ADVENTURE_TriPatch

Automatic generation of triangular surface patches from IGES data

Version: 1.X

User's Manual

March 1, 2002

ADVENTURE Project

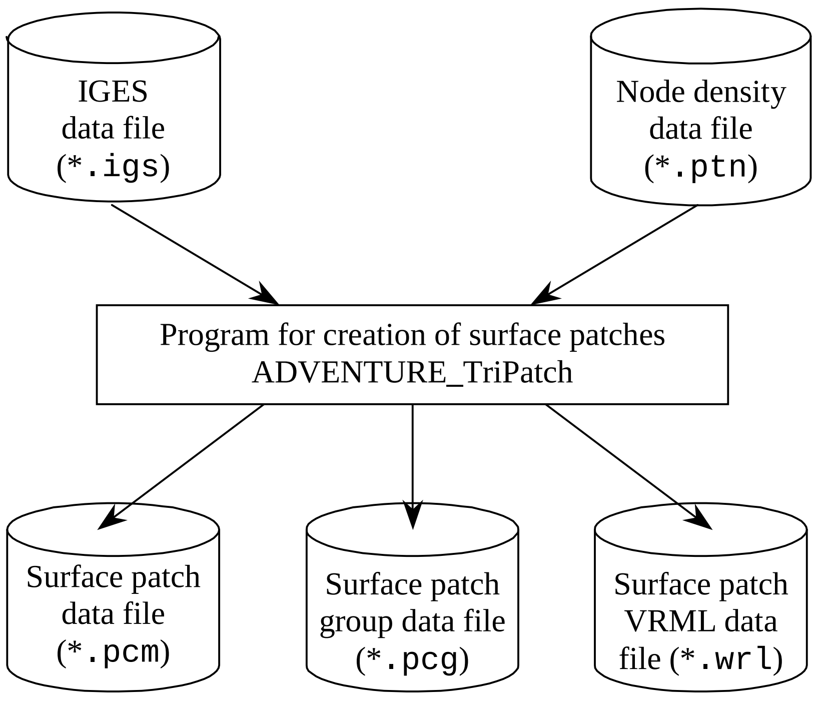
Contents

1. Outline

ADVENTURE_TriPatch is a program for automatic generation of triangular surface patches from the IGES solid model data [corresponding curved surface is the Non Uniform Rational B-Spline (NURBS) Surface]. Created data of triangular surface patches are stored in the following files.

(1). Surface patch data file (file extension: `.pcm`).
This file contains information about coordinates and connectivity of surface patches.
(2). Surface patch VRML file (file extension: `.wrl`).
This file contains the surface patch data in the VRML format.
(3). Surface patch group data file (file extension: `.pcg`).
This file contains the data on grouped surface patches.

A tool program for merging of surface patches (`mrpach` Ver. β) is provided in the ADVENTURE_TriPatch package (see Chapter 5 for details).



Data Processed by ADVENTURE_TriPatch

2. Operational Environment

ADVENTURE_TriPatch operates in the following environment.

- Operating System: UNIX or Linux
- Required Compiler: *g++* (Ver. 2.8.1 or higher)

3. Program Installation

3.1. Installation Method

Extract the module from the **tar+gz** form, and install the programs according to the contents of **`INSTALL`** file, located in the top directory.

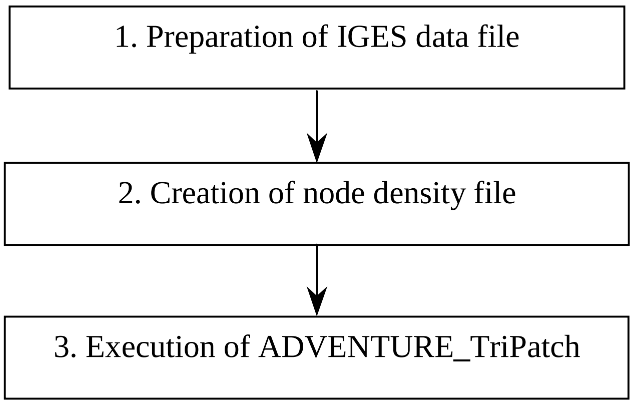
3.2. Structure of Directories

Reference README.eucJP in the top directory.

4. Program Handling

4.1. Flowchart of Program Operation

The execution flow of the program is shown below.


1. Preparation of IGES data file
2. Creation of node density file
3. Execution of ADVENTURE_TriPatch

1. Preparation of IGES data file.
   ・Refer t・Refer to Chapter *6.1 "IGES data file"* for entity limitations of the IGES format.
   ・Convert the file into the DOS→UNIX format when you make IGES file on Microsoft Windows environment.
   ・The file extension should be **`.igs`**.

2. Creation of node density file.
   ・The node density file can be created according to Chapter *6.4 "Node density data file".*
   ・The file extension should be **`.ptn`**.

