ADVENTURE_TetMesh

Automatic generation of tetrahedral mesh from triangular surface patches

Version: β - 0.91

User Manual

October 05, 2005

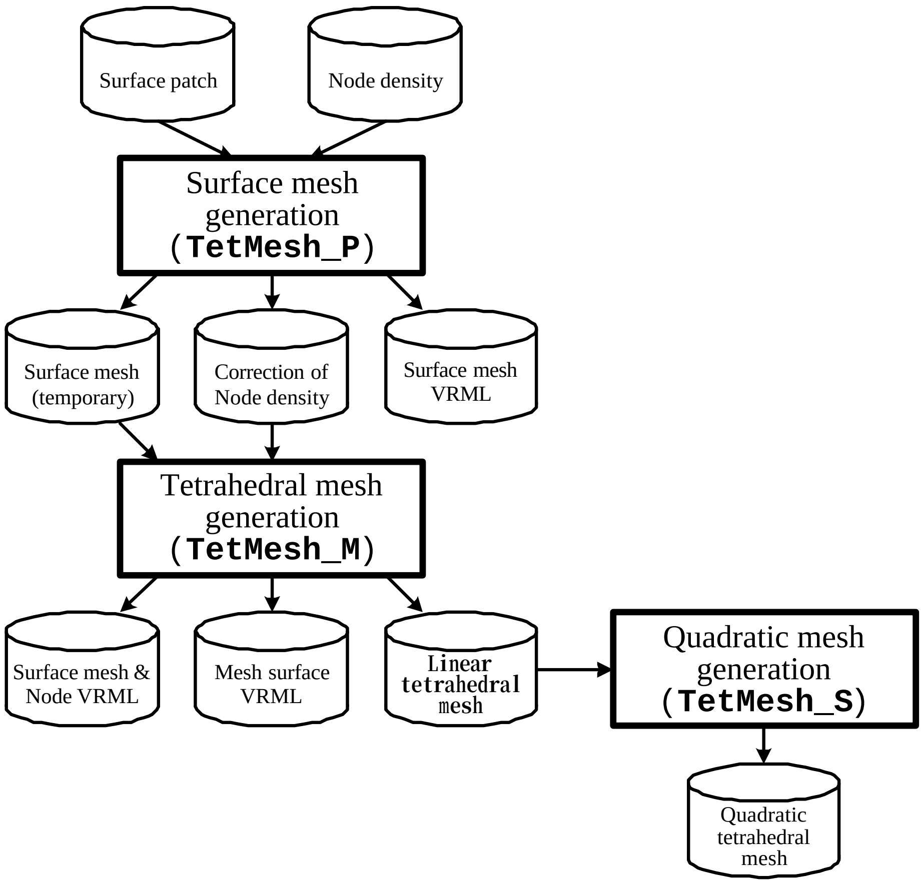
ADVENTURE Project

Contents

1. Outline

This program generates tetrahedral element mesh system from input triangular surface patches using the Delaunay triangulation method. The program consists of three modules: **TetMesh_P** , **TetMesh_M** and **TetMesh_S**. The module **TetMesh_P** smoothes the surface patches by using Pliant Delaunay re-triangulation method. The module **TetMesh_M** generates the tetrahedral mesh system by the Delaunay triangulation method. The module **TetMesh_S** generates quadratic tetrahedral mesh system from linear tetrahedral mesh system. The program also contains tetrahedral mesh evaluation tool **TetMesh_E**. The information about the generated meshes is contained in the following files.

(1) Tetrahedral mesh data file（extension：**.msh**）
Node coordinates and element connectivity of the tetrahedral mesh
(2) Surface VRML file（Extension：**.wrl**）
Data set of mesh surface converted into VRML format (two sets)



ADVENTURE_TetMesh

2. Operational Environment

The program operation is confirmed in the following environments.

(1) Operating System

UNIX, Linux

(2) Compilers

`TetMesh_P` : *Fortran90* (Operation is confirmed with *DIGITAL Fortran 90 V5.2-705* , *PGI Fortran 90 V3.2-3,* *g95 (after Sep 25 2005)*)

`TetMesh_M, TepMesh_S, TetMesh_E` : *C++* (Operation is confirmed with *Compaq C++ Ver. 6.2-024,* *g++ Ver. 2.9x, 3.x, 4.0.1*)

