and has a positive value in tension, and has a negative value in compression. Constraints will be expressed by means of each component as follows.

$$L_{2x}=\frac{L(x_2-x_1)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}}$$

$$L_{2y}=\frac{L(y_2-y_1)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}}$$

$$L_{2z}=\frac{L(z_2-z_1)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}}$$

$$L_{1x}=\frac{L(x_1-x_2)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}}$$

$$L_{1y}=\frac{L(y_1-y_2)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}}$$

$$L_{1z}=\frac{L(z_1-z_2)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}}$$

Here, (x_1, y_1, z_1) and (x_2, y_2, z_2) are the coordinates of the nodes P_1 and P_2 respectively. (L_{1x}, L_{1y}, L_{1z}) and (L_{2x}, L_{2y}, L_{2z}) the load vector components at the nodes P_1 and P_2 respectively.

6.3 General MPCs for Arbitrary Number of Nodes

The constraint equation is as follows.

$$\sum_{i=1}^{n} f_{xi} u_i + \sum_{i=1}^{n} f_{yi} v_i + \sum_{i=1}^{n} f_{zi} w_i = c$$

where n is the number of nodes with general MPC conditions, i is the index among n nodes, f_{xi}, f_{yi} and f_{zi} are the coefficients for x, y, and z components of the displacement of the i^{th} node respectively and c is a constant for the linear equation.

Appendix

A.1. File Formats

A.1.1. General Analysis Data File (extension a)

Several examples are available in the samples folder for user's reference.

Here, a simple explanation is given by taking the example of samples/cube.dat.

As for the unit system, create an analysis model using a consistent unit system because unit conversion is not performed neither in the a2adv.pl nor in the solvers.

```
----------------------------------------------------------
#
#    Sample analysis model data  for a2adv.pl
#

#############################################
Element 3DLinearHexahedron 64       # Element connectivity,Element type,# of
elements
#############################################

        0       1       6       5       25      26      31      30
        1       2       7       6       26      27      32      31
        2       3       8       7       27      28      33      32
        3       4       9       8       28      29      34      33
        5       6       11      10      30      31      36      35
        6       7       12      11      31      32      37      36
        7       8       13      12      32      33      38      37
                .
                .
                .
        90      91      96      95      115     116     121     120
        91      92      97      96      116     117     122     121
        92      93      98      97      117     118     123     122
        93      94      99      98      118     119     124     123

#############################################
```

```
Node 125                                  # Nodal coordinates, # of nodes
############################################
    0.00000000e+00    0.00000000e+00    0.00000000e+00
    2.50000000e+00    0.00000000e+00    0.00000000e+00
    5.00000000e+00    0.00000000e+00    0.00000000e+00
    7.50000000e+00    0.00000000e+00    0.00000000e+00
    1.00000000e+01    0.00000000e+00    0.00000000e+00
    0.00000000e+00    2.50000000e+00    0.00000000e+00
                .
                .
                .
    2.50000000e+00    1.00000000e+01    1.00000000e+01
    5.00000000e+00    1.00000000e+01    1.00000000e+01
    7.50000000e+00    1.00000000e+01    1.00000000e+01
    1.00000000e+01    1.00000000e+01    1.00000000e+01

##################################################################
ForcedDisplacement      75                # Forced displacement condition, # of data
items
##################################################################
# Node, x/y/z, displacement value
      0    0     0.000000                 # <- A first data item
      0    1     0.000000                 # <- A second data item
      0    2     0.000000
      2    0     0.000000
      2    1     0.000000
                .
                .
                .
     24    2     0.000000                 # <- A 75th data item


##################################################################
Load       25                             # Nodal load condition, # of data items
##################################################################
# Node, x/y/z, load value
```

```
    100    2      -0.025000                     # <- A first data item
    101    2      -0.050000                     # <- A second data item
                .
                .
                .
    124    2      -0.025000                     # <- A 25th data item


######################################################################
YoungModulus              # Young's modulus
######################################################################
   2.1e4

######################################################################
PoissonRatio            # Poisson's ratio
######################################################################
   0.4

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

* A character '#' means the beginning of a comment and the characters after it until the end of the line will be ignored.

* Blank lines are allowed.

* The data structure corresponds to the FEGA Document in ADVENTURE Format, and it has the following structure for a single Document. For details, refer to Chapter A.4.

```
-------------------------------------
   [content type] [label name | element name]      [# of data items]
    [ <key> = <val> ]
            .
            .
            .
    data[0][0]     data[0][1]    ....    # <-   A first data item
    data[1][0]     data[1][1]    ....    # <-   A second data item
            .
            .
```

```
        .
   data[n][0]    data[n][1]   ....    # <-   nth (= # of data items) data item
-------------------------------------
```

There are three content types that is Element (element connectivity), Node (nodal coordinates) and the FEGA. The FEGA is optional.

In the case of the FEGA, a label name must follow. In the case of Element, element type must follow.

For example, the label name of a FEGA is a name that has been determined for each type of data. For example, a label "Load" is used for nodal load boundary conditions.

The number of data items is the number of lines of data that follow. It is assumed to be 1, unless otherwise noted.

The line of [<key> = <val>] is a property (property defined in the "Document" of ADVENTURE Format) that is used to add additional information to this data.

This line is used when additional properties are necessary other than the several default properties set at the time of data conversion.

Then, actual data are listed. In each line, data items whose numbers are determined according to their label names are listed. Lines are repeated as many as the number of specified data items.

A.1.2. Mesh Data File (extension msh)

The format of a mesh data file (for a linear tetrahedral element)

```
-----------------------------------------------------------------
420                       <- The number of elements
171 116 117 99            <- Nodes that compose a 0th element
145 132 172 156           <- Nodes that compose a 1st element
145 105 132 156
145 131 105 156
130 110 131 156
... (Omitted) ...
72 63 60 74
63 61 60 74
63 62 61 74
```

