ADVENTURE_Shape

Nonparametric shape optimization based on traction method
Topology optimization based on density approach

Version: 0.11 (beta)

User's Manual

March 1, 2002

ADVENTURE Project

Contents

1. Outline

This document contains information on handling of the program modules ADVENTURE_Shape for nonparametric optimizations of shape and topology designed in the ADVENTURE_Project [1]. ADVENTURE_Shape has the following features.

- ADVENTURE_Shape performs two types of optimizations: optimization of shape and optimization of topology.

- ADVENTURE_Shape modules use the parallel finite element analysis solver ADVENTURE_Solid and the domain decomposer ADVENTURE_Metis (designed in ADVENTURE Project) for optimization analysis (only elastic analysis features of solver can be used).

- ADVENTURE_Shape supports linear tetrahedral, quadratic tetrahedral, linear hexahedral, and quadratic hexahedral elements without their combinations in one model.

- ADVENTURE_Shape modules are designed to operate in UNIX and Linux environments.

The algorithm of analysis by ADVENTURE system using ADVENTURE_Shape is shown in *Fig. 1*. The analysis consists of the following steps.

(1). Creation of mesh.
ADVENTURE_TetMesh creates the finite element mesh of analysis model.

(2). Creation of analysis model.
Boundary conditions are set to the mesh using the pre-processor module ADVENTURE_BCtool, which later creates an entire-type analysis model data file. A shape-restricted model is created after addition of shape optimization to the finite element analysis model. Topology optimizations do not create the shape-restricted model. Detailed information on the shape-restricted model is given in *Sections 5.1* and *5.5.3*.

(3). Optimizations of shape and topology.
ADVENTURE_Shape modules perform the shape and topology optimizations executing ADVENTURE_Metis and ADVENTURE_Solid modules (*Fig. 1*).

(4). Visualization of results.
The results of analysis (optimized shape, optimized topology, displacements, and stresses) can be visualized using the post-processor ADVENTURE_Visual.

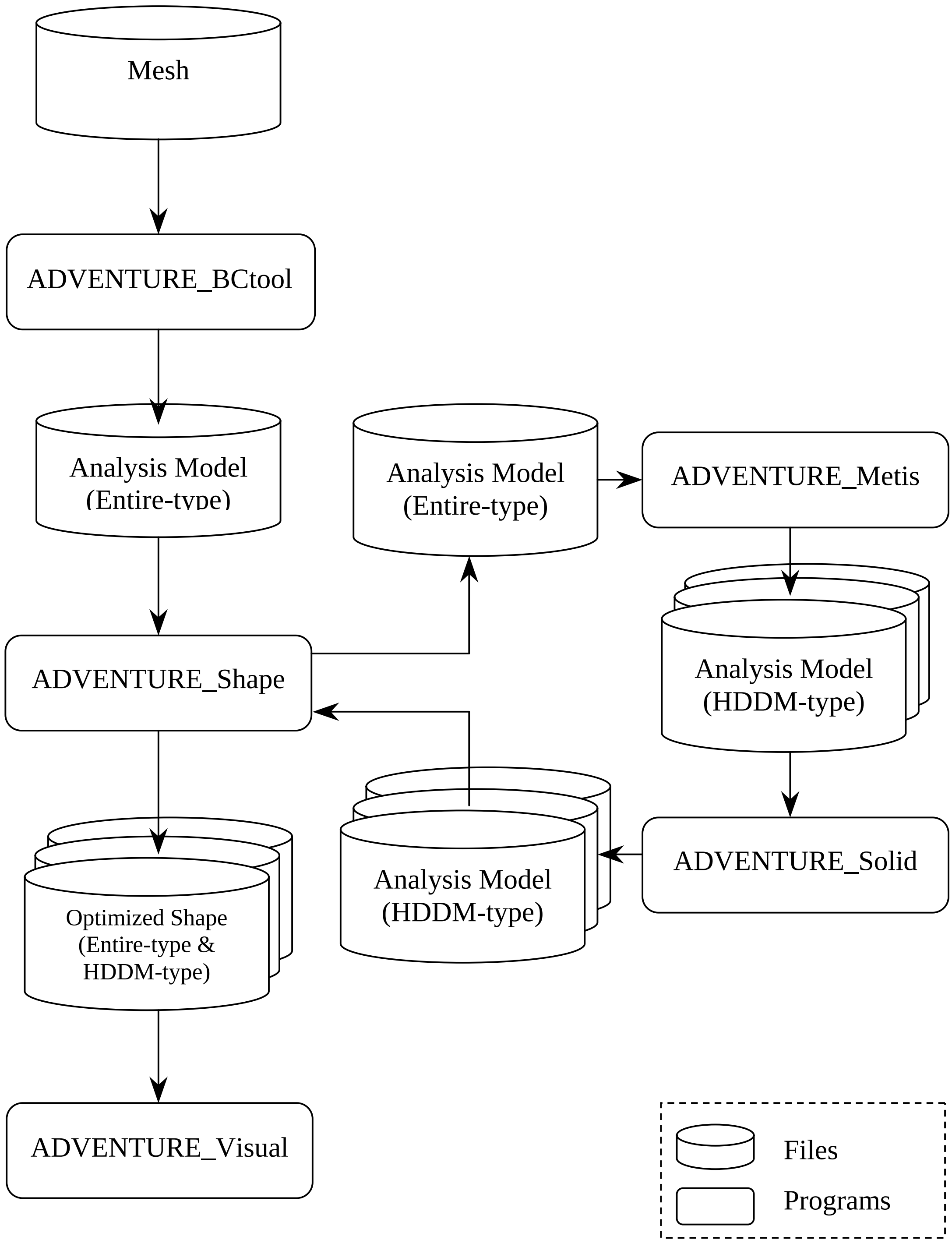


Fig. 1. Analysis by ADVENTURE System Using ADVENTURE_Shape

2. Program Features

ADVENTURE_Shape supports two kinds of optimization: the shape optimization and the topology optimization. An optimum shape of the model is achieved by changing its surfaces (shape optimization process), grouping of inner elements and removing them from the model (topology optimization process). ADVENTURE_Shape consists of 2 executable modules: the module for shape optimization `advshape`, and the module for optimization of topology `advtopology`. Both modules use ADVENTURE_Metis and ADVENTURE_Solid for calculations.

2.1. Optimization of Shape

Shape optimization uses the Traction method [2,3,4]. It can solve volume minimization problems for linear elastic bodies restricted by mean compliance with respect to given external loads. After evaluation of boundary shape gradients, the boundaries are moved by virtual external forces, which are proportional to the negative value of the shape gradient function. The model's shape remains smooth even after its boundaries have been moved.

2.2. Optimization of Topology

Topology optimization uses the Density approach [5], which performs minimization of mean compliance with respect to the given external forces restricted by mass conditions. The density approach assumes that the elastic stiffness of material is proportional to power of its density and uses the density ratio as a design parameter for finite element analyses. Consequently, the optimum topology is obtained by distribution of density ratio, which values are in range from 0 to 1.

3. Operating Environment

ADVENTURE_Shape modules require the following computing environment.

Operating System:	UNIX or Linux.
Compiler:	GNU gcc Ver. 2.95.3 or higher.
Library module:	ADVENTURE_IO
Program modules:	ADVENTURE_Metis and ADVENTURE_Solid.