3. Execution of ADVENTURE_TriPatch.
   The following command can be used to execute the program:

   `ADVENTURE_TriPatch` *[Name of IGES data file] [Name of node density data file]*

<u>*Notes*</u>

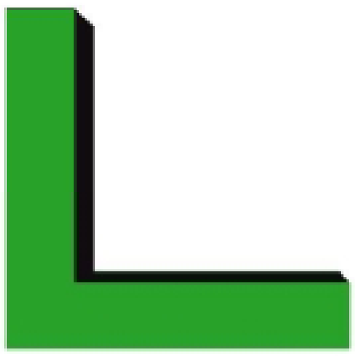
1. The filenames of IGES data and node density data should be specified without file extensions.
2. To create the surface patch data file using old format with file extension `*.pch` (not `*.pcm`), add the option `--out_pch_form` after the name of node density data file.

4.2. Example of Program Execution

An example of ADVENTURE_TriPatch execution is shown here using the sample data files *adventure_manual_data01.igs* and *adventure_manual_data01.ptn*, which are stored in the subdirectory *sample_data*.

```
% ADVENTURE_TriPatch adventure_manual_data01 adventure_manual_data01
```

The shape of the sample model is shown below.



4.3. Results of Program Execution

The following two items will be explained in this chapter:

(1). Execution log (the messages displayed on the screen during program execution).

(2). Confirmation of surface patch.
A method to confirm the surface patch created after execution of the program.

4.3.1. Execution Log

The following execution log was created during execution using the sample data *adventure_manual_data01.igs* and *adventure_manual_data01.ptn*.

＜Execution log＞
Start message of surface patch making program.

```
-------------------------------------------
---   triangular patch generator         ---
---  (IGES ver5.3 --> triangular patch)   ---
-------------------------------------------
```

The number of the entities contained in IGES data is outputted.

```
----- entity list  of input  -----
  entity    n    entity_name(*:skip)
   126     54       Rational B-Spline Curve
   128      8       Rational B-Spline Surface
   186      1       Manifold Solid B-Rep Object
   314      2      *Color Definition
   502      1       VertexList
   504      1       EdgeList
   508      8       Loop
   510      8       Face
   514      1       Shell
```

Here, the column **`entity`** shows the number of entity, the column **`n`** shows the number of entities, and the column **`entity_name`** shows the names of entities.

If there is an entity, which is not supported, the symbol "*" will be displayed in the column `entity_name`. In the above-mentioned example, the entity related to color is displayed with the symbol "*". Since the information about colors is not needed to create the surface patch, this sample can be executed without any troubles.

```
---  report of CAD data required for mesh creation ---
data type --> solid
minimum edge length --> 9.746596e+00( edge number = 1)
```

The massage `data type--> solid` shows that the input data are of “solid” type. If the input data for solid are not of the “solid” type, the surface patch will not be created.
The `minimum edge length` shows the shortest length of edge in the IGES data.
If there are big differences in the base node intervals of the node density data, distorted triangles can be occasionally made. In this case, the values will be printed out for reference.

```
base edge length =  2.500000e+00
pattern--->Line
densityStrength = 2.500000e+00
densityRange    = 2.000000e+01
position1     = ( -6.152538e+01, 8.132398e+00, 0.000000e+00 )
position2     = ( -6.152538e+01, 8.132398e+00, 1.000000e+01 )
```

Contents of an input file with node density data are shown .

```
created boundary vertex = 311
faceID = 0 / 8  generated patch = 39
faceID = 1 / 8  generated patch = 516
faceID = 2 / 8  generated patch = 1117
faceID = 3 / 8  generated patch = 42
faceID = 4 / 8  generated patch = 538
faceID = 5 / 8  generated patch = 553
faceID = 6 / 8  generated patch = 1129
faceID = 7 / 8  generated patch = 486
```

In this case, the CAD model has eight surfaces, which numbers are shown depending on the surface patch. The denominator of `0/8` shows the total number of the surfaces, and the numerator shows the number of the current surface (starts from `0`).

```
---  patch generator normally ended ---
created vertex = 2212
created triangular patch = 4420
```

The message **`--- patch generator normally ended ---`** shows that the surface patch creation is ended normally.
The message **`--- err patch generator abnormally ended---`** is displayed when the surface patch creation is failed.
The message **`created vertex`** shows the number of the vertices.
The message **`created triangular patch`** shows the number of triangular patches.

```
-----------------
---   check   ---
------------------

~~~~Omitted~~~~~~~

------ TOPOLOGY  CHECK ------

~~~~Omitted~~~~~~~

output mode is solid
```

The message `output mode is solid` means that the topology check is finished normally (no errors).   In case of errors, the following message is printed out:
`--- err  out put mode is un solid---.`

The program message of starting the corrections of normal vectors is shown below. Corrections are made in a way to direct the normal vectors of surface patches into the body.

```
---------------------------------------------------
--- repair normal vector of triangular patch    ---
---------------------------------------------------