3. Program Installation

3.1. Installation Procedure

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

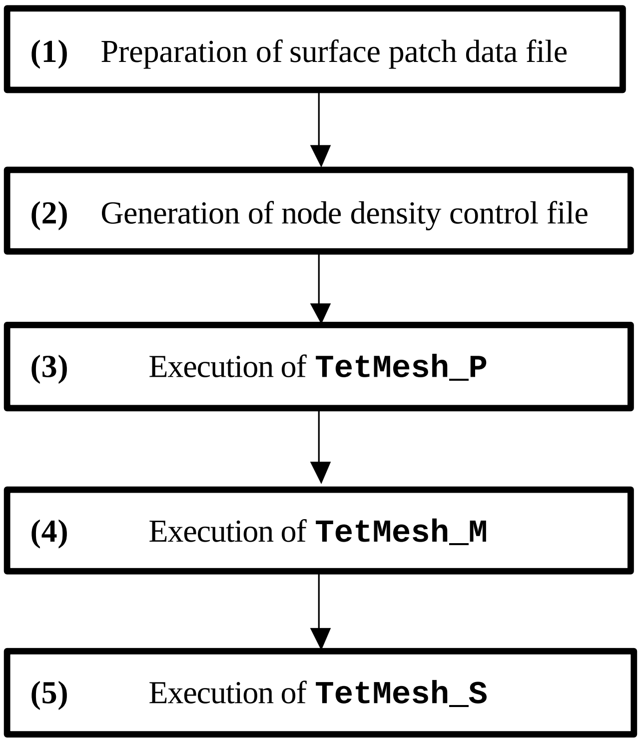
3.2. Structure of Directories

The information about files and directories structure is given in **`README`** file located in the top directory.

4. Program Handling and Operation

4.1. Program Operation Flowchart

The execution flow of the program is shown below.



(1) Preparation of the surface patch data file

- The surface patch data file should be prepared according to the format shown in Chapter *6.1 "Surface Patch Data File".*
- The surface patch data file is compatible with an output of ADVENTURE_TriPatch module of the ADVENTURE System.
- The file extension should be **.pcm**.
- If it is single domain, the old format (**.pch**) can also be inputted.

(2) Generation of node density control file

- Prepare the node density control file according to the format shown in Chapter *6.2 "Node Density Control File".*
- If the use of ADVENTURE_TriPatch module output is considered, no changes are necessary in the nodal density control file after preparation of the patch.
- The file extension should be **.ptn**.

(3) Execution of **`TetMesh_P`**

This program can be executed by the following commands:

`Advtmesh9p` ***Surface_patch_data_file_name*** **`-d`**

Input the surface patch data file name without file extension. If the node density control file is used, the command option **`-d`** should be added. The command options are explained below. The surface mesh data file (the file extension is **`.pcc`**) and the corrected node density control file (the extension is **`.ptn`**) will be generated as an output after execution of **`TetMesh_P`**, and the character **`“c”`** will be added to each original file name. If needed, the surface mesh can be converted into the VRML format (VRML format Ver 1.0) by adding the command option **`-p`**. The extension **`_c.wrl`** will be added to the specified surface mesh data file name. The contents of the converted file can be displayed using a VRML browser.

(4) Execution of **`TetMesh_M`**

This program can be executed by the following commands:

`Advtmesh9m` ***Surface_mesh_data_file_name***

Input the surface mesh data file name without file extension. The command options are explained below. The linear tetrahedral mesh output file with the extension **`.msh`** will be generated after execution of **`TetMesh_M`**. If needed, the mesh surface can be converted into the VRML format (VRML format Ver 1.0) by adding the command option **`-p`**. The extensions **`_e.wrl`** and **`_n.wrl`** will be added to the specified surface mesh data file name. Contents of the converted file can be displayed using a VRML browser. The created tetrahedral mesh output file can be used as input data for the ADVENTURE_BCtool module of the ADVENTURE System.

(5) Execution of **`TetMesh_S`**

This program can be executed by the following commands:

`Advtmesh9s` ***Linear_tetrahedral_mesh_data_file_name***

Input the linear tetrahedral mesh data file name without file extension. The command options are explained below. The quadratic tetrahedral mesh output file with the extension **`.msh`** will be generated after execution of **`TetMesh_S`**, and the character **`“s”`** will be added to original file name. The created quadratic tetrahedral mesh output file can be used as input data for the ADVENTURE_BCtool module of the ADVENTURE System.

4.2. Program Execution Sample (single domain)

4.2.1. Program Execution

Sample data files are located in the subdirectory **sample_data**. An example of program execution using the files **adventure_manual_data01.pcm** and **adventure_manual_data01.ptn** is shown here.

(1) An execution of **TetMesh_P** can be started by the following command:

```
% advtmesh9p   adventure_manual_data01   -d   -p
```

The program will input two files:**adventure_manual_data01.pcm**, and **adventure_manual_data01.ptn**.
As a result, three files will be created:**adventure_manual_data01c.pcc**, **adventure_manual_data01c.ptn**, and **adventure_manual_data01_c.wrl**.

(2) An execution of **TetMesh_M** can be started by the following command:

```
% advtmesh9m   adventure_manual_data01c   -p
```

The program will input two files:**adventure_manual_data01c.pcc**, and **adventure_manual_data01c.ptn**.
As a result, three files will be created:**adventure_manual_data01c.msh,** **adventure_manual_data01c_n.wrl**, and **adventure_manual_data01c_e.wrl**.

(3) An execution of **TetMesh_S** can be started by the following command:

```
% advtmesh9s   adventure_manual_data01c
```

The program will input one file:**adventure_manual_data01c.msh**.
As a result, one file will be created:**adventure_manual_data01cs.msh**.