4. Installation Method

4.1. Compilation

Prior to compile the ADVENTURE_Shape modules, you need to install the library module ADVENTURE_IO released in ADVENTURE Project. To use the ADVENTURE_Shape modules, you need the domain decomposer ADVENTURE_Metis and the matrix solver ADVENTURE_Solid (both are the components of ADVENTURE system). ADVENTURE_Shape can be compiled by the following steps.

(1). Decompress the archive `AdvShape-0.11b.tar.gz`.

```
% tar -xvzf AdvShape-0.11b.tar.gz
```

(2). Move to the top directory `AdvShape-0.11b`.

```
% cd AdvShape-0.11b
```

(3). Open the file `Makefile` by any text editor and set the absolute path to the top directory, which contains the script `advsys-config` of the ADVENTURE_IO library.

```
ADVSYSDIR = $(HOME)/ADVENTURE/bin
```

If ADVENTURE_IO has been installed using default settings, the directory should be `$(HOME)/ADVENTURE/bin`. If any other path is used, change `Makefile` according to your computing environment. Set the compiler to be used. The default compiler is `gcc`. In some environments, `cc` compiler can be set instead of `gcc`.

```
CC = gcc
```

(4). Execute the command `make`.

```
% make
```

If the compilation process ended up successfully, the executable modules `advshape`, `advtopology`, and `topo_tool` will be created in the directory `src`. If the compilation process failed, the macros `CC`, `LIBS`, and `CFLAGS` should be set manually according to the computing environment. Here, `CC` is the macro for *C* compiler, `CFLAGS` is the macro for *C* compiler's options, and `LIBS` is the macro for libraries.

4.2. Installation

To install documentation and executable modules into a specified directory, execute the command

```
% make install
```

By default, the executables will be copied into the directory `$(HOME)/ADVENTURE/bin`, and the documentation will be copied into the directory `$(HOME)/ADVENTURE/doc/AdvShape`. To change the directories for installation, use the command

`% make install prefix =` *install_dir*

Here, *install_dir* is the directory for installation, which should be specified by an absolute path. To use ADVENTURE_Shape together with ADVENTURE_Metis and ADVENTURE_Solid, the paths to each module should be set.

5. Shape Optimization

5.1. Input Data

The following 5 input files should be prepared to execute `advshape`.

(1). The entire-type finite element analysis (FEA) model file (file extension: `.adv`).
The entire-type FEA model created using the module ADVENTURE_BCtool released in ADVENTURE Project (refer to its User's Manual for operation instructions) is saved in one binary file of ADVENTURE format. The sensitivity (shape gradient) is evaluated using the analysis results of this FEA model. The evaluated sensitivity is used in later analyses of the shape-restricted model (so-called velocity field analyses).

(2). The entire-type shape-restricted model file (file extension: `.adv`).
The entire-type shape-restricted model is used to determine displacement fields (velocity fields) at the time when the negative value of shape gradient acts on boundaries as external forces. The shape of the model is changed in accordance with displacement fields obtained from analysis using ADVENTURE_Solid. The shape optimization processes can be controlled by setting of displacement boundary conditions to the parts of the model, which should not be changed. If displacement and load boundary conditions are set for nodes and surfaces of the entire-type FEA model, the fixed displacements must be set in the direction of normal. It must be done in order to reach the theoretical solution for optimization and prevent rigid motions. The fixed displacement boundary conditions can be set using ADVENTURE_BCtool, however the load boundary conditions are obtained from the shape gradient data of analyzed FEM model. The shape-restricted model file has ADVENTURE binary format.

(3). The shell script file for ADVENTURE_Metis (file extension: `.sh`).
The executable module `advshape` invokes the domain decomposer ADVENTURE_Metis using a shell script file. Setup methods for this shell script is described in *Section 5.5.4*.

(4). The setup file for `advsolid` (file extension: `.conf`).
This setup file is used by `advshape` to execute a linear elastic analysis by ADVENTURE_Solid. Paths to the executable modules should be specified. Refer to the User's Manual of ADVENTURE_Solid for detailed information on this file. Because, `advshape` and `advtopology` use the node displacement data, the results must be printed to files using the option "`-- disp`". The entire-type FEA model file and the entire-type shape-restricted model file are treated equally.

(5). The setup file for `advshape` (file extension: `.conf`).
The setup file for `advshape` contains data (parameters) necessary for analysis. The contents of setup file are shown in *Fig. 2*. An advantage of the Traction

method lies in small mesh distortion, because adequate movements of inner nodes accompany the changes of the model's shape. To prevent large distortion of mesh, which can be accumulated after some repetitions of analysis, the maximum mesh distortion at the time of shape updating is set by the restriction parameter `MAX_STR`. A ratio of the mean compliance at the end of optimization to the mean compliance of the initial shape should also be set by the restriction parameter `SUBJ_RATIO`. "`SUBJ_RATIO = 1.0`" means that the computations will be done with equal mean compliance at all shape updates.

IT_MAX	30	← Maximum number of iterations for optimization
MAX_STR	0.3	← Maximum distortion for shape updating
SUBJ_RATIO	1.0	← Restriction for mean compliance
SUBJ_REL_ERROR	1.0E-4	← Tolerance for relative error of restriction
OBJ_REL_ERROR	1.0E-4	← Judgment for relative convergence of volume

Fig. 2. Format of Setup File for `advshape`

5.2. Execution Method

The `advshape` module can be executed using a command of the following format.

`% advshape` [*options*] *advmetis_sh advsolid_conf advshape_conf fem_model rest_model data_dir*

[*options*]	the output options for `advshape`
advmetis_sh	the name of ADVENTURE_Metis shell script file
advsolid_conf	the name of setup file for `advsolid`
advshape_conf	the name of setup file for `advshape`
fem_model	the name of the entire-type FEA model data file
data_dir	the name of the top directory containing input and output data for ADVENTURE_Solid

5.3. Command Options

The following options can be used for execution of `advshape`.

If you are going to use non-default names for files and directories, use different names for ADVENTURE_Metis, ADVENTURE_Solid, and ADVENTURE_Shape.

- `-model-file` *file*

  The option should be used to specify the name of input FEA model file (if a non-default filename is considered). This option is same as used with ADVENTURE_Solid.

- `-model-dir` *dir*
  The option should be used to specify the name of directory with input FEA model files (if a non-default name is considered for directory). This option is same as used with ADVENTURE_Solid.

- `-result-file` *file*
  The option should be used to specify the name of output file (if a non-default filename is considered). This option is same as used with ADVENTURE_Solid.

- `-result-dir` *dir*
  The option should be used to specify the name of directory with output files (if a non-default name is considered for directory). This option is same as used with ADVENTURE_Solid.

- `-log` *logfile*
  The execution log of `advshape` displayed on the screen will be printed to the file *logfile*`.log`. The saved data are: mean compliance and volume compliance at each optimization step. The file extension `.log` will be automatically added to the filename specified by *logfile*.

- `-output-shape` *file* [*sub-options*]
  The model with optimized shape will be saved to an entire-type file *file_F*`.adv`. The step number *F* of final repetition of calculations and the extension `.adv` will be added to the filename automatically. The option "`-output-shape`" can be followed by sub-option "`--interval`". To view the model by ADVENTURE_Visual, the entire-type model data created at this step should be decomposed using ADVENTURE_Metis. After decomposition, the model can be analyzed using the FEA solver ADVENTURE_Solid. If the option "`-output-shape`" is not specified, the model will be saved after the last step *F* of optimization with the name `adv_shape_`*F*`.adv` (default filename).