~~~~~~Omitted~~~~~
```

4.3.2. Confirmation of Surface Patch

After execution of ADVENTURE_TriPatch, the surface patches are saved in the VRML-format file. The data can be displayed using any VRML browser (VRML format Ver. 1.0).

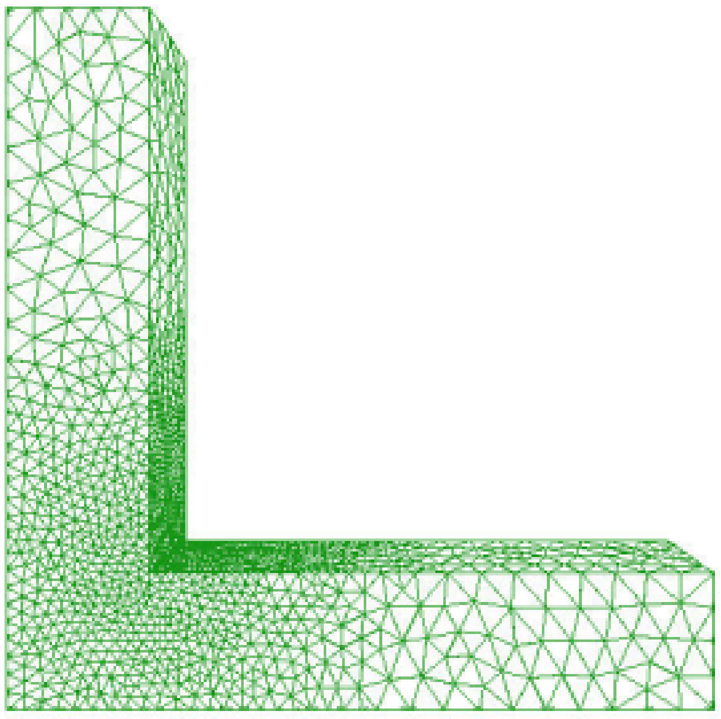


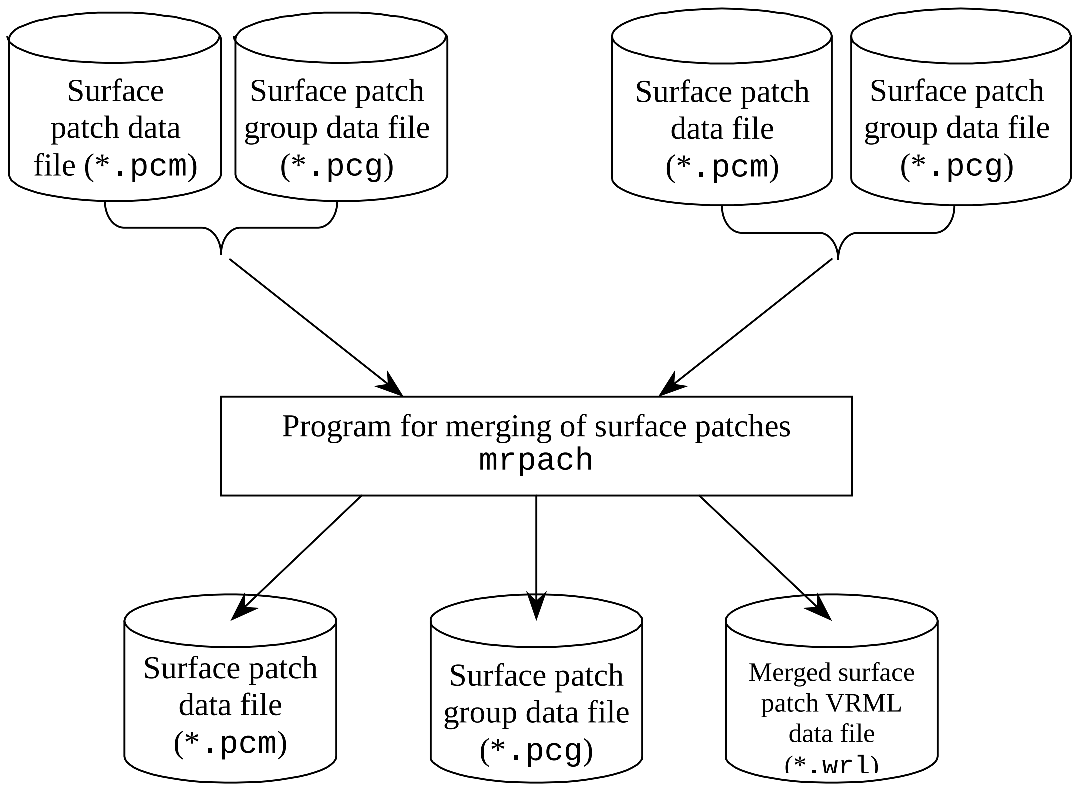
Fig. 4-1. Sample Data Displayed by VRML Browser

5. Other Tools

A tool program for merging of surface patches (`mrpach` Ver. β) is provided with the ADVENTURE_TriPatch module. This chapter provides brief information about handling of the program.

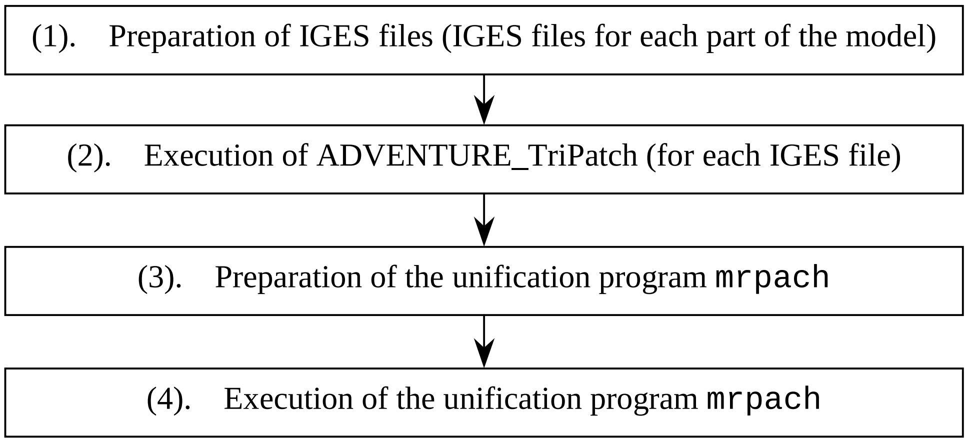
5.1. Program for Merging of Surface Patch Data

The program `mrpach` is used to merge the surface patch data. `mrpach` reprocess 2 sets of surface patch data files and surface patch group data files and create 1 set of the surface patch data file and the surface patch group data file for multi-material finite element analysis model.


Surface patch data file (*.pcm)
Surface patch group data file (*.pcg)
Surface patch data file (*.pcm)
Surface patch group data file (*.pcg)
Program for merging of surface patches mrpach
Surface patch data file (*.pcm)
Surface patch group data file (*.pcg)
Merged surface patch VRML data file (*.wrl)

5.1.1. Program Operational Flowchart

The flow of program execution is shown below.


(1). Preparation of IGES files (IGES files for each part of the model)
(2). Execution of ADVENTURE_TriPatch (for each IGES file)
(3). Preparation of the unification program mrpach
(4). Execution of the unification program mrpach

(1). Preparation of IGES files. The IGES files, which can be used for multi-material model, should be created taken into account the precautions provided in *Paragraph 5.1.2*. One IGES file should contain information on one part (a single entire body) of the multi-material model.

(2). Execution of ADVENTURE_TriPatch. ADVENTURE_TriPatch reprocesses a number of IGES files created at the first step (as it was mentioned in *Chapter 4*). Each set of files is reprocessed separately using utmost same node density. The following files are created:

- Surface patch data file (file extension: `*.pcm`)
- Surface patch group data file (file extension: `*.pcg`)

(3). Preparation of `mrpach`. Refer to *Paragraph 5.1.2* for precautions about execution of `mrpach`.

(4). Execution of the merge program `mrpach`. The execution command is