```
62 32 61 74
62 33 32 74                 <- Nodes that compose a (420-1)th element
180                         <- The number of nodes
2.5 -4.33013 0              <- X, Y, and Z coordinates of a 0th node
-2.5 -4.33013 0             <- X, Y, and Z coordinates of a 1st node
-5 0 0
-2.5 4.33013 0
2.5 4.33013 0
... (Omitted) ...
0.173913 -4.33013 9.48837
0.560861 -4.33013 7.98095
3.73934 -2.18354 9.54426
4.24419 -1.30912 8.32609
3.49047 -2.61458 7.93914    <- X,Y,Z coordinates of a (180-1)th node
2                           <- Number of volumes                      ┐
210                         <- Number of elements in volume # 0       │
0                           <- 0th element in volume # 0              │
1                           <- 1st element in volume # 1              │  Only for
... (Omitted) ...                                                     │  multi-material data
209                         <- (210-1)th element in volume # 0        │
210                         <- Number of elements in volume # 1       │
210                         <- 0th element in volume # 1              │
... (Omitted) ...                                                     │
419                         <- (210-1)th element in volume # 1        ┘
-----------------------------------------------------------------
```

*The number of nodes that compose an element is as follows.

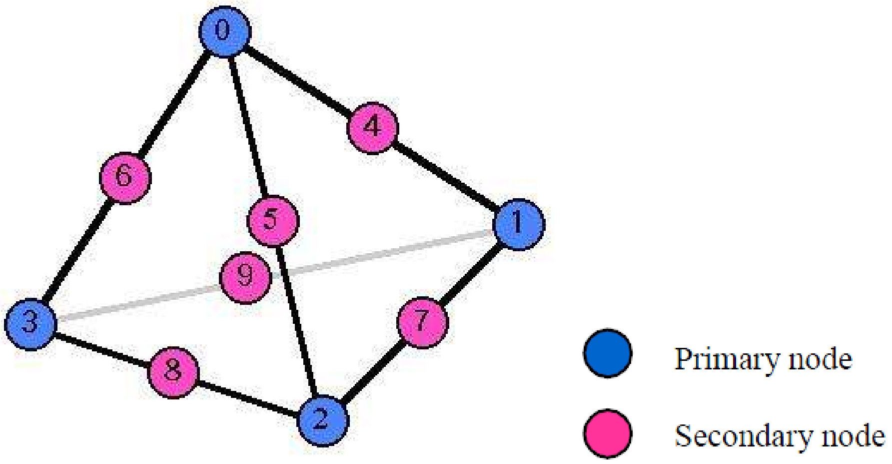
For linear tetrahedron, 4 nodes are given.

For quadratic tetrahedron, 10 nodes are given.

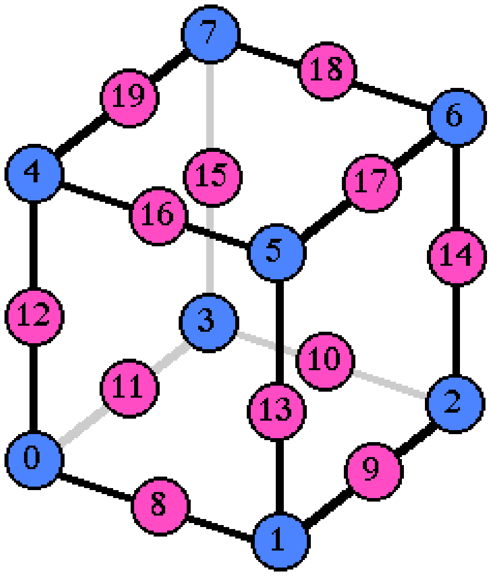
For linear hexahedron, 8 nodes are given.

For quadratic hexahedron, 20 nodes are given.

The following figures show the node sequence in the connectivity of an element.



Tetrahedral element



Hexahedral element

A.1.3. A Mesh Surface Data File (extension fgr)

The format of a mesh surface data file (for a linear tetrahedral element)

```
-----------------------------------------------------------------
8                  <- Type of element*
12                 <- # of surface groups
4                  <- # of surfaces that compose a 0th surface group
0 0 0 3 4 1        <- Info on 0th surface of 0th surface group**
```

```
2 0 3 6 7 4        <- Info on 1st surface of 0th surface group
1 0 1 4 5 2        <- Info on 2nd surface of 0th surface group
3 0 4 7 8 5        <- Info on 3rd surface of 0th surface group
4                  <- # of surfaces that compose a 1st surface group
0 2 0 1 10 9       <- Info on 0th surface of 1st surface group
1 2 1 2 11 10      <- Info on 1st surface of 1st surface group
4 2 9 10 19 18     <- Info on 2nd surface of 1st surface group
5 2 10 11 20 19    <- Info on 3rd surface of 1st surface group
... (Omitted) ...
4                  <- # of surfaces that compose a (12-1)th surface group
12 1 45 46 49 48   <- Info on 0th surface of (12-1)th surface group
13 1 46 47 50 49   <- Info on 1st surface of (12-1)th surface group
14 1 48 49 52 51   <- Info on 2nd surface of (12-1)th surface group
15 1 49 50 53 52   <- Info on 3rd surface of (12-1)th surface group
-----------------------------------------------------------------
```

*Definition of a type of element:

Linear tetrahedron	4
Quadratic tetrahedron	10
Linear hexahedron	8
Quadratic hexahedron	20

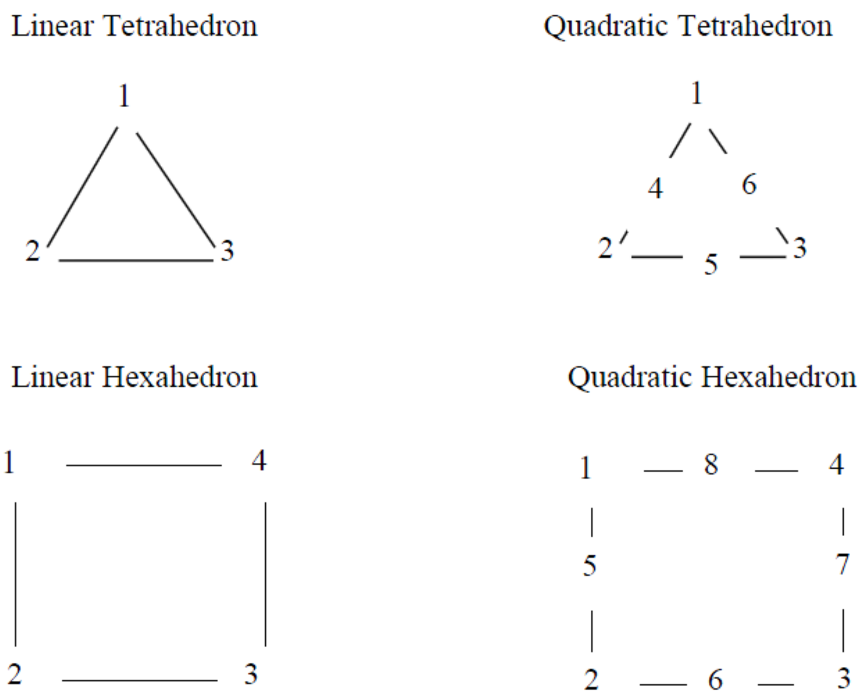
*Information on a surface

Here, "a surface" means a triangle or a quadrilateral that consists of nodes. The first number represents the ID of the element the surface belongs to, and the next number refers to the surface ID in the element, and the rest are the node numbers that consist the surface.

The number of nodes that make up the surface is as follows:

Linear tetrahedron	3
Quadratic tetrahedron	6
Linear hexahedron	4
Quadratic hexahedron	8

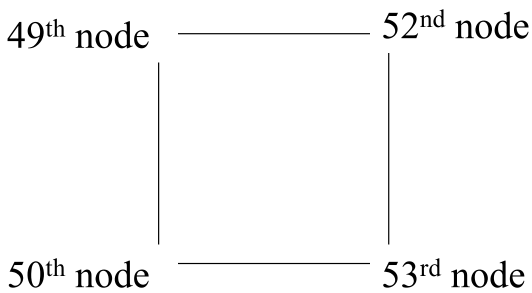
The order of the nodes in a surface is shown for each element type in the following figures.



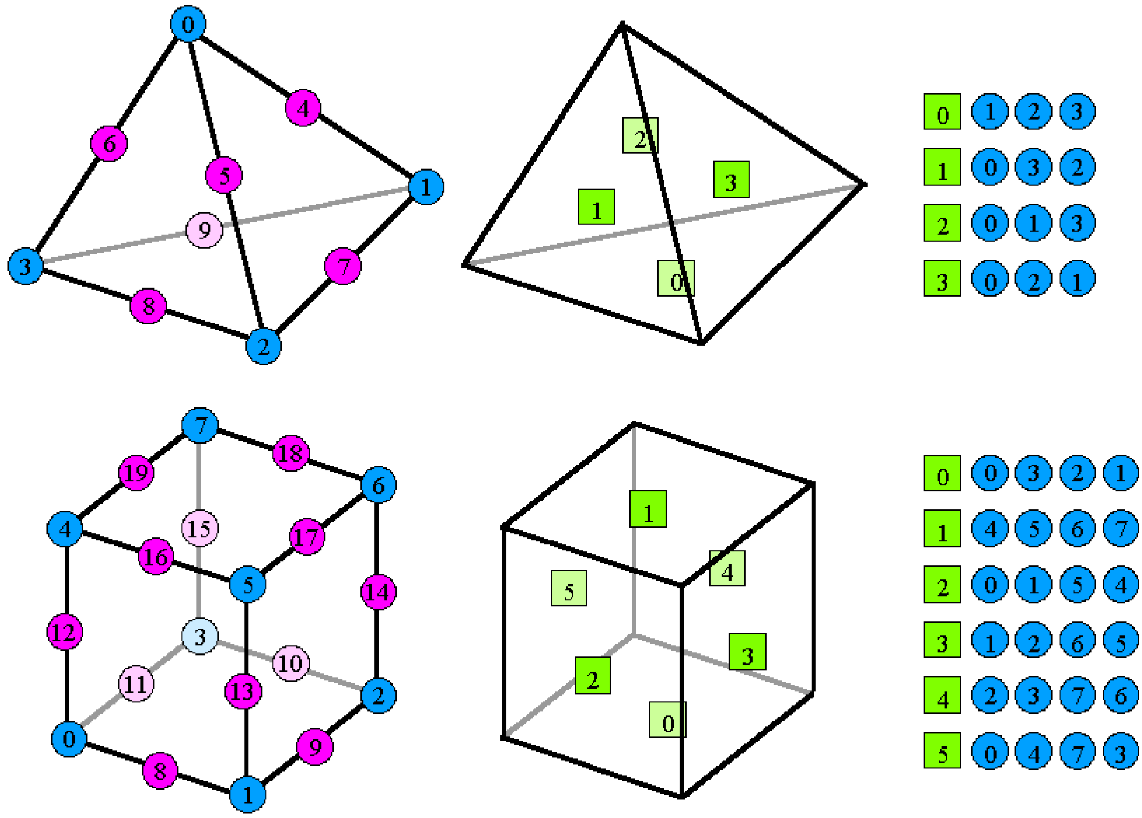
The last line of the preceding file format description is as follows.