4.2.2. Execution Log

An output message log file will be generated after the program execution. Explanations about the message contents for the above-mentioned sample files are presented in *Appendix A, Appendix B* and *Appendix C*.

4.2.3. Sample Results

The VRML format files (VRML format Ver 1.0) generated after program execution can be displayed using a VRML browser.

(1) Input patch

By executing the following commands, the input original surface patch can be converted into VRML format:

```
% advtmesh9p   adventure_manual_data01   -cr   -p
```

The file named **`adventure_manual_data01_c.wrl`** will be created. The input patch file can be converted into the VRML format without correction of the patch by adding the option **`-cr`** to the execution command. Figure 4.2.3-1 shows an example of the VRML output file displayed by a VRML browser.

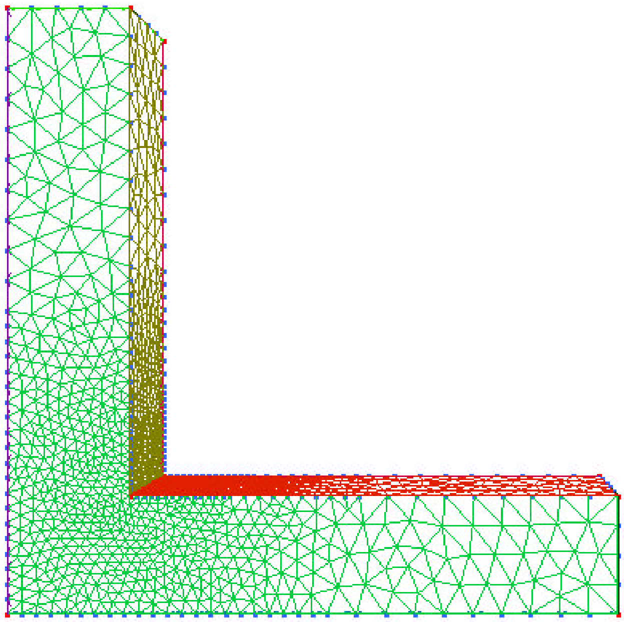


Fig. 4.2.3-1. Example of displayed input surface patch in VRML format

(2) Surface mesh

The surface mesh generated by **`TetMesh_P`** (see ***Chapter 4.2.1 (1)***) and the simultaneously created VRML output files are presented in Fig. 4.2.3-2 (displayed by a VRML browser).

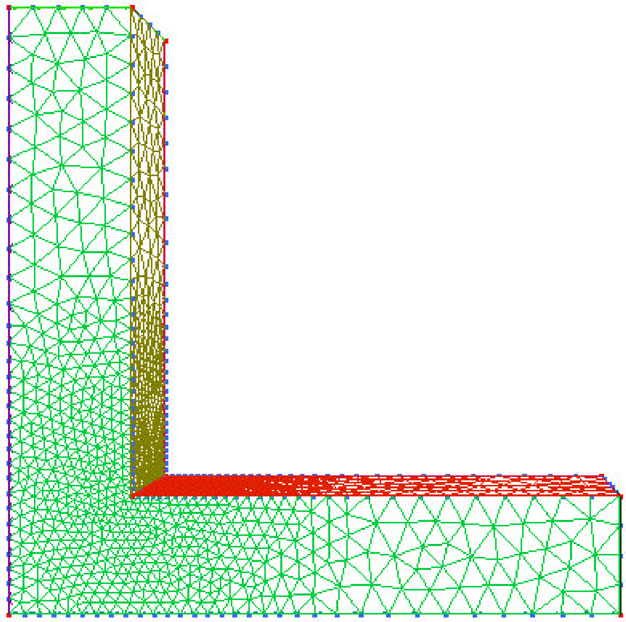


Fig. 4.2.3-2.　Example of surface mesh displayed by VRML browser

(3) Tetrahedral mesh surface

- The surface of tetrahedral mesh made by **TetMesh_M** can be displayed by a VRML browser opening the file with **c_e.wrl** at the end of the original surface patch file name.
- The nodes of tetrahedral mesh can be displayed by a VRML browser opening the file with **c_n.wrl** at the end of the original surface patch file name.
- The points and surface nodes (the surface mesh coincided with apexes of the triangle) are shown by red color; and the internal nodes are shown by blue color.