```
mrpach  Domain_A.pcm  Domain_A.pcg  Domain_B.pcm  Domain_B.pcg  -o
                          Result_file.pcm –g Result_file.pcg –v Result_file.wrl
```

where *Domain_A.pcm* and *Domain_A.pcg* are the filenames of the 1st dataset,
Domain_B.pcm and *Domain_B.pcg* are the filenames of the 2nd dataset,
Result_file.pcm and *Result_file.pcg* are the filenames of the merged dataset,
Result_file.wrl is the filename of results in VRML format, which can be used for confirmation of the data,
-*o* is used to specify the name of surface patch data file,
-*g* is used to specify the name of the surface group data file,
-v is used to specify the name of output file in VRML format.

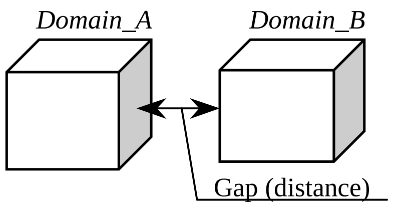
In some cases, if there is a small gap between *Domain_A* and *Domain_B*, the merging process ends up with errors. In this case, try to use the option "–*d* value" to set a distance (a gap) between domains. Default value is 1.0e-5.

5.1.2. Cautions

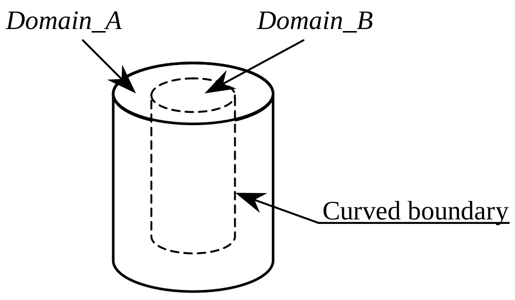
<u>*1. Cautions for preparation of CAD data*</u>

Before you execute the program for surface patch merging, it is recommended to prepare the CAD data taken into account the cautions listed below.

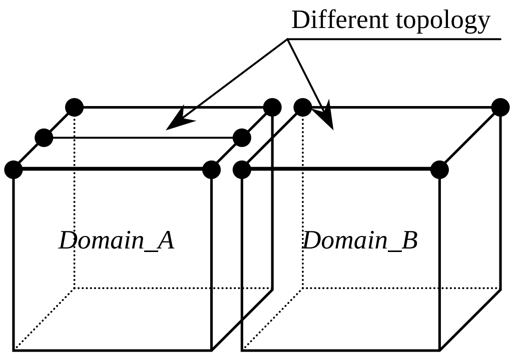
1). Make the gap (distance) between *Domain_A* and *Domain_B* as small as possible (Case 1). The best is a model without gaps.
2). Avoid to have curved boundary surfaces between *Domain_A* and *Domain_B* (Case 2).
3). Make the boundary surface between *Domain_A* and *Domain_B* of the same shape and topology (Case 3).
4). Prevent intersection between *Domain_A* and *Domain_B* (Case 4).



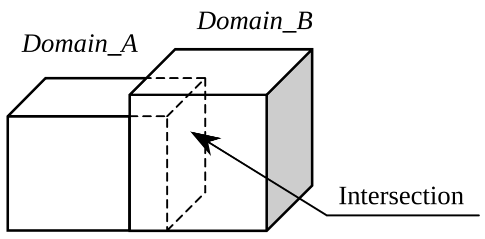
Case 1



Case 2



Case 3



Case 4

2. Cautions for merging of the surface patch data

Before you execute the program for surface patch merging, check your data taken into account the cautions listed below.

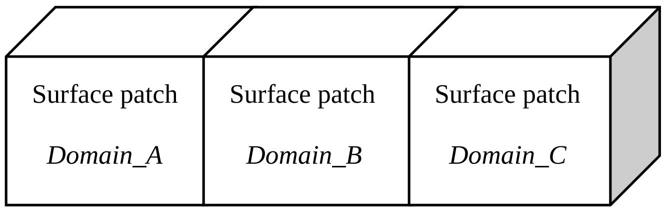
1). Set the same (or as close as possible) node densities when making the surface patches of *Domain_A* and *Domain_B*.

2). Note that it is possible to merge only the surface patches of neighboring domains. If you need to merge surface patches of several domains, merge the neighboring domains sequentially by pairs. For example, if you have surface patches of *Domain_A*, *Domain_B*, and *Domain_C*:

a). Execute the program `mrpach` to merge the surface patches of *Domain_A* and *Domain_B*.

b). Execute the program `mrpach` to merge the surface patches of *Domain_C* with the merged surface patches of *Domain_A*+*Domain_B*.

As it is shown in the following figure, the surface patches of *Domain_C* cannot be directly merged with the surface patches of *Domain_A*.



5.1.3. Example of Program Execution

The merging of surface patches using `mrpach` will be demonstrated on the sample data supplied with ADVENTURE_TriPatch. The files are located in the subdirectory *sample_data*.

Step 1. Here, 2 sets of files are used for ADVENTURE_TriPatch input.

1st set: *adv_mat_sample01.igs* and *adv_mat_sample01.ptn*
2nd set: *adv_mat_sample02.igs* and *adv_mat_sample02.ptn*

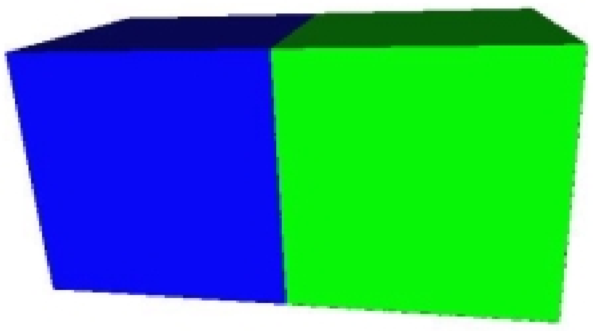
Reprocessing of the data by ADVENTURE_TriPatch resulted in creation of 2 datasets:

1st set: *mat_in01.pcm* and *mat_in01.pcg*
2nd set: *mat_in02.pcm* and *mat_in02.pcg*

Step 2. The obtained results are reprocessed using the program `mrpach`.