- `--interval` *num*
  This sub-option can be used together with the option "`-output-shape`". The model will be saved each *num* times into the file *file_N*`.adv`. *N* is the repetition step number of the analysis. The sub-option "`--interval`" must be placed after the option "`-output-shape`". If this sub-option is not specified, the model will be saved at the step where the calculations have been converged, or the number of repetitions `IT_MAX` has been overcome without convergence.

5.4. Output Data

The following data can be printed out.

- The program execution log (file extension is `log`).
  Information on the volume and mean compliance of the model will be printed out for each optimization step. If the option described in *Section 5.3* is not specified, only standard output will be done.

- The shape of linear elastic body with minimized volume restricted by conditions of mean compliance (entire-type data file, HDDM-type data files).
  The model with optimized shape will be saved into entire-type and HDDM-type files. If the option "`-output-shape`" is not specified, the model will be saved after the last step *F* of optimization with the default filename `adv_shape_F.adv`. The data will be saved by default into the HDDM-type files *data_dir*`/model/advhddm_in_P.adv`, where *data_dir* is the top directory for the data and *P* is the *Part* number. The shape of the model can be visualized using ADVENTURE_Visual.

- The stresses and displacements in the optimized model (HDDM-type data files).
  The results of analysis by ADVENTURE_Solid can be printed to files using the corresponded options. The default output will be done to files *data_dir*`/result/advhddm_out_P.adv`, where *data_dir* is the top directory for data and *P* is the *Part* number. The results can be visualized using ADVENTURE_Visual.

5.5. Example of Analysis

An example of analysis which model is shown in *Fig. 3* will be briefly discussed here. One vertical surface of the model is fixed in all directions. The distributed load of –1 is applied to the surface, which is opposite to the fixed surface (as it is shown in *Fig. 3*). Since, the model has symmetry, it is possible to analyze only half of the model (40 x 20 x 10).

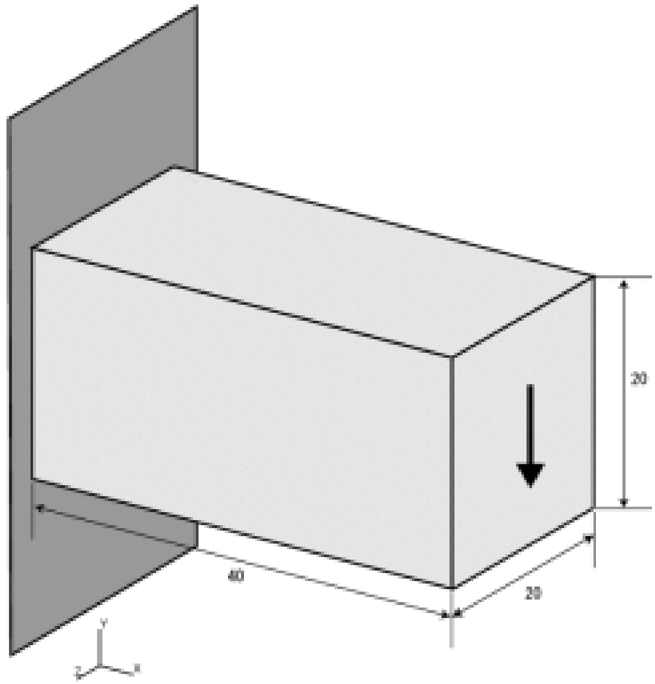


Fig. 3. Model for Analysis

5.5.1. Extraction of Mesh Surface

(1). Create any directory, which could be used for analysis.

(2). Copy the files `ADVShape-0.11b/sample_data/sample1.msh` (mesh data file) and `ADVShape-0.11b/sample_data/material.dat` (material properties data) into the created directory and execute the command

```
% msh2pch sample1.msh 3
```

The number "3" in the filename represents that the angle between mesh surfaces is set to 60 degrees. The following files will be created.

`sample1_3.frg` (File with mesh surface data)
`sample1_3.pch` (File with mesh surface patches)
`sample1_3.pcg` (File with surface patch group data)
`sample1_3.trn` (Global index file)

5.5.2. Creation of Entire-type Analysis Model Data

- To create the entire-type model data file using GUI, start the ADVENTURE_BCtool module by the following command.

```
% bcGUI sample1_3.pch sample1_3.pcg
```

  Displacement boundary conditions can be set using GUI as it is shown in *Fig. 4* and *Fig. 5*.

- Select the left surface (*Fig. 4a*) and fix it in X, Y, and Z directions using the dialog shown in *Fig. 4b*.

- Since, only half of the model will be used for analysis, select the surface that has symmetry and fix it in Z direction to prevent rigid motion (*Fig. 5*).

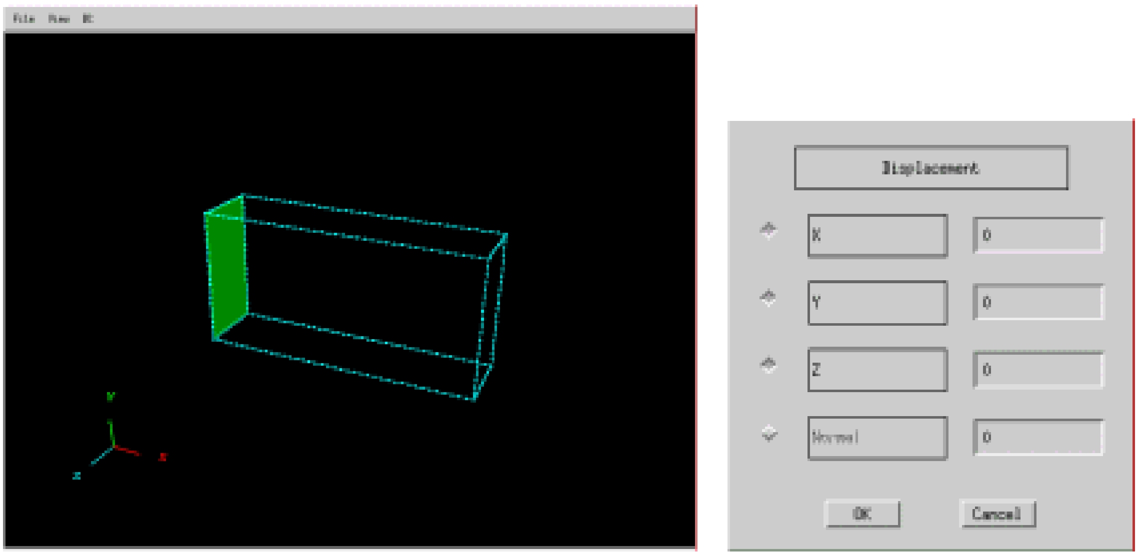
- Set the load by selecting the surface (*Fig. 6a*) and entering –1 in the Y dialog box of the setup dialog (*Fig. 6b*).

- Save the boundary conditions data using the submenu *Save Condition* of the menu *File*. The boundary conditions file `fem_model.cmd` for entire-type model will be created (here its name is defined as `fem_model.cmd`). The contents of this file are shown in *Fig. 7*.