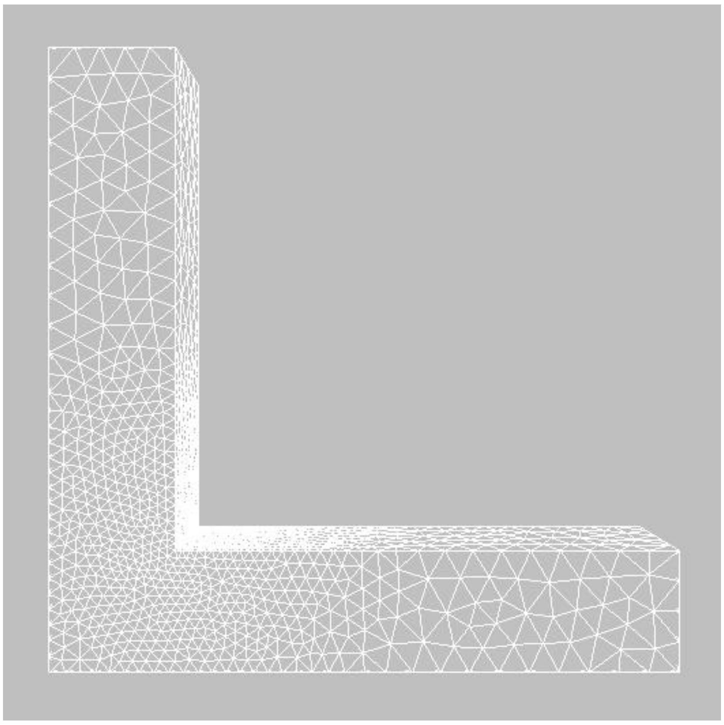


Fig. 4.2.3-3. Example of tetrahedral mesh surface displayed by VRML browser

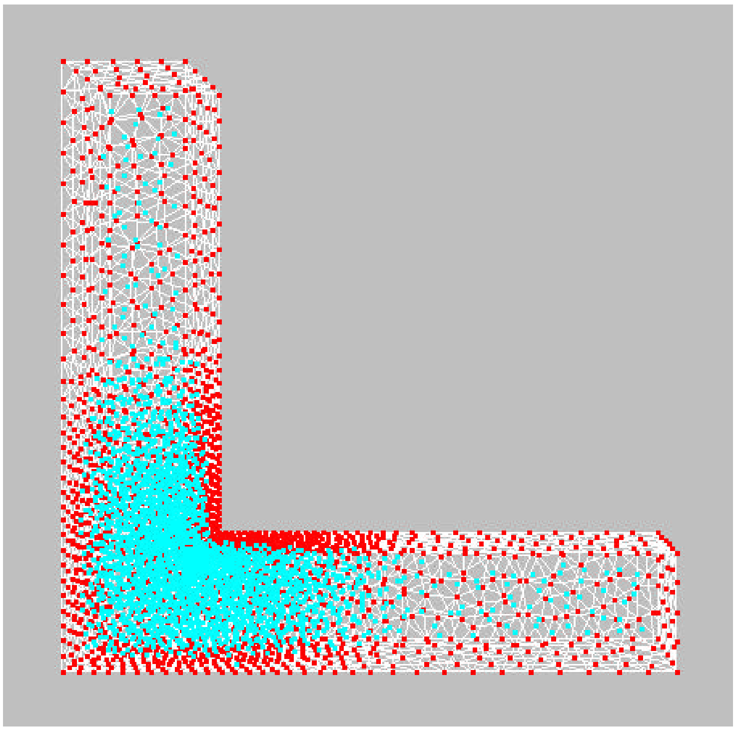


Fig. 4.2.3-4. Example of tetrahedral mesh nodes displayed by VRML browser (wireframe)

4.3. Program Execution Sample (multiple domain)

4.3.1. Program Execution

Sample data files are located in the subdirectory **sample_data**. An example of program execution in the case of multiple materials using the files **mat_in0102.pcm** and **mat_in0102.ptn** is shown here.

(1) An execution of **TetMesh_P** can be started by the following command:

```
% advtmesh9p   mat_in0102   -d   -p
```

The program will input two files: **mat_in0102.pcm**, and **mat_in0102.ptn**.
As a result, three files will be created: **mat_in0102c.pcc**, **mat_in0102c.ptn**, and **mat_in0102_c.wrl**.

(2) An execution of **TetMesh_M** can be started by the following command:

```
% advtmesh9m   mat_in0102c   -p
```

The program will input two files: **mat_in0102c.pcc**, and **mat_in0102c.ptn**.
As a result, three files will be created: **mat_in0102c.msh, mat_in0102c_n.wrl**, and **mat_in0102c_e.wrl**.

(3) An execution of **TetMesh_S** can be started by the following command:

```
% advtmesh9s   mat_in0102c
```

The program will input one file: **mat_in0102c.msh**.
As a result, one file will be created: **mat_in0102cs.msh**.

4.3.2. Execution Log

An output message log file will be generated after the program execution. Explanations about the message contents for the above-mentioned sample files are presented in *Appendix D, Appendix E* and *Appendix F*.

4.3.3. Sample Results

The VRML format files (VRML format Ver 1.0) generated after program execution can be displayed using a VRML browser.

(1) Input patch

By executing the following commands, the input original surface patch can be converted into VRML format:

```
% advtmesh9p   mat_in0102   -cr   -p
```

The file named **`mat_in0102_c.wrl`** will be created. The input patch file can be converted into the VRML format without correction of the patch by adding the option **`-cr`** to the execution command. Figure 4.3.3-1 shows an example of the VRML output file displayed by a VRML browser.