```
% mrpach mat_in01.pcm mat_in01.pcg mat_in02.pcm mat_in02.pcg –o
                                merge.pcm –g merge.pcg –v merge.wrl
```

The results are saved as *merge.pcm, merge.pcg* and *merge.wrl*. The model can be confirmed using the file *merge.wrl*. The resultant model is shown in the following figure.



5.1.4. Confirmation of Surface Patches

After execution of `mrpach`, the merged surface patches can be confirmed using any available VRML file viewer (VRML format Ver. 1.0).

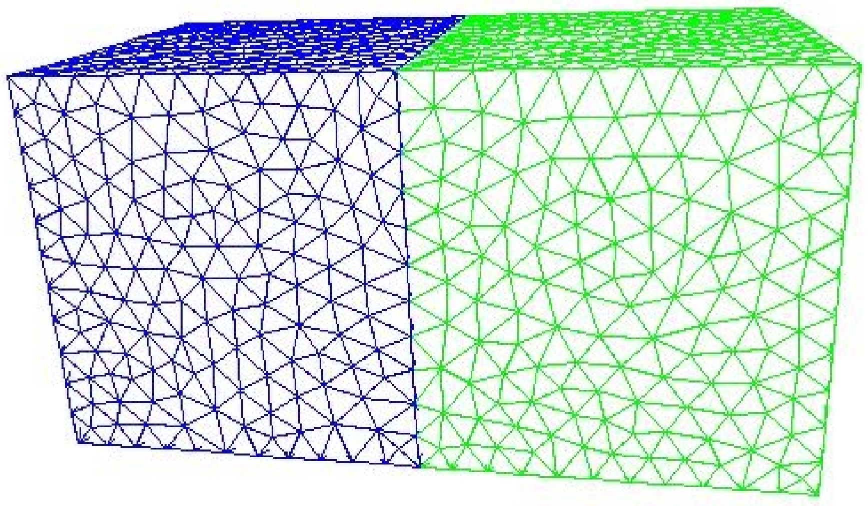


Fig. 5-1. Sample Data Displayed by VRML File Viewer

6. File Specifications

The files used by ADVENTURE_TriPatch are shown in the following table.

File Name	*File Contents*
IGES data file	A file of IGES format made by a CAD program.
Node density data file	A file which is used to control the density of triangular patches
Surface patch data file	A file which contains the data on nodal coordinates and triangular patches
Surface patch VRML file	A file which contains the surface patch data in VRML format (VRML format Ver1.0)

6.1. IGES Data File

1). IGES is based on Ver. 5.3 specifications. (ASCII format).

2). This program is matched with a solid model of NURBS (Non Uniform Rational B-Spline Surface). The entity number 186 is shown if the IGES data are made from a solid model. The program generates an error if the entity number 186 does not exist.

3). This program can use IGES files created by the following CAD programs.
   a). *I-DEAS MasterSerise 8*
   b). *MicroCADAM V4R2*

4). Possible entities are presented in the following table

No.	*Entity number*	*Entity name*
1	100	Circular arc
2	110	Line
3	124	Conversion matrix
4	126	Rationalization B-spline curve
5	128	Rationalization B-spline surface
6	186	Manifold solid B-Rep object
7	502	Vertex
8	504	Edge
9	508	Loop
10	510	Surface
11	514	Shell

6.2. Surface Patch Data File

The format of surface patch data is shown below.

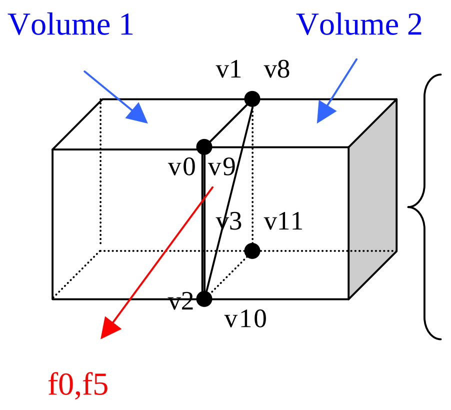
- The connectivity of each domain (volume) is assumed in the clockwise direction if we look at the model from the outside.
- The file extension is `.pch`.
- The method to display the volume's boundaries is discussed in *Section 6.1* of the current Manual.