- Create the entire-type FEA model data file using the command

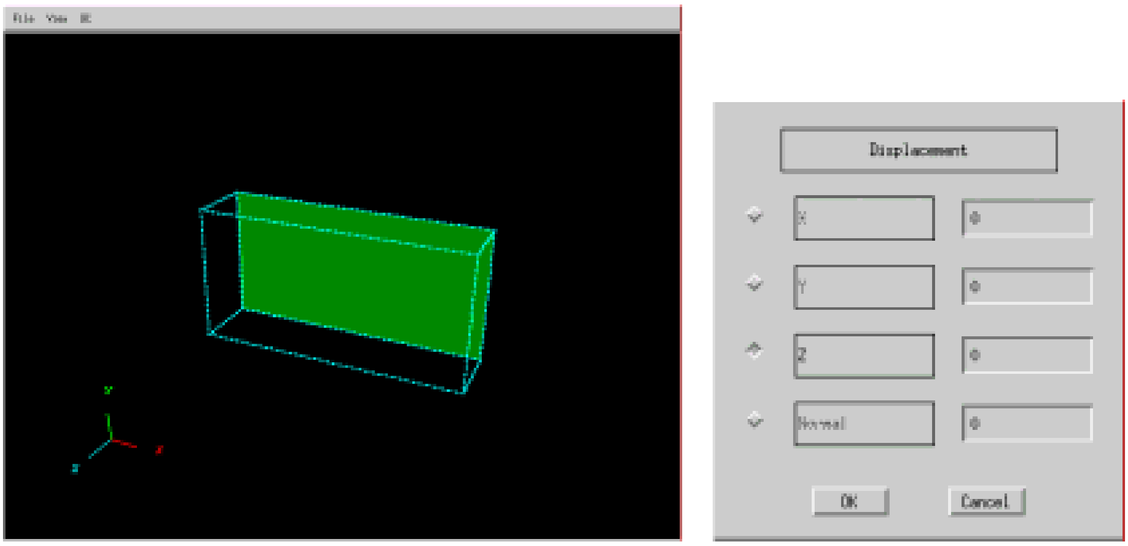
```
% makefem sample1.msh sample1_3.frg fem_model.cnd
            material.dat fem_model.adv –t sample1_3.trn
```

  The entire-type FEA model file `fem_model.adv` will be created in binary ADVENTURE format (here its name is defined by the 5th command option as `fem_model.adv`).



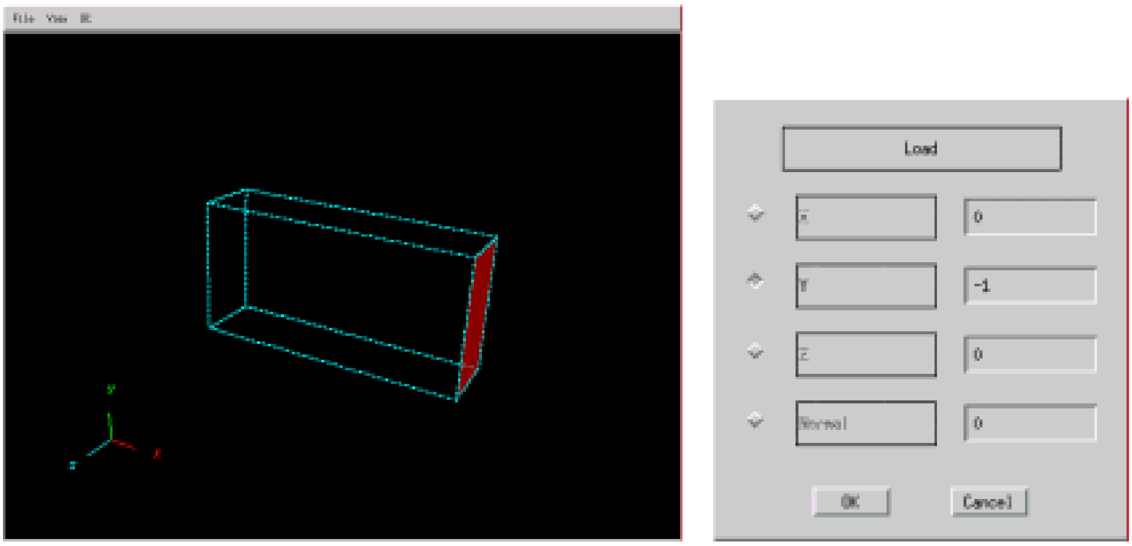
(a). Selected Surface *(b). Setup Dialog*

Fig. 4. Setup of Displacement Boundary Conditions



(a). Selected Surface *(b). Setup Dialog*

Fig. 5. Setup of Displacement Boundary Conditions



(a). Selected Surface *(b). Setup Dialog*

Fig. 6. Setup of Load

```
gravity 0 0 0
boundary 5
loadOnFaceGroup 3 0 1 -1
dispOnFaceGroup 0 0 2 0
dispOnFaceGroup 1 0 0 0
dispOnFaceGroup 1 0 1 0
dispOnFaceGroup 1 0 2 0
```

Fig. 7. Contents of Boundary Conditions File (fem_model.cnd)

5.5.3. Creation of Entire-type Shape-Restricted Model

The entire-type shape-restricted model can be created using ADVENTURE_BCtool's GUI in the way it was done for the entire-type analysis model.

- To create the entire-type shape-restricted model data file using GUI, start the ADVENTURE_BCtool module by the following command.

```
% bcGUI sample1_3.pch sample1_3.pcg
```

  Displacement boundary conditions can be set using GUI as it is shown in *Fig. 8* and *Fig. 9*.

- Select the left and right surfaces (*Fig. 8a*) and fix them in X direction using the dialog shown in *Fig. 8b*.

- Select the upper and lower surfaces (*Fig. 9a*) and fix them in Y direction (*Fig. 9b*).

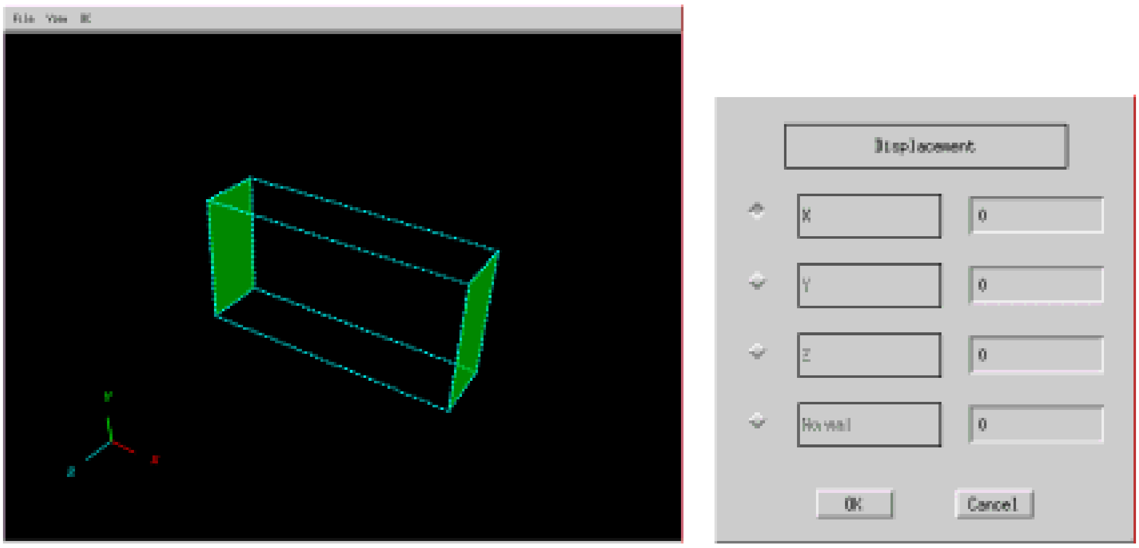
- Since, only half of the model will be used for analysis, select the surface that has symmetry and fix it in Z direction to prevent rigid motions (*Fig. 10*).