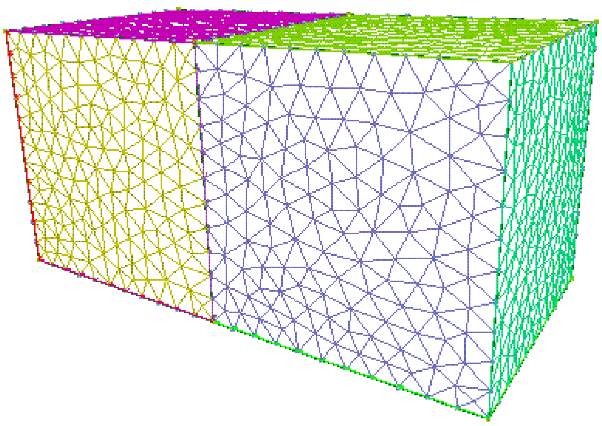


Fig. 4.3.3-1. Example of displayed input surface patch in VRML format

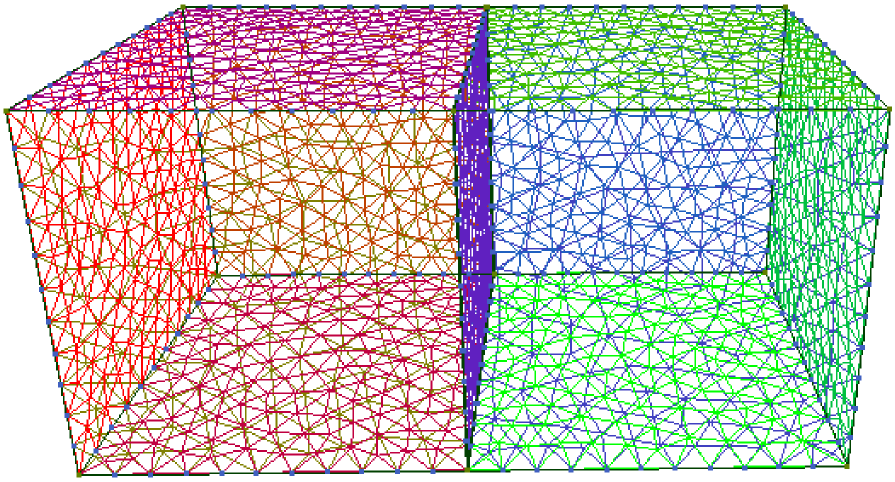


Fig. 4.3.3-2. Example of displayed input surface patch in VRML format (wireframe)

(2) Surface mesh

The surface mesh generated by **`TetMesh_P`** (see ***Chapter 4.2.1 (1)***) and the simultaneously created VRML output files are presented in Fig. 4.3.3-3 (displayed by a VRML browser).

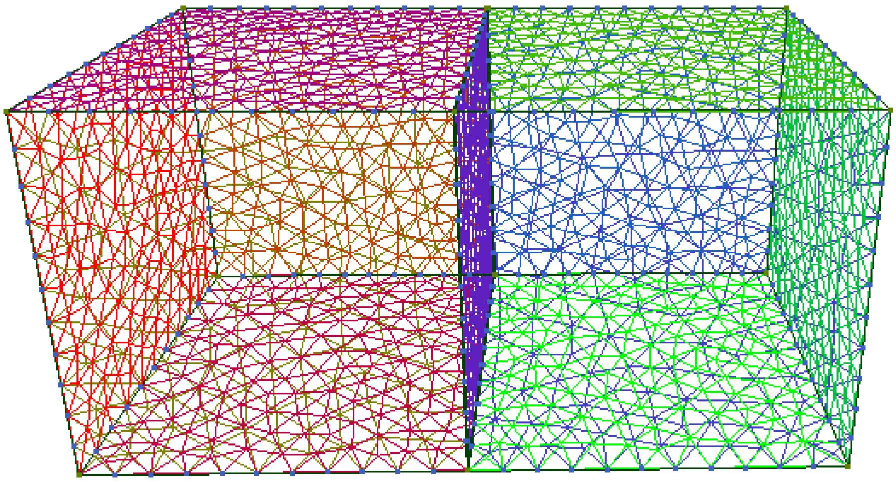


Fig. 4.3.3-3. Example of surface mesh displayed by VRML browser (wireframe)

(3) Tetrahedral mesh surface

- The surface of tetrahedral mesh made by **TetMesh_M** can be displayed by a VRML browser opening the file with **c_e.wrl** at the end of the original surface patch file name.
- The nodes of tetrahedral mesh can be displayed by a VRML browser opening the file with **c_n.wrl** at the end of the original surface patch file name.
- The points and surface nodes (the surface mesh coincided with apexes of the triangle) are shown by red color; and the internal nodes are shown by blue color.