```
NV  0  NR    ←The number of vertices, a reserved number (input 0), the number of domains
x[0]  y[0]   z[0]     ←The coordinates of vertices
x[1]  y[1]   z[1]
x[2]  y[2]   z[2]
~~~~~Omitted~~~~~~
x[NV-1]  y[NV-1] z[NV-1]
(The following block is repeated NR times)
NP0  0   0       ←The number of surface patches, a dummy number (input 0), a reserved number (input 0)
  e1[0]  e2[0]  e3[0]  ←The patch connectivity data
  e1[1]  e2[1]  e3[1]
  e1[2]  e2[2]  e3[2]
~~~~~Omitted~~~~~
  e1[NP0 - 1]  e2[NP0 - 1]  e3[NP0 - 1]
```

An example of surface patch data is given below.

```
*pcm ver. 1.0      ←The version of file format
347  0   2        ←The number of vertices,    a reserved number (1),    the number of domains
0.0  0.0  0.0     ←The X, Y, and Z coordinates of 0 vertex
1.0  9.0  88.0    ←The X, Y, and Z coordinates of 1st vertex
~~~~~Omitted~~~~~~~~
84.05 34.6  98.1    ←The X, Y, and Z coordinates of 346th vertex
5786  0  0  ←The number of patches of 0 volume, a reserved number (2), a reserved number (3)
153   55    412     ←The list of nodes which compose the surface patch No. 0
567   45    34      ←The list of nodes which compose the surface patch No. 1
~~~~~Omitted~~~~~~~~
567   45    34         ←The list of nodes which compose the surface patch No. 5785
456  0  0  ←The number of patches of 1st volume, a reserved number (2), a reserved number (3)
99   42    765        ←The list of nodes which compose the surface patch No. 0
19   32    67         ←The list of nodes which compose the surface patch No. 1
~~~~~Omitted~~~~~~~~
99   23    21         ←The list of nodes which compose the surface patch No. 455
```

The reserved numbers (1, 2, and 3) should be specified as 0.