```
15 1 49 50 53 52
```

And it refers to a quadrilateral shown below, and it belongs to the 15th hexahedral element and its surface ID in the element is 1.



Numbering of surfaces in an element is defined as follows.



A.1.4. Extracted Surface Mesh Data File (extension pch)

The format of an extracted surface mesh data file (for a linear tetrahedral element)

```
-----------------------------------------------------------------
180                            <- # of nodes
2.5   -4.33013 0               <- X, Y, and Z coordinates of 0th node
-2.5   -4.33013 0              <- X, Y, and Z coordinates of 1st node
-5   0 0
-2.5   4.33013 0
2.5   4.33013 0
... (Omitted) ...
0.173913   -4.33013 9.48837
0.560861   -4.33013 7.98095
3.73934   -2.18354 9.54426
4.24419   -1.30912 8.32609
3.49047   -2.61458 7.93914     <- X, Y, and Z coordinates of (180-1)th node
360                            <- # of patches
99 117 116                     <- Nodes that make up the 0th patch*
```

```
92 117 99                   <- Nodes that make up the 1st patch
116 100 99
92 99 118
116 91 100
... (Omitted) ...
62 71 11
63 71 62
63 70 71
63 18 70
18 63 23                    <- Nodes that make up the (360-1)th patch
-----------------------------------------------------------------
```

*A patch consists of mesh surface primary nodes.

Linear tetrahedral element:	3 nodes
Quadratic tetrahedral element:	3 nodes
Linear hexahedral element:	4 nodes
Quadratic hexahedral element:	4 nodes

Node IDs are assigned only to primary nodes on the mesh surface. Connectivity of the patches is clockwise as viewed from the outside of the shape.

A.1.5. A Surface Patch Group Data File (extension pcg)

The format of a surface patch group data file

```
-----------------------------------------------------------------
#mainVertexInfo
mainVertexN 144             <- # of main nodes*
0                           <- 0th main node
1                           <- 1st main node
2
3
4
... (Omitted) ...
157
158
159
160
```

```
161                        <- (144-1)th main node

#edgeGroupInfo
edgeGroupN 168             <- # of edge groups
edgeGroup 2                <- # of nodes that compose 0th edge group
0                          <- 0th node in 0th edge group
24                         <- 1st node in 0th edge group
edgeGroup 2                <- # of nodes that compose 1st edge group
0                          <- 0th node in 1st edge group
35                         <- 1st node in 1st edge group
... (Omitted) ...
edgeGroup 2                <- # of nodes that compose (168-1)th edge group
99                         <- 0th node in (168-1)th edge group
157                        <- 1st node in (168-1)th edge group
#faceGroupInfo
faceGroupN 28              <- # of face groups
faceGroup 24               <- # of patches that compose 0th face group
0                          <- ID of 0th patch in 0th face group
1                          <- ID of 1st patch in 0th face group
... (Omitted) ...
22                         <- ID of (24-2)th patch in 0th face group
23                         <- ID of (24-1)th patch in 0th face group
... (Omitted) ...
faceGroup 6                <- # of patches that compose (28-1)th face group
354                        <- ID of 0th patch in (28-1)th face group
355                        <- ID of 1st patch in (28-1)th face group
356                        <- ID of 2nd patch in (28-1)th face group
357                        <- ID of 3rd patch in (28-1)th face group
358                        <- ID of 4th patch in (28-1)th face group
359                        <- ID of 5th patch in (28-1)th face group
-----------------------------------------------------------------
```

*A main node is a representative node, which characterizes the shape of a model. In ADVENTURE_BCtool, primary nodes on the boundary of the surface group are selected as main nodes.

A.1.6. A Global Index File (extension trn)

The format of a global index file

```
------------------------------------------------------------
0
1
2
3
4
... (Omitted) ...
177
178
179
------------------------------------------------------------
```

The # of lines is the same as that of primary nodes on the overall mesh surface.

*A lookup table where a node ID in an extracted surface mesh data file (extension pch) can be mapped to a node ID in the corresponding mesh data file (extension msh).

* The number described in the N^{th} row is the node ID in the mesh data file (extension msh) corresponding to the N^{th} node in the extracted surface mesh data file (extension pch).

A.1.7. A Material Properties Data File (extension dat)

Eleven types of material properties shown in table A.1.7-1 can be specified.

Table A.1.7-1 List of material properties that can be specified

Name of material properties	Name of labels	Remarks
Young's modulus	YoungModulus	
Poisson's ratio	PoissonRatio	
Work hardening parameter	HardeningParameter	For elastic-plastic analysis
Initial yield stress	YieldStress	For elastic-plastic analysis
Mass density	Density	For self weight analysis
Linear expansion coefficient	ThermalExpansionCoefficient	For thermal stress analysis
Reference temperature	ReferenceTemperature	For thermal stress analysis
Thermal conductivity	HeatConductivity	For heat conduction analysis
Specific heat	SpecificHeat	For transient heat conduction analysis
Stefan-Boltzmann constant	StefanBoltzmanConstant	For thermal radiation analysis
Internal heat generation	InternalHeatGeneration	For thermal analysis

The format of a material properties data file (single material model)