- Save the boundary conditions data using the submenu *Save Condition* of the menu *File*. The boundary conditions file `rest_model.cmd` for entire-type shape-restricted model will be created (here its name is defined as `rest_model.cmd`). The contents of this file are shown in *Fig. 11*.

- Create the entire-type model data file using the command

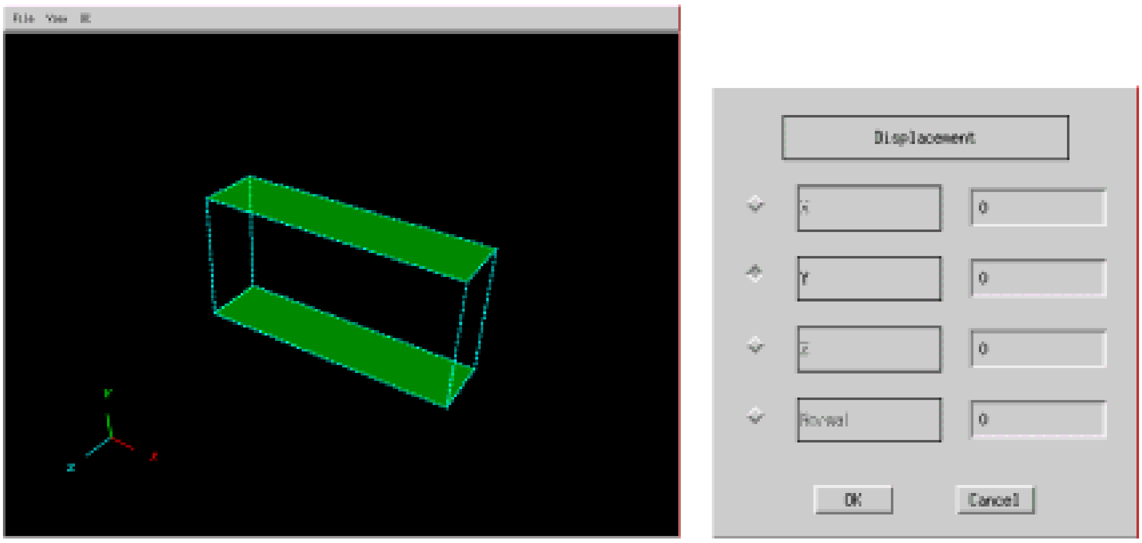
```
% makefem sample1.msh sample1_3.frg rest_model.cnd
            material.dat rest_model.adv –t sample1_3.trn
```

  The entire-type model file `rest_model.adv` will be created in binary ADVENTURE format (here, its name is defined by the 5th command option as `rest_model.adv`).



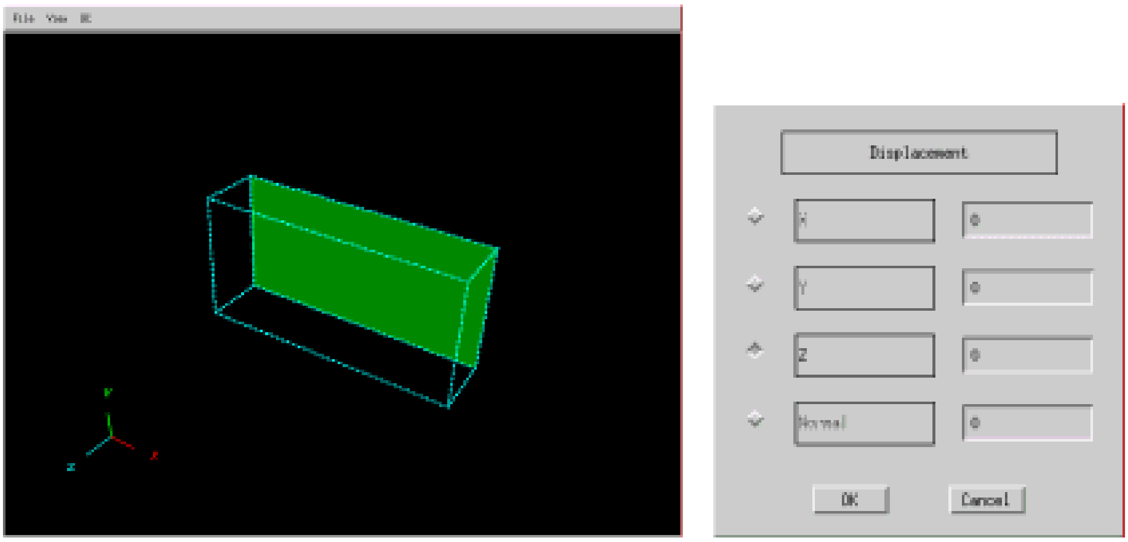
(a). Selected Surfaces *(b). Setup Dialog*

Fig. 8. Setup of Displacement Boundary Conditions



(a). Selected Surfaces *(b). Setup Dialog*

Fig. 9. Setup of Displacement Boundary Conditions



(a). Selected Surfaces *(b). Setup Dialog*

Fig. 10. Setup of Displacement Boundary Conditions

```
gravity 0 0 0
boundary 5
dispOnFaceGroup 0 0 2 0
dispOnFaceGroup 1 0 0 0
dispOnFaceGroup 2 0 1 0
dispOnFaceGroup 3 0 0 0
dispOnFaceGroup 4 0 1 0
```

Fig. 11. Contents of Boundary Conditions File (`rest_model.cnd`*)*

5.5.4. Creation of Executable Script for ADVENTURE_Metis

An example of shell script file (here, the name `advmetis.sh` is used) for ADVENTURE_Metis is shown in *Fig. 12*. It should be changed in accordance with the user's computing environment. This shell script file can be created as a plain text. The program execute this shell script using `/bin/sh`.

The 8th option in the command line of *Fig. 12* sets the filename of entire-type analysis model data. This filename is fixed and must be specified, because ADVENTURE_Solid saves the model data with file extension of `.adv` at each optimization step.