1). The boundaries between volumes have double vertices.
- The vertices of Volume 1 are v0~v3.
- The vertices of Volume 2 are v8~v11.

2). The boundaries between volumes have double surfaces (triangular patches)
- f0 of Volume 1 consists of v0, v1, and v2.
- f5 of Volume 2 consists of v8, v9, and v10.

Fig. 6-1. Representation of volume boundaries in surface patch data file

6.3. Surface Patch Group Data File

```
#mainVertexInfo
mainVertexN 299       ←The number of Main nodes (see Note 1)
0                     ←Main node No. 0
1                     ← Main node No. 1
~~~~~Omitted~~~~~~~~
10
27
~~~~~Omitted~~~~~~~~
2161
2162                  ← Main node No. 299-1
#edgeGroupInfo
edgeGroupN 305        ←The number of edge groups
edgeGroup 2          ←The number of nodes which form the edge group No. 0
0                    ←The node No. 0 of the edge group No. 0
1                    ←The node No. 1 of the edge group No. 0
edgeGroup 2          ←The number of nodes which form the edge group No. 1
0                    ←The node No. 0  of the edge group No. 1
10                   ←The node No. 1 of the edge group No. 1
~~~~~Omitted~~~~~~~~
edgeGroup 2      ←The number of nodes which form the edge group No. 305-1
9                    ←The node No. 0 of the edge group No. 305-1
30                   ←The node No. 1 of the edge group No. 305-1
#faceGroupInfo
faceGroupN 8        ←The number of surface groups
faceGroup 470      ←The number of patches which form the surface group No 0
0                   ←The patch No. 0 of the group No. 0
1                   ←The patch No. 1 of the group No. 0
~~~~~Omitted~~~~~~~~
469                 ←The patch No. 470-1 of the group No. 0
~~~~~Omitted~~~~~~~~
faceGroup 39    ←The number of patches which form the surface group No. 8-1
4283               ←The patch No. 0 of the group No. 8-1
4284               ←The patch No. 1 of the group No. 8-1
~~~~~Omitted~~~~~~~~
4321               ←The patch No. 39-1 of the group No. 8-1
```

Note 1. The Main nodes are the nodes, which represent (specify) a shape of the model.

6.4. Node Density Data File

(1). Outline of node density data

The node density data are subdivided into the base node interval and the local node density.

a). Base node interval.
The length of surface patch ridgeline is specified and the surface patch is created following this length.
b). Local node density.
The local node density is used to create a detailed surface patch of an arbitrary part of the model. The local node density has two patterns:
1). "Inverse proportion to the distance from the point", and
2). "Inverse proportion to the distance from the segment". To set the local node density, specify the density intensity parameter, the location (x, y, and z coordinates), and the applicable boundaries.

(2). Example of node density application

Applications of the node density parameter are demonstrated on examples. The patterns, the application results, and the relationships between the density and the distance are shown in *Figs. 6-2 ~ 6-4*. Three patterns can be seen: one pattern of "Inverse proportion to the distance from the point" and two patterns of "Inverse proportion to the distance from the segment".

- The horizontal axis is corresponded to the distance r or r_1~r_4, and the vertical axis shows the density d.
- The distance from a specified point is shown if the option is set to "Inverse proportion to the distance from the point" and the distance from a specified segment is shown if the option is set to "Inverse proportion to the distance from the segment".
- For the cases "Inverse proportion to the distance from the point" and "Inverse proportion to the distance from the segment", the distance and the density are in inverse proportion. However, for the pattern of "Inverse proportion to the distance from the line", the density can be controlled by the distance from the segment.

<Example>
The example shown in *Fig. 6-2* is corresponded to "Inverse proportion to the distance from the point". The density decreases as the distance from the point increases (the node interval grows with moving away from the point).

(Notes)
The data used for "*Example of the node density application*" are stored in the subdirectory `sample_data`.
(*adventure_manual_data02.igs* and *adventure_manual_data02.ptn*)

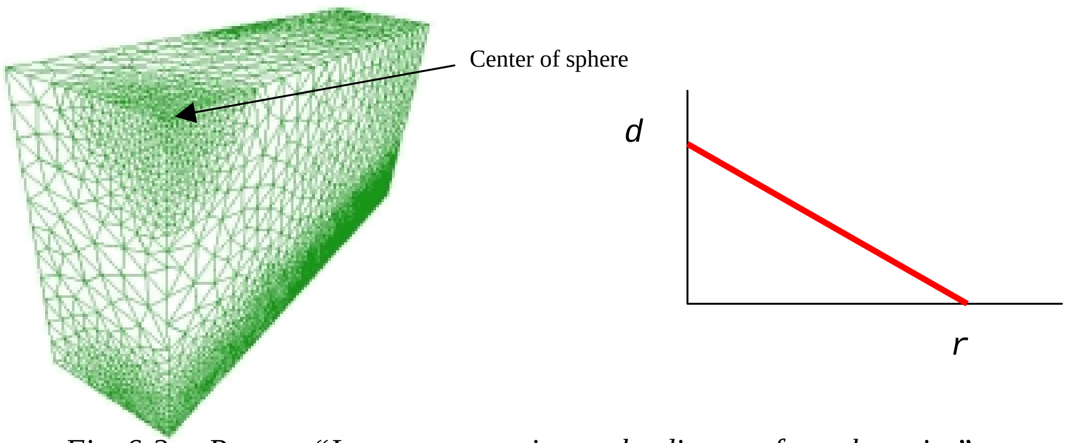


Fig. 6-2. Pattern "Inverse proportion to the distance from the point" (NodalPatternOnPoint is used)