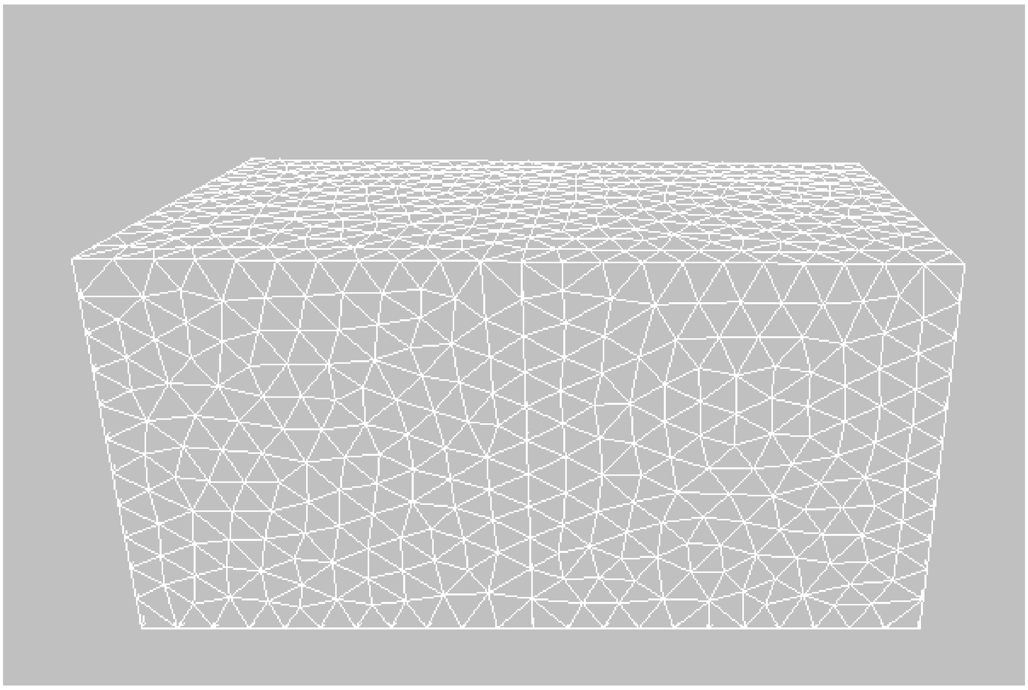


Fig. 4.3.3-4. Example of tetrahedral mesh surface displayed by VRML browser

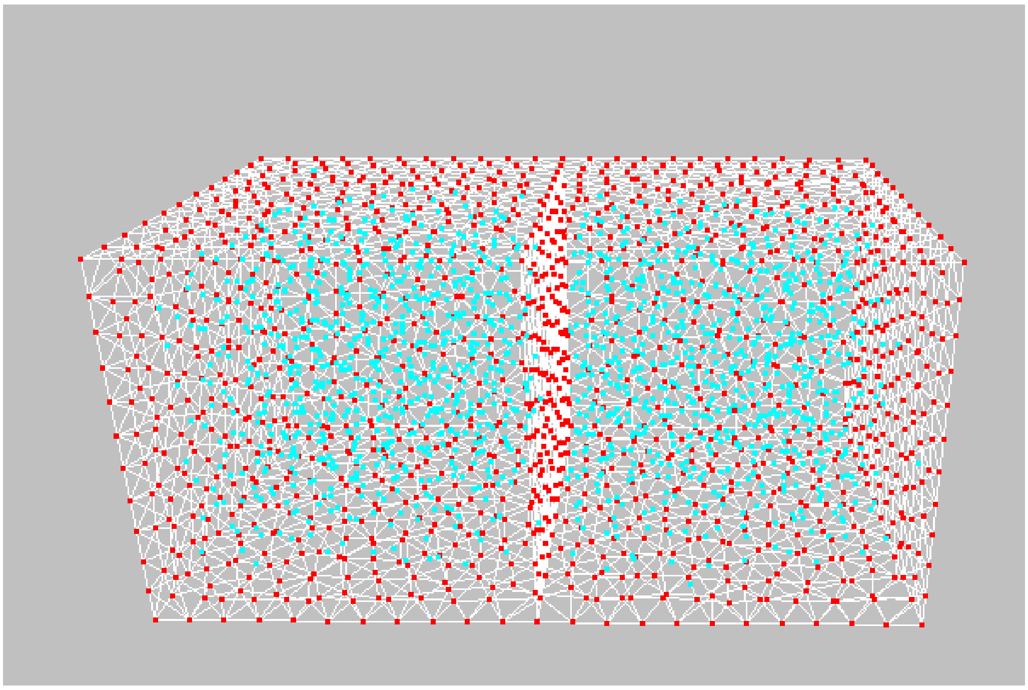


Fig. 4.3.3-5. Example of tetrahedral mesh nodes displayed by VRML browser (wireframe)