Long line command continued with a backslash before the beginning of a new line.

```
mpirun -np 12 -machinefile machine \
        /usr/local/Adventure/bin/adventure_metis –ls \
        metis.log adventure_shape_temp_file.adv . 12
```

Fig. 12. Example of Shell Script for ADVENTURE_Metis (advmetis.sh)

5.5.5. Creation of Setup File for Executable Script `advsolid`

An example of setup file (here, its name is `advsolid.conf`) for `advsolid` is shown in *Fig. 13*. It should be changed in accordance with computing environment. Refer to the User's Manual for ADVENTURE_Solid for details.

```
##############################################################
# Copyright (C) 2000, 2001, 2002 Shinobu Yoshimura, University of
# Tokyo,
# the Japan Society for the Promotion of Science (JSPS) #
# All Rights Reserved #
##############################################################
# Example of config file for the script "advsolid" to run solver
# Run like
# % advsolid -conf advsolid.conf
##### set parallel mode #######################
# MODE=single
# MODE=parahddm
MODE=para
##### program name of mpirun ###################
MPIRUN=/usr/local/bin/mpirun
##### options for mpirun #######################
MPIOPTS="-np 12 -machinefile machine"
##### set if you want save log to file #######################
LOGFILE="run.log"
##### Options for AdvSolid #####################
PROGOPTS="-solver bdd -result --disp --estr-n -no-result --estr"
##### Data directory to be analyzed #####################
DATADIR=.
```

Fig. 13. Setup File for `advsolid`

5.5.6. Creation of Setup File for advshape

An example of setup file (here, its name is `advshape.conf`) is shown in *Fig. 14*. Here, the restriction parameter for mean compliance `SUBJ_RATIO` is set to 1.0.

```
IT_MAX           30
MAX_STR          0.3
SUBJ_RATIO       1.0
SUBJ_REL_ERROR   1.0E-4
OBJ_REL_ERROR    1.0E-4
```

Fig. 14. Setup File for `advshape`

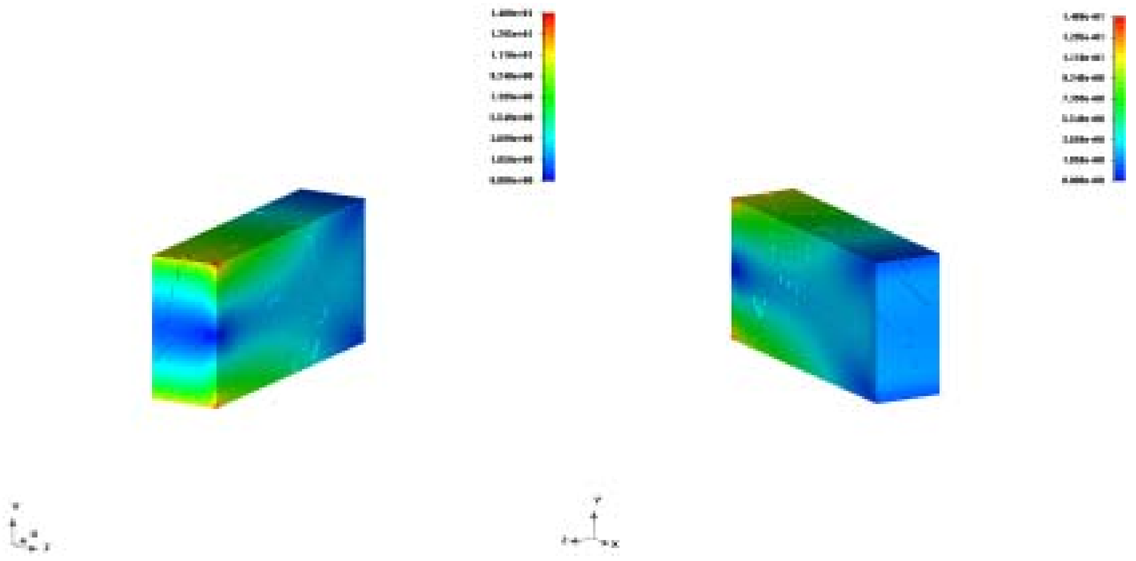
5.5.7. Execution of advshape

ADVENTURE_Shape can be executed by the command

```
%advshape advmetis.sh advsolid.conf advshape.conf
                      fem_model.adv rest_model.adv .
```

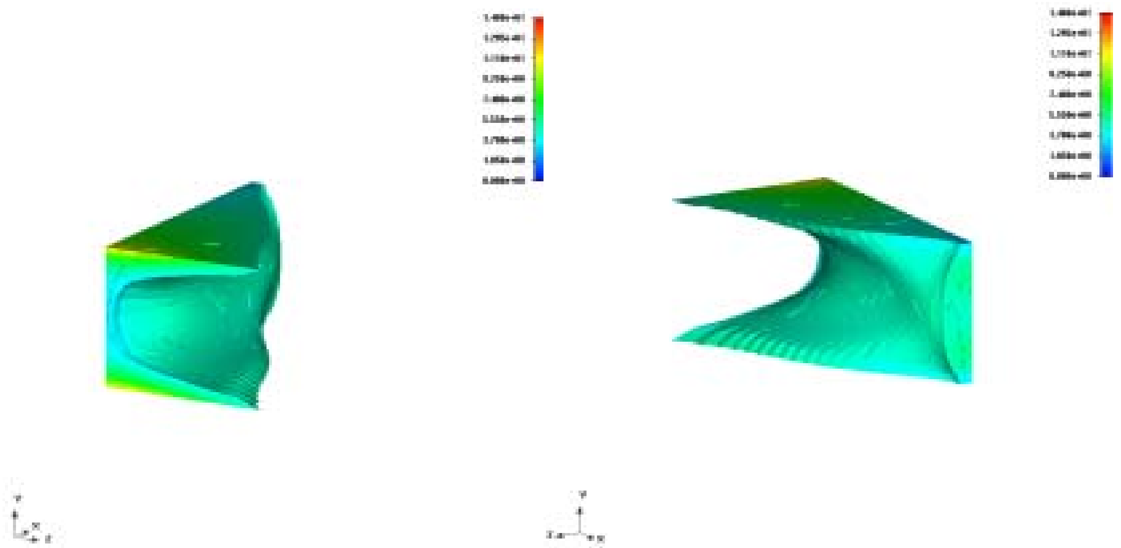
5.5.8. Output Results

The distributions of nodal equivalent stress are shown in *Fig. 15* and *Fig. 16* for the initial and optimized models, correspondingly. *Fig. 16* represents the model which volume was minimized with conditions of constant mean compliance. The volume was decreased by 22% due to optimization. In the initial model, bending stresses at the distance from normal axis were high and to reach the uniform shapes of stress distributions, the model became of I-shape (*Fig. 16a*). Moreover, to compensate the load that would reflect in shear stresses (not in bending stresses), the optimized model has a barrel shape. Finally, the uniform stress distributions are achieved in the optimized shape.



(a) *(b)*

Fig. 15. Distributions of Nodal Equivalent Stress of Initial Model



(a) *(b)*

Fig. 16. Distributions of Nodal Equivalent Stress of Optimized Model

6. Topology Optimization

6.1. Input Data

To execute the analysis with `advtopology`, the following four files should be prepared.

(1). The entire-type FEA model file (file extension: `.adv`).
The entire-type FEA model is saved in a binary file of ADVENTURE format using ADVENTURE_BCtool (refer to its User's Manual for operation instructions).

(2). The shell script file for ADVENTURE_Metis (file extension: `.sh`).
Setup methods for this shell script is described in *Section 5.1*.

(3). The setup file for `advsolid` (file extension: `.conf`).
Setup methods for this shell script is described in *Section 5.1*.