```
-----------------------------------------------------------------
YoungModulus 21000.0        <- Young's modulus
PoissonRatio 0.4            <- Poisson's ratio
HardeningParameter 1000.0   <- Work hardening parameter
YieldStress 500.0           <- Initial yield stress
Density 760.0               <- Mass density
-----------------------------------------------------------------
```

The format of a material properties data file (multi-material model)

```
-----------------------------------------------------------------
#materialInfo
materialN 2                 <- # of materials
propertyN 2                 <- # of properties to be defined
YoungModulus 21000.0        } 2 properties defined  }
PoissonRatio 0.4            }                       }
YoungModulus 205940.0                               } 2 materials defined
PoissonRatio 0.3                                    }
#volumeInfo
volumeN 3                   <- # of volumes
1                           <- Material ID of 0th volume
0                           <- Material ID of 1st volume
1                           <- Material ID of 2nd volume
-----------------------------------------------------------------
```

*The BCtool automatically determines if a data file contains data for a single material model or for a multiple material model.

*In the format of multi-material physical properties, it is acceptable even if there is an unused material ID.

*If the format of single material physical properties is applied to mesh of multiple volumes, the material properties for the material are set to all volumes.

*If the format of multi-material physical properties is applied to mesh of a single volume, the material properties for the 0th volume are set to the mesh.

A.1.8. An Integrated FEM Model File (extension adv)

It is a binary file can be read and written by means of the ADVENTURE_IO.

A.1.9. A Boundary Condition File (extension cnd)

The format of a boundary condition file

```
-----------------------------------------------------------------
gravity
0 0 -9.8                  <- X,Y, and Z components of gravity acceleration
boundary 12               <- # of boundary condition data items
loadOnVertex 271 0 10.5   <- Load on node 271 in X direction is 10.5
loadOnVertex 271 1 10.5   <- Load on node 271 in Y direction is 10.5
loadOnVertex 271 2 10.5   <- Load on node 271 in Z direction is 10.5
dispOnVertex 8 0 0        <- Displacement on node 8 in X direction is fixed
dispOnVertex 8 1 0        <- Displacement on node 8 in Y direction is fixed
dispOnVertex 8 2 0        <- Displacement on node 8 in Z direction is fixed
tracOnFaceGroup 1 1 5.2   <- Normal surface traction of 5.2 on surface group 1
tracOnFaceGroup 2 0 2 -1.1 <- Surface traction of -1.1 on surface group 2 in Z direction
dispOnFaceGroup 3 0 0 0   <- Displacement on surface group 3 in X direction is fixed
dispOnFaceGroup 3 0 1 0   <- Displacement on surface group 3 in Y direction is fixed
dispOnFaceGroup 3 0 2 0   <- Displacement on surface group 3 in Z direction is fixed
dispOnFaceGroup 4 1 1     <- Normal displacement on surface group 4 is fixed
-----------------------------------------------------------------
```

loadOnVertex means a load on a node.

dispOnVertex means a displacement on a node.

tracOnFaceGroup means a surface traction on a face group

dispOnFaceGroup means a displacement on a face group

The 3 numbers following the loadOnVertex or the dispOnVertex mean the following:

First	Node ID
Second	0:X component, 1:Y component 2:Z component
Third	Magnitude of the load or the displacement

The numbers following the tracOnFaceGroup or dispOnFaceGroup mean the following:

First	Surface group ID	
Second	0:X, Y, or Z component	1:Normal to the surface group
Third	If XYZ option selected,	0:X, 1:Y, and 2:Z

	If normal option selected,	Magnitude of surface traction or displacement
Fourth	If XYZ option selected,	Magnitude of surface traction or displacement

Other than those presented above, the following analysis conditions exist.

*If boundary conditions do not change with time,

loadOnVertex nodeID co value	A co-axis load on a node with 'nodeID'.
presOnFaceGroup faceID 1 value	A normal pressure on a surface group with 'faceID'.
tracOnFaceGroup faceID 0 co value	A co-axis surface traction on a surface group with 'faceID'.
dispOnVertex nodeID co value	A co-axis displacement on a node with 'nodeID'.
dispOnFaceGroup faceID 0 co value	A co-axis displacement on a surface group with 'faceID'.
dispOnFaceGroup faceID 1 value	A normal displacement on a surface group with 'faceID'.
velocOnVertex nodeID co value	A co-axis velocity on a node with 'nodeID'.
velocOnFaceGroup faceID 0 co value	A co-axis velocity on a surface group with 'faceID'.
velocOnFaceGroup faceID 1 value	A normal velocity on a surface group with 'faceID'.
accelOnVertex nodeID co value	A co-axis acceleration on a node with 'nodeID'.
accelOnFaceGroup faceID 0 co value	A co-axis acceleration on a surface group with 'faceID'.
accelOnFaceGroup faceID 1 value	A normal acceleration on a surface group with 'faceID'.
tempOnVertex nodeID 0 value	A temperature on a node with 'nodeID'.
tempOnFaceGroup faceID 0 0 value	A temperature on a surface group with 'faceID'.
fluxOnFaceGroup faceID 0 0 value	A thermal flux on a surface group with 'faceID'.
transOnFaceGroup faceID temp coef.	A heat transfer to a surface group with 'faceID'.
radiOnFaceGroup faceID temp emissi. factor	A thermal radiation on a surface group with 'faceID'.

*If boundary conditions change with time,

Transient id loadOnVertex nodeID co value	A co-axis load on a node with 'nodeID'.
Transient id presOnFaceGroup faceID 1 value	A normal pressure on a surface group with 'faceID'.
Transient id tracOnFaceGroup faceID 0 co value	A co-axis surface traction on a surface group with 'faceID'.
Transient id dispOnVertex nodeID co value	A co-axis displacement on a node with 'nodeID'.
Transient id dispOnFaceGroup faceID 0 co value	A co-axis displacement on a surface group with 'faceID'.
Transient id dispOnFaceGroup faceID 1 value	A normal displacement on a surface group with 'faceID'.
Transient id velocOnVertex nodeID co value	A co-axis velocity on a node with 'nodeID'.
Transient id velocOnFaceGroup faceID 0 co value	A co-axis velocity on a surface group with 'faceID'.
Transient id velocOnFaceGroup faceID 1 value	A normal velocity on a surface group with 'faceID'.
Transient id accelOnVertex nodeID co value	A co-axis acceleration on a node with 'nodeID'.
Transient id accelOnFaceGroup faceID 0 co value	A co-axis acceleration on a surface group with 'faceID'.

Transient id accelOnFaceGroup faceID 1 value	A normal acceleration on a surface group with 'faceID'.
Transient id tempOnVertex nodeID 0 value	A temperature on a node with 'nodeID'.
Transient id tempOnFaceGroup faceID 0 0 value	A temperature on a surface group with 'faceID'.
Transient id fluxOnFaceGroup faceID 0 0 value	A thermal flux on a surface group with 'faceID'.
Transient id transOnFaceGroup faceID temp coef.	A heat transfer to a surface group with 'faceID'.
Transient id radiOnFaceGroup faceID temp emmisi. factor	A thermal radiation on a surface group with 'faceID'.

A.1.10. An MPC Condition File (extension cnd)

A cnd file is a file that is output by the BcGUI of the BCtool originally, and is intended to describe boundary conditions. For the purpose of MPCs, this format will be extended this time. This section describes and provides examples for each type below. The MPC ID will be unique regardless of the type of MPC conditions.

A.1.10.1. Rigid Beam

(1) Rigid Beam I

rbeamOnVertex Node ID MPCID Rigid Beam type

Example