4.4. Command Options

4.4.1. Command Options for `TetMesh_P`

The surface mesh generation program **TetMesh_P** uses the technique of Pliant Delaunay re-triangulation, which concurrently smoothing and making the Delaunay triangulation of the input surface patch. Smoothing is achieved by the coupling of the method of Lennard-Jones potential approximation function applied by Bossen and Heckbert [1] for elements and the Laplacian smoothing method, where the node is moved toward the center of gravity, calculated taking into account the neighboring nodes. After this program performs, Delaunay tessellation which performs the above-mentioned smoothing, addition and deletion of the vertices according to node density control, by making the vertices of the inputted surface patch into starting points, the surface mesh which becomes the vertex arrangement in which the surface appears automatically is created. In Delaunay tessellation, when there are four or more points on the same circumference (referred to as degeneracy), uncertainty is in division. Therefore, the triangulation generated here was called "surface mesh", and it has distinguished from the surface of a tetrahedral mesh. In addition, on a domain boundary, even if it makes it the vertex arrangement in which degeneracy does not occur and creates a tetrahedral mesh separately for every domain, the triangle element of a border plane is made in agreement. The program **TetMesh_P** is executed according to the following processing procedures:

(1) Input of surface patch data.
(2) Input of nodal density control data.
(3) Deletion of extremely collapsed elements.
(4) Creation of surface groups.
(5) Delaunay re-triangulation without moving the input vertices.
(6) Search for the fine shapes and automatic adjustment of the node density.
(7) Rough deletion or addition of vertexes according to the node density distribution.
(8) Pliant Delaunay re-triangulation, where the smoothing and the Delaunay re-triangulation are concurrently performed.
(9) Protection of the boundary edges and adjacent surfaces.

To reach the convergence, Procedures (8) and (9) are performed twice in a loop.

The following command options can be used:

`-d` Specifies the nodal density control file. A file name different than the surface patch file name can be specified following after **`–d`** option. This option can be used only if **`-base`** option is <u>not</u> used.

`-base` Specifies the basic node intervals. The basic node interval should be specified after the **`–base`** option. The option can be used even if the node density control file is not prepared or the mesh is made homogeneous or automatically adjusted. An addition of

-d option acts the same as if the **BaseDistance** option would be specified in the node density control file. It is recommended to specify the node density control in the node density control file if complicated shapes are considered for the analysis because the automatic adjustment of the nodal density increases computing time.
This option can be used only if **-d** option is <u>not</u> used.
If neither **–d** nor **–base** options are specified, an average length of the input surface patch is applied as the basic node interval.

-eh Specifies the minimum value of the permissible ratio of the element's height to the local node interval. The minimum value can be specified within the range of 0 ~ 0.2. The value should be placed after the **–eh** option. If **–eh** option is not used, the default value of 0.05 will be automatically set. If some of the elements are extremely collapsed, the equation of surface cannot be set up or the very precise mesh system will be generated by automatic adjustment of the node density. In this case, the program deletes the elements, where the ratio of the element's height to the local node interval is smaller than a specified minimum value. If the value is not specified (only **–eh** option), the program does not delete any element.
If the value specified by this option is very large, there are conditions when the calculations can fail.

-sm Smoothing option.
The values **2** or **3** can be placed without a space after the **–sm** option. The default value is **3**.
If **3** is set up as the **–sm** value, the smoothing is achieved by the Bossen method together with the Laplacian smoothing method. An initial smoothing is done by the Bossen method and, after convergence is reached, the Laplacian smoothing method performs additional re-smoothing. Depending on the element shape considered for the analysis, the convergence of both methods may not be achieved simultaneously. This problem can be overcome by using the **–sm2** option, which eliminates the Laplacian smoothing.

-cr Using this option together with **–p** option, it is possible to display the input surface patch in the VRML form without patch correction.

-p[n] VRML file output option. The normalized output coordinates data can be prepared if the option **–pn** is specified. The program performs an element partitioning depending on the input surface patch angle. Using this option, the partitioned groups of surfaces stored in the VRML output file can be displayed by different colors. If the conversion is not reached, the overall object will be shown by blue color and the points, where the conversion is not reached will be illustrated by red color. The characters **_c.wrl** are added to the original specified surface patch data file name.

4.4.2. *Command Options for* `TetMesh_M`

The tetrahedral mesh generator program **`TetMesh_M`** is designed to generate a tetrahedral mesh system from the triangular surface patches generated by **`TetMesh_P`** by the addition of the internal node. The Bucketing method and the Delaunay triangulation method are adopted to generate the inner nodes and elements [2]. **`TetMesh_M`** is executed according to the following processing procedure:

(1) Input of surface patch data.
(2) Input of nodal density control data.
(3) Generation of surface node.
(4) Generation of internal node by Bucketing method.
(5) Element creation by Delaunay triangulation method.
(6) Outside-of-shape element deletion.
(7) Correction of the internal sliver elements.

In the case of multiple domains, Procedures (3) to (7) are performed for every domain.

The following command option can be used:

`-p` The VRML file output option. If this option is specified, two output VRML files are created, and **`_n.wrl`** and **`_e.wrl`** are added to the specified surface mesh data file names. The VRML file **`_n.wrl`** contains the input surface mesh and the generated node data. The surface node is displayed with a red color (fit to the vertex of the surface mesh), and the internal node is displayed with a blue color.
The VRML file **`_e.wrl`** contains the surface of tetrahedral mesh, which can be displayed.