(4). The setup file for `advtopology` (file extension: `.conf`).
The setup file for `advtopology` contains parameters necessary for analysis. Contents of the setup file are shown in *Fig. 17*. "`MOVE_LIMIT`" is the design parameter (density ratio) for optimization process, which should be set in the range of [0<`MOVE_LIMIT`≤1] (usually, 0.3 is used). "`MASS`" is the restriction parameter for mass, which is set as a part of mass of initial model for topology optimization. The "`MASS`" should be set in the range of [0<`MASS`≤1]. The optimization calculations are done in the way to keep the accuracy set by the allowable relative error for mass restriction "`SUBJ_REL_ERROR`" constant. The judgment about the convergence of mean compliance is done by the value set by "`OBJ_REL_ERROR`".

`IT MAX`	25	← Maximum number of iterations for optimization
`MOVE_LIMIT`	0.3	← Design parameter for optimization
`MASS`	0.375	← Mass restriction parameter
`SUBJ_REL_ERROR`	1.0E-4	← Allowable relative error for mass restriction
`OBJ_REL_ERROR`	1.0E-4	← Judgment for relative convergence of mean compliance

Fig. 17. Example of Setup File for `advtopology`

6.2. Execution Method

`advtopology` can be executed using the command

`% advtopology [`*options*`]` *advmetis_sh advsolid_conf advtopology_conf fem_model data dir*

where, the options are:

[*options*]	the output options for `advtopology`
advmetis_sh	the name of ADVENTURE_Metis shell script file
advsolid_conf	the name of setup file for `advsolid` executable script
advtopology_conf	the name of setup file for `advtopology`
fem_model	the name of the entire-type FEA model data file
data_dir	the name of the top directory containing input and output data for ADVENTURE_Solid

6.3. Setup Options for Input and Output Data

The following options can be used with `advtopology` for execution. If you are going to use non-default names for files and directories, use different names for ADVENTURE_Metis, ADVENTURE_Solid, and `advtopology`.

- `-model-file` *file*
  The option should be used to specify the name of input FEA model file (if a non-default filename is considered). This option is same as used with ADVENTURE_Solid.

- `-model-dir` *dir*
  The option should be used to specify the name of directory with input FEA model files (if a non-default name is considered for directory). This option is same as used with ADVENTURE_Solid.

- `-result-file` *file*
  The option should be used to specify the name of output file (if a non-default filename is considered). This option is same as used with ADVENTURE_Solid.

- `-result-dir` *dir*
  The option should be used to specify the name of directory with output files (if a non-default name is considered for directory). This option is same as used with ADVENTURE_Solid.

- `-log` *logfile*
  The `advtopology` execution log displayed on the monitor's screen will be printed to the file *logfile*`.log`. The saved data are: mean compliance and

volume compliance at each optimization step. The file extension `.log` will be added to the filename specified by *logfile* automatically.

- `-output-topology` *file* [*sub-option*]
  The model with optimized topology will be saved to an entire-type file *file_*`F.adv`. The number of the final step of optimization calculations *F* and the extension `.adv` will be added to the filename automatically. The option "`-output-topology`" can be used together with sub-option "`--interval`". To view the model by ADVENTURE_Visual, the entire-type model file created at this step should be reprocessed by `topo_tool` (see *Section 6.4*) and decomposed using ADVENTURE_Metis. If the option "`-output-topology`" is not specified, the model will be saved after the last optimization step *F* with the default name `adv_topology_F.adv`.

- `--interval` *num*
  This sub-option is used together with the option "`-output-topology`". The model will be saved each *num* times into the file *file_*`N.adv`. Here, *N* is the step number of the analysis. The sub-option "`--interval`" must follow the option "`-output-topology`". If this sub-option is not specified, the topology will be saved at the step, where the calculations have been converged or the number of repetitions `IT_MAX` has been overcome without convergence.

6.4. Tool Program `topo_tool`

The program `topo_tool` deletes the elements which density ratio is less than the threshold value setup by user. The threshold value should be in range of [0<*value*<1]. User must specify the input file with the extension `.adv`. The results of topology optimization can be visualized using ADVENTURE_Visual after decomposing them with ADVENTURE_Metis. To execute topo_tool, use the following command.

`% topo_tool` *input_data output_data threshold_value*

input_data	the name of input data file (the file extension is `.adv`)
output_data	the name of output data file (the file extension is `.adv`)
threshold_value	the threshold value for deleting of elements

6.5. Output Data

There following data can be saved.

- Program execution log (file extension: `.log`).

  Information on the mass and mean compliance will be printed out at each optimization step. If this option is not specified, only standard output will be done.

- Topology, optimized under conditions of mean compliance restricted by constant mass (entire-type data file).

  The optimized topology will be printed out to an entire-type file with extension `.adv`. If the output option "`-option-topology`" is not specified, the results will be saved into the file `adv_topology_`*F*`.adv`, where *F* is the last step of optimization analysis. To visualize the results using ADVENTURE_Visual, the entire-type file with results should be reprocessed by `topo_tool` (*Section 6.4*) and decomposed by ADVENTURE_Metis.

6.6. Example of Analysis

The topology optimization analysis will be described using the model shown in *Fig. 18*. It consists of 1000 linear hexahedral elements and has a size of 40 x 25 x 1. The load of –1 is applied in Z direction at the middle point of the surface which is opposite to the fixed surface (*Fig. 18*).

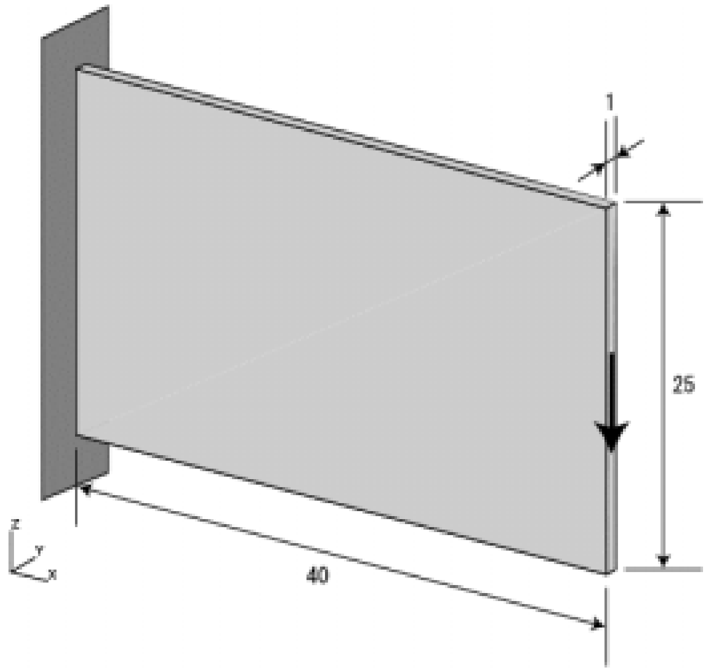


Fig. 18. Model for Analysis

6.6.1. Creation of Entire-type Analysis Model Data

The topology optimization can be done with the entire-type model `AdvShape-0.11b/sample_data/sample2.adv`. You can use the procedures described in previous sections.

6.6.2. Creation of Shell Script for ADVENTURE_Metis

Preparation of shell script for ADVENTURE_Metis is described in *Section 5.5.4*.

6.6.3. Creation of Setup File for Executable Script `advsolid`

Preparation of setup file for ADVENTURE_Solid is described in *Section 5.5.5*.