```
boundary 2
rbeamOnVertex 3 0 1
rbeamOnVertex 51 0 1
```

where

boundary:	A label meaning # of conditions.
2:	# of MPC conditions of this example.
rbeamOnVertex:	A label meaning a Rigid Beam.
3 and 51:	Node IDs defined in a pch file respectively.
0:	Since the node 280 and node 309 are using a common MPC ID of 0, it is seen that these two nodes constitute a Rigid Beam.
1:	Meaning this condition is a Rigid Beam I.

Note: In the case of Rigid Beam I, a condition is decomposed when this program is executed, and the number of conditions will increase from one to three per node. Therefore, MPC IDs that are in the input data file will be ignored and will be numbered again from zero in the output data file. There is also some chance where generated IDs match those in the input file, but think that a totally different group of IDs will be used.

(2) Rigid Beam II

rbeamOnVertex Node ID MPCID Rigid Beam type

Example:It is assumed that the positions of two nodes do not match.

boundary 2

rbeamOnVertex 102 3 2

rbeamOnVertex 87 3 2

where

boundary: A label meaning # of conditions.

2: # of MPC conditions of this example.

rbeamOnVertex:A label meaning a Rigid Beam.

102 and 87: Node IDs defined in a pch file respectively.

3: Since the node 102 and node 87 are using a common MPC ID of 3, it is seen that these two nodes constitute a Rigid Beam.

2: Meaning this condition is a Rigid Beam II.

Note: In the case of the Rigid Beam II, the number of conditions will not change when this program is run. MPC IDs are numbered from zero per file without omission and without duplication. Note that IDs that are used in the input data file are not necessarily used in the output data file.

(3) Rigid Beam III

rbeamOnVertex Node ID MPCID Rigid Beam type

Example

boundary 2

rbeamOnVertex 280 0 3

rbeamOnVertex 309 0 3

where

boundary: A label meaning # of conditions.

2: # of MPC conditions of this example.

rbeamOnVertex:A label meaning a Rigid Beam.

280 と 309: Node IDs defined in a pch file respectively.

0: Since the node 280 and node 309 are using a common MPC ID of 0, it is seen that these two nodes constitute a Rigid Beam.

3: Meaning this condition is a Rigid Beam III.

Note: In the case of the Rigid Beam III, the number of conditions will not change when this program is run. MPC IDs are numbered from zero per file without omission and without duplication. Note that IDs that are used in the input data file are not necessarily used in the output data file.

(4) Rigid Beam IV

rbeamOnVertex Node ID MPCID Rigid Beam type Surface group # the node belongs to

Example

boundary 2

rbeamOnVertex 102 3 4 5

rbeamOnVertex 87 3 4 -1

where

boundary: A label meaning # of conditions.

2: # of MPC conditions of this example.

rbeamOnVertex:A label meaning a Rigid Beam.

102 and 87: Node IDs defined in a pch file respectively.

3: Since the node 102 and node 87 are using a common MPC ID of 3, it is seen that these two nodes constitute a Rigid Beam.

4: Meaning this condition is a Rigid Beam IV.

5 and -1: The surface groups that nodes 102 and 87 belongs to respectively (A normal vector associated with a node will be used. When a node matches a vertex or located on an edge, it will belong to more than one surface groups. Depending on which surface group the node belongs to, its normal vector is different. Therefore the user must specify the surface group to which the node belongs. Since it is sufficient to specify a surface group only for one of the node pair, the surface group that needs not be designated is written as -1.

Note: In the case of the Rigid Beam IV, the number of conditions will not change when this program is run. MPC IDs are numbered from zero per file without omission and without duplication. Note that IDs that are used in the input data file are not necessarily used in the output data file.

(5) Rigid Beam V

rbeamOnVertex Node ID MPCID Rigid Beam type 0

rbeamOnVertex Virtual node ID MPCID Rigid Beam type 1 x coord. y coord. z coord.

Example

boundary 4

rbeamOnVertex 3 4 5 0

rbeamOnVertex 94 4 5 0

rbeamOnVertex 162 4 5 0

rbeamOnVertex 754 4 5 1 25.0 30.0 35.0

where

boundary: A label meaning # of conditions.

4: # of MPC conditions of this example.

rbeamOnVertex: A label meaning a Rigid Beam.

3, 94, and 162 :Node IDs defined in a pch file respectively. Three nodes or more necessary.

754: A virtual node ID.

4: Since the nodes 182, 263, 309, and 754 are using a common MPC ID of 4, it is seen that these four nodes constitute a Rigid Beam.

5: Meaning this condition is a Rigid Beam V.

0 or 1: If 1, then nodal coordinates follow, if 0, then no coordinates follow. Only virtual nodes have coordinates.

25.0: An x coordinate of the virtual node.

30.0: A y coordinate of the virtual node.

35.0: A z coordinate of the virtual node.

Note: In the case of Rigid Beam V, a condition is decomposed when this program is executed, and the number of conditions will increase from one to three times the number of designated nodes. Therefore, MPC IDs that are in the input data file will be ignored and will be numbered again from zero in the output data file. There is also some chance where generated IDs match those in the input file, but think that <u>a totally different group of IDs will be used</u>.

A.1.10.2. Simple Beam

sbeamOnVertex Node ID 1 Node ID 2 MPCID Simple Beam load

Example

boundary 2

sbeamOnVertex 494 1 100.

sbeamOnVertex 515 1 100.

where

boundary: A label meaning # of conditions.

2: # of MPC conditions of this example.

sbeamOnVertex:A label meaning a Simple Beam.

494 and 515: Node ID 1 and Node ID 2 respectively.

1: The MPCID of the Simple Beam which consists of two node 494 and 515.

100.: A load to the Simple Beam (a positive number means tension, and a negative number means compression). The load value is specified duplicately.

Note: In the case of the Simple Beam, the number of conditions will not change when this program is run. MPC IDs are numbered from zero per file without omission and without duplication. Note that IDs that are used in the input data file are not necessarily used in the output data file.

A.1.10.3. MPC's about An Arbitrary Number of Nodes

lmpcOnVertex Node ID MPCID Axis Right hand term flag Coefficient Right hand term

Example

boundary 6

lmpcOnVertex 272 4 0 0 0.2 // Node 272, MPCID 4, X Axis, No right hand term(0), Coefficient

lmpcOnVertex 272 4 1 0 0.8 //Node 272, MPCID 4, Y Axis, No right hand term(0),Coefficient

lmpcOnVertex 272 4 2 1 -0.5 0.3 //Node272, MPCID 4, Z, With right hand term(1), Coefficient, Right hand term value

lmpcOnVertex 564 2 0 0 -1 // Node 564, MPCID 2, X Axis , No right hand term(0), Coefficient

lmpcOnVertex 564 2 1 0 1 // Node 564, MPCID 2, Y Axis, No right hand term(0), Coefficient

lmpcOnVertex 564 2 2 0 0.4 // Node 564, MPCID 2, Z Axis, No right hand term(0), Coefficient

where

boundary:	A label meaning # of conditions.
6:	# of MPC conditions of this example.
lmpcOnVertex:	A label meaning a Linear MPC.
272,564	Node IDs
4,2	MPCID。A constraint expression has a value of the right-hand side in the example of MPCID = 4. A constraint expression does not have a value on the right-hand side in the example of MPCID = 2.
0,1.2	A flag indicating an axis. If 0, then X axis, if 1, then Y axis, and if 2, then Z axis.
0,1	A flag to set the right-hand side. If 1, to set it, and if 0, not to set it.
0.2,0.8,-0.5,-1,1,0.4	Coefficients of the MPC.
0.3	A value of the right-hand side.

A.1.11. MPC Description File (extension mpc)

An MPC description file is an output file of the MpcLocal2Global, it is a file that matches the input format of the a2adv.pl in Part IV. Instead of describing the format, examples are shown. Note that MPC IDs after conversion are numbered from zero per file without omission.

A.1.11.1. Rigid Beam I