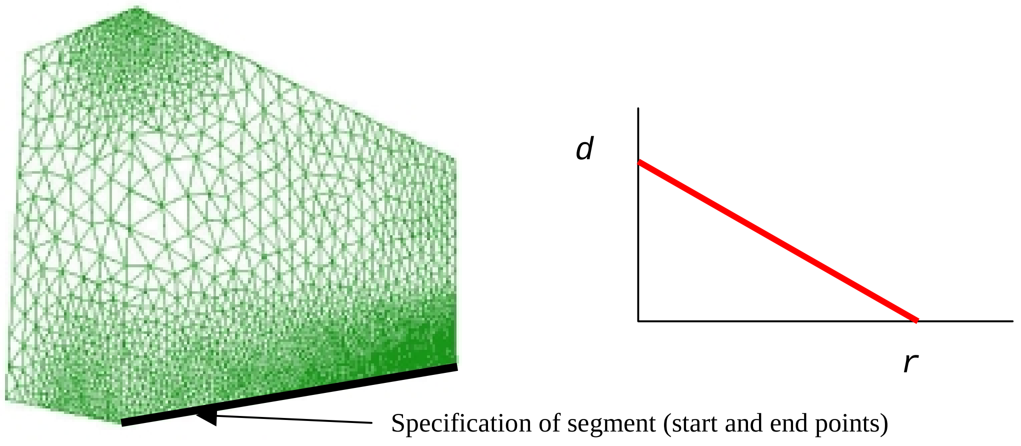


Fig. 6-3. Pattern "Inverse proportion to the distance from the segment" (NodalPatternOnLine is used)

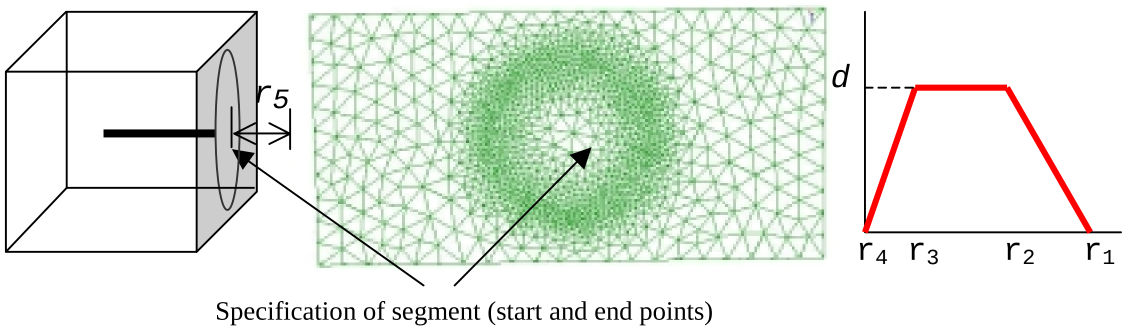


Fig. 6-4. Pattern "Inverse proportion to the distance from the segment" (NodalPatternOnCylinder is used)

(3). Format of the node density data file

The format of the node density data is shown below.

```
BaseDistance                                  ← Base node interval
  1.00E+00

NodalPatternOnPoint                   ← The pattern "Inverse proportion to the distance from the point"
  2.00E+01     4.7                            ← Range from center of sphere (r),   density intension
  1.00000E+01  0.00000E+00  0.00000E+00       ← Coordinates of the center of sphere

NodalPatternOnLine                 ← The pattern "Inverse proportion to the distance from the segment"
  2.00E+01     4.7                          ← Range from segment (r),   density intension
  1.00000E+01  0.00000E+00  0.00000E+00        ← Coordinates of the beginning of segment
  1.00000E+01  2.00000E+00  0.00000E+00        ← Coordinates of the end of segment

NodalPatternOnCylinder               ← The pattern "Inverse proportion to the distance from the segment"
                                                (the range of the node density can be specified)
12.0    10.0    9.0    8.0    3.0    1.5    ← Range 1 to Range 5 (r1~r5),    density intension
347.1    0.0    100.0                        ← Coordinates of the beginning of segment
406.1    0.0    100.0                        ← Coordinates of the end of segment
```

- The parameter `BaseDistance` is essential to execute the program.
- Other items (`NodalPatternOnPoint`, `NodalPatternOnLine`, and `NodalPatternOnCylinder`) are used to create more detailed mesh for an arbitrary part of the FEA model.