6.6.4. Creation of Setup File for advtopology

An example of the setup file for advtopology is shown in *Fig. 19*. The restriction for mass is 37.5%.

```
IT_MAX          25
MOVE_LT         0.30
MASS            0.375
SUBJ_REL_ERROR  1.0E-4
OBJ_REL_ERROR   1.0E-4
```

Fig. 19. Setup File for advtopology

6.6.5. Execution of advtopology

advtopology can be executed in the following way.

```
%advtopology advmetis.sh advsolid.conf advtopology.conf
                                        sample2.adv .
```

6.6.6. Execution of topo_tool

The tool program topo_tool can be executed in the following way.

```
%topo_tool adv_topology_22.adv optimum.adv 0.1
```

Here, the elements with density ratio less than 0.1 will be deleted from the model after 22nd step of optimization analysis. The final model will be saved in the file optimum.adv.

6.6.7. Execution of ADVENTURE_Metis

To view the model optimim.adv by ADVENTURE_Visual, it should be decomposed with ADVENTURE_Metis. Refer to the User's Manual for ADVENTURE_Metis.

6.6.8. Analysis Results

The results are shown in *Fig. 20* and *Fig. 21*.

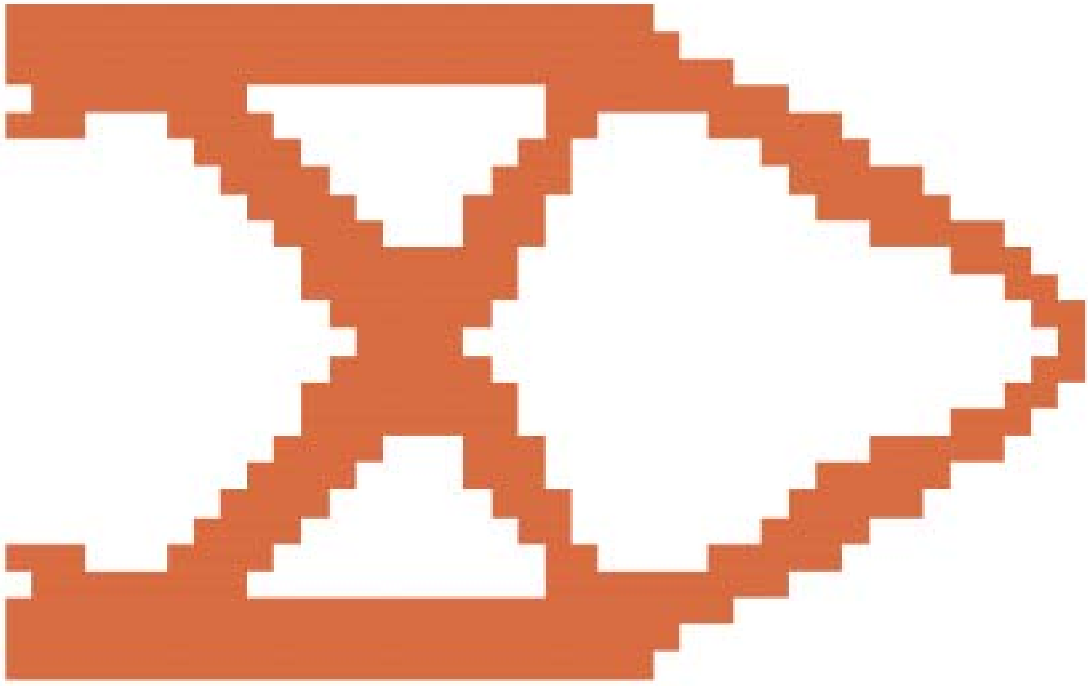


Fig. 20. Optimized Topology

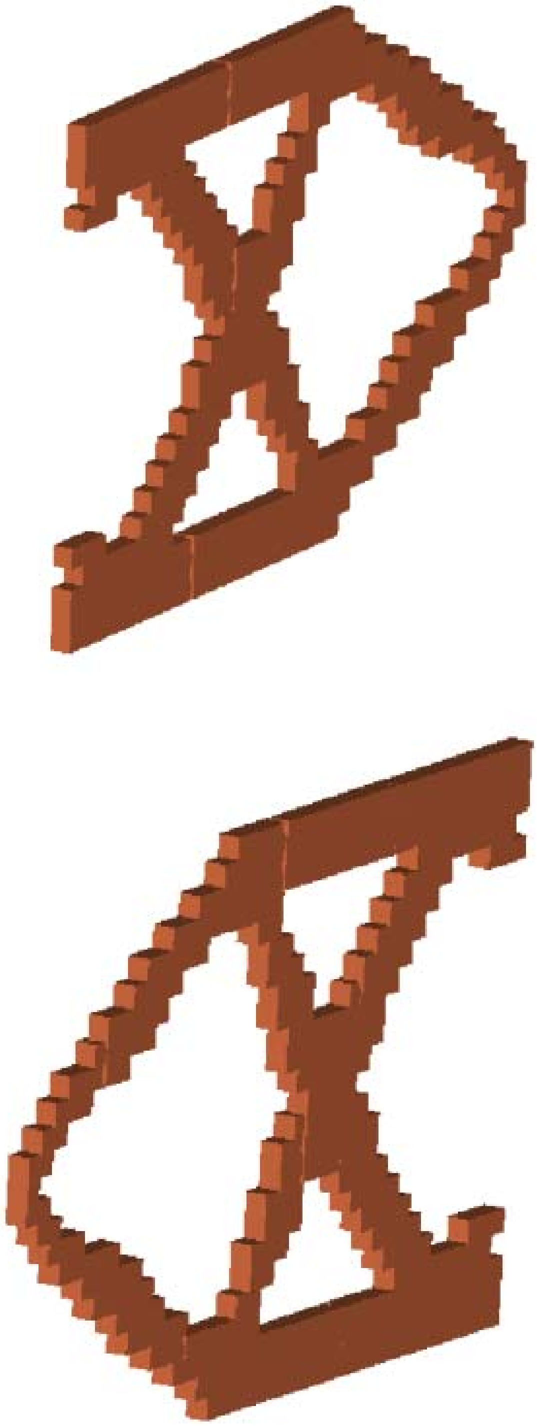


Fig. 21.   Optimized Topology (View angle is changed)

References

[1]. ADVENTURE Project Home Page: *http://adventure.q.t.u-tokyo.ac.jp/*

[2]. Azegami, H. (1994), A Solution to Domain Optimization Problems, Transactions of the Japan Society of Mechanical Engineers (JSME), Series A, Vol. 60, pp. 1479 - 1486 (in Japanese).

[3]. Azegami, H., Shimoda, M., Katamine, E. and Wu, Z. C. (1995), A Domain Optimization Technique for Elliptic Boundary Value Problems. In Hernadez, S., El-Sayed M. and Brebbia, C. A. editors, Computer Aided Optimization Design of Structures IV, Structual Optimization, Computational Mechanics Publications, Southampton, pp. 51-58.

[4]. Azegami, H. and WU, Z. C. (1996), Domain Optimization Analysis in Linear Elastic Problems (Approach using traction method), JSME International Journal, Ser. A, Vol. 39, pp. 272-278.

[5]. Fujii D., Suzuki K. and Ohtsubo H, Topology optimization of structures using the voxel finite element method, Internet Transactions of The Japan Society for Computational Engineering and Science (JSCES), Paper No.20000010 (2000) (in Japanese), *http://homer.shinshu-u.ac.jp/jsces/trans/trans2000/No20000010.pdf*