```
#Two sets of FEGAs must be prepared for each degree of freedom. The right-hand side
value is defined by the MPCRHS.
LinearConstraint    6       # # of lines for constraints
   3 0 0 1.                 # msh node ID 1 x MPC ID u1 coefficient
   51 0 0 -1.               # msh node ID 2 x MPC ID u2 coefficient
   3 1 1 1.                 #msh node ID 1 y MPC ID v1 coefficient
   51 1 1 -1.               #msh node ID 2 y MPC ID v2 coefficient
   3 2 2 1.                 #msh node ID 1 z MPC ID w1 coefficient
   51 2 2 -1.               #msh node ID 2 z MPC ID w2 coefficient
MPCRHS 3 # The total # of right-hand side conditions of the MPC
0 0 0.0            # msh node ID(borrowed) MPC ID Right-hand side value
1 1 0.0            # msh node ID(borrowed) MPC ID Right-hand side value
2 2 0.0            # msh node ID(borrowed) MPC ID Right-hand side value
```

Description

LinearConstraint i

LinearConstraint is a label indicating MPC, and i is the number of data lines that follow. For an MPC ID, it is the general rule to use the same ID across multiple lines. Therefore, note that the number of MPC IDs does not match the "i". The format of each line of the following is "i4i4i4f8", and a node ID, a dof ID, an MPC ID, and a coefficient are written, respectively. i4 indicates a four byte integer and f8 indicates a double precision real number.

An msh node ID

Note that this is not a node ID in a pch file. A pch node ID (local node ID) will be converted to an msh node ID (global node ID) by means of the trn data.

An axis

Either 0, 1, or 2. The 0 indicates the X-axis, the 1 indicates the Y-axis, and the 2 the Z-axis.

An MPC ID

An ID to uniquely define an MPC condition. In the case of the Rigid Beam I, two nodes are associated by this ID.

A coefficient

A coefficient to multiply each displacement component of each node.

MPCRHS 3

The total number of right-hand side conditions of this MPC is equal to 3.

0

A msh node ID. This line's data actually has nothing to do with the node ID 0. Because this line's data must be described as those attached to some node ID, a node number 0 is borrowed. A node ID is incremented by one for each line in the following lines (The format rule allows the node IDs to take discrete values, the purpose is not to have duplicate values).

0

An MPC ID to which this right-hand side value is related. Together with the data in which the MPC ID of LinearConstraint is 2, an MPC expression is constituted.

0.0

The right-hand side value of the expression for the MPC ID equal to 2.

A.1.11.2. Rigid Beam II

LinearConstraint 6 # # of constraint conditions Rigid Beam II	
102 0 0 0.	# msh node ID x MPC ID u_1 coefficient $(x_2 - x_1)$
87 0 0 0.	# msh node ID x MPC ID u_2 coefficient $-(x_2 - x_1)$
102 1 0 0.	# msh node ID y MPC ID v_1 coefficient $(y_2 - y_1)$
87 1 0 0.	# msh node ID y MPC ID v_2 coefficient $-(y_2 - y_1)$
102 2 0 10.	# msh node ID z MPC ID w_1 coefficient $(z_2 - z_1)$
87 2 0 -10.	# msh node ID z MPC ID w_2 coefficient $-(z_1 - z_1)$
MPCRHS 1 # The total # of right-hand side conditions of the MPC	
0 0 100.	# msh node ID(borrowed) MPC ID Right-hand side value $(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2$

Description

In this example of the Rigid Beam II, a constraint for the two nodes of 102 and 87 is specified. Therefore, the number of MPC IDs used is one. Since three degrees of freedom appear for each node, the total of six lines of LinearConstraint data are to be specified. The right-hand side is non-zero.

A.1.11.3. Rigid Beam III

LinearConstraint 6 # # of constraint conditions	
280 0 0 0.	# msh node ID x MPC ID u_1 coefficient $-(x_2 - x_1)$

```
309 0 0 0.          # msh node ID x MPC ID u2 coefficient (x2 - x1)
280 1 0 0.          # msh node ID y MPC ID v1 coefficient -(y2 - y1)
309 1 0 0.          # msh node ID y MPC ID v2 coefficient (y2 - y1)
280 2 0 -10.        # msh node ID z MPC ID w1 coefficient -(z2 - z1)
309 2 0 10.         # msh node ID z MPC ID w2 coefficient (z2 - z1)
MPCRHS 1 # The total # of right-hand side conditions of the MPC
0 0 0.    # msh node ID(borrowed) MPC ID Zero right-hand side
```

Description

In this example of the Rigid Beam III, a constraint for the two nodes of 280 and 309 is specified. Therefore, the number of MPC IDs used is one. Since three degrees of freedom appear for each node, the total of six lines of LinearConstraint data are to be specified. The right-hand side is zero.

A.1.11.4. Rigid Beam IV

```
LinearConstraint 6 # # of constraint conditions
102 0 7 -0.6           # msh node ID x MPC ID u1 coefficient -nx
87 0 7 0.6             # msh node ID x MPC ID u2 coefficient +nx
102 1 7 0.             # msh node ID y MPC ID v1 coefficient -ny
87 1 7 0.              # msh node ID y MPC ID v2 coefficient +ny
102 2 7 -0.8           # msh node ID z MPC ID w1 coefficient -nz
87 2 7 0.8             # msh node ID z MPC ID w2 coefficient +nz
MPCRHS 1 # The total # of right-hand side conditions of the MPC
0 7 0.    # msh node ID(borrowed) MPC ID Zero right-hand side
```

Description

In this example of the Rigid Beam IV, a constraint for the two nodes of 102 and 87 is specified. Therefore, the number of MPC IDs used is one. Since three degrees of freedom appear for each node, the total of six lines of LinearConstraint data are to be specified. Since three degrees of freedom appear for each node, the total of six lines of LinearConstraint data are to be specified. The right-hand side is zero.

A.1.11.5. Rigid Beam V

```
LinearConstraint    57      # # of constraint conditions
#The part about x-axis of equation (5-1) in section 6.1(5), Part V is applied to 1st node.
   3 0 0 0.03               # msh node ID 1 x MPC ID Coefficient
```

```
    3 1 0 -0.24                  # msh node ID 1 y MPC ID Coefficient
    3 2 0 0.51                   # msh node ID 1 z MPC ID Coefficient
    94 0 0 0.17                  # msh node ID 2 x MPC ID Coefficient
    94 1 0 -0.36                 # msh node ID 2 y MPC ID Coefficient
    162 0 0 0.48                 # msh node ID 3 x MPC ID Coefficient
# The part about y-axis of equation (5-1) in section 6.1(5), Part V is applied to 1st node.
    3 0 1 0.09                   # msh node ID1 x MPC ID Coefficient
    3 1 1 -0.12                  # msh node ID1 y MPC ID Coefficient
    3 2 1 0.76                   # msh node ID1 z MPC ID Coefficient
    94 0 1 0.38                  # msh node ID2 x MPC ID Coefficient
    94 1 1 -0.17                 # msh node ID2 y MPC ID Coefficient
    162 0 1 0.84                 # msh node ID3 x MPC ID Coefficient
#The part about z-axis of equation (5-1) in section 6.1(5), Part V is applied to 1st node.
    3 0 2 0.01                   # msh node ID1 x MPC ID Coefficient
    3 1 2 -0.52                  # msh node ID1 y MPC ID Coefficient
    3 2 2 0.89                   # msh node ID1 z MPC ID Coefficient
    94 0 2 0.13                  # msh node ID2 x MPC ID Coefficient
    94 1 2 -0.92                 # msh node ID2 y MPC ID Coefficient
    162 0 2 0.76                 # msh node ID3 x MPC ID Coefficient
# The part about x-axis of equation (5-1) in section 6.1(5), Part V is applied to 2nd node.
    3 0 3 0.02                   # msh node ID1 x MPC ID Coefficient
    3 1 3 0.28                   # msh node ID1 y MPC ID Coefficient
    3 2 3 -0.79                  # msh node ID1 z MPC ID Coefficient
    94 0 3 -0.11                 # msh node ID2 x MPC ID Coefficient
    94 1 3 -0.36                 # msh node ID2 y MPC ID Coefficient
    162 0 3 -0.54                # msh node ID3 x MPC ID Coefficient
# The part about y-axis of equation (5-1) in section 6.1(5), Part V is applied to 2nd node.
    3 0 4 -0.04                  # msh node ID 1 x MPC ID Coefficient
    3 1 4 0.57                   # msh node ID 1 y MPC ID Coefficient
    3 2 4 -0.38                  # msh node ID 1 z MPC ID Coefficient
    94 0 4 0.05                  # msh node ID 2 x MPC ID Coefficient
    94 1 4 -0.17                 # msh node ID 2 y MPC ID Coefficient
    162 0 4 -0.26                # msh node ID 3 x MPC ID Coefficient
#The part about z-axis of equation (5-1) in section 6.1(5), Part V is applied to 2nd node.
    3 0 5 -0.13                  # msh node ID 1 x MPC ID Coefficient
    3 1 5 0.66                   # msh node ID 1 y MPC ID Coefficient
```

```
    3 2 5 -0.47           # msh node ID 1 z MPC ID Coefficient
    94 0 5 0.14           # msh node ID 2 x MPC ID Coefficient
    94 1 5 -0.26          # msh node ID 2 y MPC ID Coefficient
    94 2 5 0.38           # msh node ID 2 z MPC ID Coefficient
    162 0 5 -0.35         # msh node ID 3 x MPC ID Coefficient
# The part about x-axis of equation (5-1) in section 6.1(5), Part V is applied to 3rd node.
    3 0 6 -0.41           # msh node ID 1 x MPC ID Coefficient
    3 1 6 0.09            # msh node ID 1 y MPC ID Coefficient
    3 2 6 -0.01           # msh node ID 1 z MPC ID Coefficient
    94 0 6 0.03           # msh node ID 2 x MPC ID Coefficient
    94 1 6 -0.77          # msh node ID 2 y MPC ID Coefficient
    162 0 6 0.78          # msh node ID 3 x MPC ID Coefficient
# The part about y-axis of equation (5-1) in section 6.1(5), Part V is applied to 3rd node.
    3 0 7 0.89            # msh node ID 1 x MPC ID Coefficient
    3 1 7 0.91            # msh node ID 1 y MPC ID Coefficient
    3 2 7 -0.12           # msh node ID 1 z MPC ID Coefficient
    94 0 7 -0.21          # msh node ID 2 x MPC ID Coefficient
    94 1 7 -0.42          # msh node ID 2 y MPC ID Coefficient
    162 0 7 0.13          # msh node ID 3 x MPC ID Coefficient
    162 1 7 0.39          # msh node ID 3 y MPC ID Coefficient
# The part about z-axis of equation (5-1) in section 6.1(5), Part V is applied to 3rd node.
    3 0 8 -0.44           # msh node ID 1 x MPC ID Coefficient
    3 1 8 0.35            # msh node ID 1 y MPC ID Coefficient
    3 2 8 -0.04           # msh node ID 1 z MPC ID Coefficient
    94 0 8 -0.07          # msh node ID 2 x MPC ID Coefficient
    94 1 8 -0.92          # msh node ID 2 y MPC ID Coefficient
    162 0 8 0.43          # msh node ID 3 x MPC ID Coefficient
    162 2 8 0.61          # msh node ID 3 z MPC ID Coefficient
MPCRHS 9 # The total # of right-hand side conditions of the MPC
    0 0 0.0 # msh node ID (borrowed) MPC ID Right-hand side value
    1 1 0.0 # msh node ID (borrowed) MPC ID Right-hand side value
    2 2 0.0 # msh node ID (borrowed) MPC ID Right-hand side value
    3 3 0.0 # msh node ID (borrowed) MPC ID Right-hand side value
    4 4 0.0 # msh node ID (borrowed) MPC ID Right-hand side value
    5 5 0.0 # msh node ID (borrowed) MPC ID Right-hand side value
    6 6 0.0 # msh node ID (borrowed) MPC ID Right-hand side value
```

```
7 7 0.0 # msh node ID (borrowed) MPC ID Right-hand side value
8 8 0.0 # msh node ID (borrowed) MPC ID Right-hand side value
```

Description

An MPC ID

An ID to uniquely define an MPC condition. In the case of the Rigid Beam V, three nodes or more are associated by this ID. The above example is the case when the number of nodes to be used for constraints is three.

0(at the top of line following the definition of the total number of MPCRHSs)

A msh node ID. This line's data actually has nothing to do with the node ID 0. Because this line's data must be described as those attached to some node ID, a node number 0 is borrowed. A node ID is incremented by one for each line in the following lines.

0(the second data item of the line following the definition of the total number of MPCRHSs)

An MPC ID to which this right-hand side value is related. Together with the data in which the MPC ID of LinearConstraint is 0, an MPC expression is constituted.

0.0(the third data item of the line following the definition of the total number of MPCRHSs)

The right-hand side value of the expression for the MPC ID equal to 0. Incidentally, it is always equal to zero for the Rigid Beam V .

A.1.11.6. Simple Beam

Let the node ID 494 be P_1 (5., 7.,0.), node ID 515 be P_2 (10., 7., 0.), a load value be 100., and an MPC ID be 1. Then,

```
SimpleBeam 2 # # of constraint conditions
494 1 100. # msh node ID   MPC ID Algebraic load value (positive value is tensile)
515 1 100. # msh node ID   MPC ID Algebraic load value(positive value is tensile)
```

A.1.11.7. General MPCs

To specify a general MPC, the label LinearConstraint is used except for specifying a right-hand side value.

A right-hand side is described by a FEGA labeled as MPCRHS. A right-hand side value is specified for each node ID, but the node ID does not have anything to do with the right-hand side value. To which MPC condition a certain right-hand side value corresponds is identified by the MPC ID. The node ID can be any number. But duplicate use of a same node ID is not allowed. Therefore, IDs will be incremented from zero to avoid duplicate numbering mistakes.

The reason why such data format is adopted is because a FEGA can only be created as data associated with a certain node. It is desirable to use a node ID that is actually related to the MPC condition of interest, but when the node is involved with more than one MPC, the idea will collapse. Therefore, the right-hand side has no choice but to use node IDs starting from zero and sequentially like this.

```
LinearConstraint 9 # # of constraint conditions MPCs for arbitrary # of nodes
3 0 5 0.2          # msh node ID x MPC ID Coefficient for u1
3 1 5 0.8          # msh node ID y MPC ID Coefficient for v1
3 2 5 -0.5         # msh node ID z MPC ID Coefficient for w1
319 0 5 -1         # msh node ID x MPC ID Coefficient for u2
319 1 5 1          # msh node ID y MPC ID Coefficient for v2
319 2 5 0.4        # msh node ID z MPC ID Coefficient for w2
127 0 5 -1.2       # msh node ID x MPC ID Coefficient for u3
127 1 5 0.7        # msh node ID y MPC ID Coefficient for v3
127 2 5 0.1        # msh node ID z MPC ID Coefficient for w3
MPCRHS 1           # Total # of MPC conditions
0 5 1.0 # msh node ID (borrowed) MPC ID Right-hand side c in eq. in Sec. 6.3, Part V
```

Note that there are cases where the line of the SimpleBeam or the MPCRHS is followed by the following data.

```
fega_type=NodeVariable     #The data type associated with the node
format=i4f8                #Data format. The format of the node ID is omitted.
```

This data is added by the a2adv if not accompanied.

A.1.12. Surface Group Pair File (extension cmb)

The format of a surface group pair file.

```
4 # Number of pairs of surfaces
3 8 #1st master surface ID slave surface ID
4 9 #2nd master surface ID slave surface ID
6 10 #3rd master surface ID slave surface ID
7 12 #4th master surface ID slave surface ID
```

A.1.13. Node Pair Related Files (extensions, np and nv)

A.1.13.1. The Format of Node Pairs Output

```
31193            # # of node pairs
794 801256       # 1st pair's 1st node ID, 2nd node ID
795 801258       # 2nd pair's 1st node ID, 2nd node ID
835 801259
836 801260
837 801261
… …
… …
757503 1012239                 #31193rd pair's 1st node ID, 2nd node ID
```

Fig. 5.1-1 Node pair file format (extension np)

A.1.13.2. The Output Format of Normal Vectors at Node Pairs

```
31193              # Total # of node pairs
0 0.047682 0.924802 –0.377448          #1st node pair ID, Normal vector of 1st node
0 –0.047912 –0.924919 0.377132         #1st node pair ID, Normal vector of 2nd node
1 0.047444 0.924918 –0.377194          #2nd node pair ID, Normal vector of 1st node
1 –0.047996 –0.925217 0.376391         #2nd node pair ID, Normal vector of 2nd node
2 0.044090 0.925059 –0.377255
2 –0.046319 –0.925042 0.377030
… …
… …
```

(Number of node pairs) times (2 lines)

Fig. 5.2-1 The file format for normal vectors at node pairs (extension nv)

A.2. Available Data Types (FEGA Document)

(i4 indicates (4 byte) integer, and f8 indicates (8 byte) real number.

Boundary Conditions

*Nodal forced displacement

Label name: ForcedDisplacement

Property to be added: None

Data structure of a line: Node ID (i4) Coordinate axis (i4)   Displacement value (f8)

*Nodal concentrated load

Label name: Load

Property to be added: None

Data structure of a line: Node ID (i4) Coordinate axis (i4) Load value (f8)

* To designate a coordinate axis, use the following rule.

x axis	0
y axis	1
z axis	2

Material properties

*Young's modulus

Label name: YoungModulus

Property to be added: None

Data structure of a line: Value (f8)

*Poisson's ratio

Label name: PoissonRatio

Property to be added: None

Data structure of a line: Value (f8)

*Work hardening parameter

Label name: HardeningParameter

Property to be added: None

Data structure of a line: Value (f8)

*Initial yield stress

Label name: YieldStress

Property to be added: None

Data structure of a line: Value (f8)

*Mass density

Label name: Density

Property to be added: None

Data structure of a line: Value (f8)

Others

*General MPCs

Label name: LinearConstraint

Property to be added: None

Data structure of a line: Node ID (i4) Axis (i4) MPC ID (i4) Coefficient Value (f8)

*Gravity acceleration vector

Label name: GravityAcceleration

Property to be added: None

Data structure of a line: x (f8) y(f8) z(f8)

(x, y, and z components of gravity acceleration)

A.3. Registration of a New Element Type and a New FEGA

To add a new element type, add its format to where the %element_format is defined in the a2adv.pl.in.

e.g.

```
%element_format = (
              .
   'NewElementType'  => i4i4i4i4, #4
              .
```

To add a new FEGA, add its fega_type to where the %fega_type is defined in the a2adv.pl.in and further add its format to where the %fega_format is defined.

e.g.

```
%fega_types = (
            .
   'NewFEGALabel'  => AllNodeVariable,
            .
%fega_format = (
            .
   'NewFEGALabel'  => i4f8,
            .
```

Note that, even for an unregistered FEGA, by describing a fega_type and a format property in the input text file, it can be converted to ADVENTURE_IO format.

Refer to the example file of samples/a2adv/unknownlabel.dat.

A.4. About the FEGA

The ADVENTURE_IO handles input and output in data units called Document. One file contains one or more Documents, and each Document consists of the following three components.

*Document ID
*Property
*Raw Data

A Document ID is intended to uniquely identify the Document, and it can be created by means of a library function. In the Property section, explanation of what the Document is about and options to be passed to each program are written. In order to specify a Property, a Key = Value combination will be used. In the Raw Data part, mass data such as coordinates, and various physical variables are stored in the binary format.

In order to describe a Document, a key "content_type" will be used. As standard content_type's, Element, Node, and HDDM_Element are available, and as generic content_type's, FEGenericAttribute (FEGA) and HDDM_FEGenericAttribute are available. Since the FEGA is generic, it is necessary to define its contents by writing information of the fega_format and the fega_type.