4.4.3. *Command Options for* `TetMesh_S`

The quadratic tetrahedral mesh generator program **`TetMesh_S`** generates secondary nodes in the middle point of the tetrahedral element's edge.

The following command option can be used:

`-show` If this option is specified, the program will not output the quadratic tetrahedral mesh file, but will perform only the display of the number of nodes at the time of making it a quadratic element, and degrees of freedom.

5. Tetrahedral Mesh Evaluation Program

This program **TetMesh_E** evaluates a tetrahedral mesh. Evaluation criteria are edge length, dihedral angle, regular tetrahedral edge length of equivalent element volume, reciprocal of element's height aspect ratio, and the minimum element height. This program can evaluate also with a linear element or a quadratic element.

5.1. Execution of TetMesh_E

This program can be executed by the following commands:

advtmesh9e ***Tetrahedral_mesh_data_file_name*** **-p**

Input the linear or quadratic tetrahedral mesh data file name without file extension. The command options are explained below.

5.1.1. Program Execution Sample

Sample data files are located in the subdirectory **sample_data**. An example of program execution using the file **mati_in0102cs.msh** is shown here. This quadratic tetrahedral mesh file was made to perform **TetMesh_P** and **TetMesh_M** and created the secondary node by **TetMesh_S** from the input data of the same as **mat_in0102.pcm**.

An execution of **TetMesh_E** can be started by the following command:

```
% advtmesh9e   mati_in0102cs   -p   -d
```

The program will input two files: **mati_in0102cs.msh**, and **mati_in0102cs.ptn**.

As a result, two files will be created: **mati_in0102cs_chk.wrl** and **mati_in0102cs_har.wrl**. It displays inaccurate elements or the elements below a valuation-basis value.

5.1.2. Sample Results

The VRML format files (VRML format Ver 1.0) generated after program execution can be displayed using a VRML browser. Figure 5.1.2-1 shows an example of the VRML output file displayed by a VRML browser. In this case, the mesh contains no inaccurate elements or the elements below a valuation-basis value. So, the VRML output file displays only the element have minimum element height by red color.

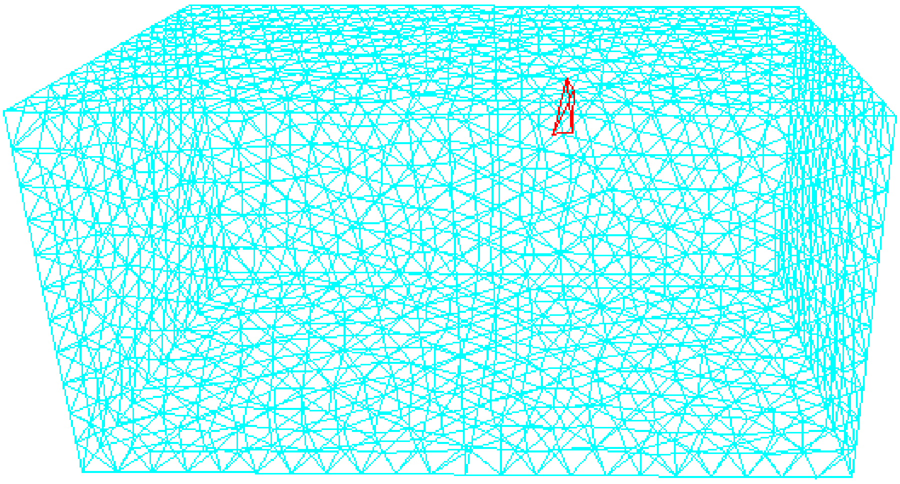


Fig. 5.1.2-1. Example of displayed evaluated mesh in VRML format (wireframe)

5.1.3. Execution Log

An output message log file will be generated after the program execution. Explanations about the message contents for the above-mentioned sample files are presented in *Appendix G*.

Evaluation criteria are as follows. In this description, <X> is the left hand side of the distribution table item, and <Y> is the right hand side of one. When **–d** option was specified, also displays the distribution of ratios to the local node interval.

(1) Edge length distribution
<X> edge length
<Y> number of edges
(2) Minimum and maximum dihedral angle distribution
<X> minimum or maximum dihedral angle of element face
<Y> number of elements
(3) Regular tetrahedral edge length of equivalent element volume distribution
<X> edge length
<Y> number of elements

(4) Reciprocal of element's height aspect ratio distribution

<X> $\text{Reciprocal of element's height aspect ratio} = \frac{2}{\sqrt{3}} \frac{\min(\text{Height of element})}{\max(\text{Edge length})}$

<Y> number of elements

(5) Minimum element height distribution

<X> element height

<Y> number of elements

5.2. Command Options

-p The VRML file output option. If this option is specified, two output VRML files are created, and **_chk.wrl** and **_har.wrl** are added to the specified tetrahedral mesh data file names. The VRML file **_chk.wrl** contains the surface of tetrahedral mesh and elements which dihedral angle is below/above a valuation-basis value (minimum:5 degree/maximum:175 degree). The VRML file **_har.wrl** contains the surface of tetrahedral mesh and elements which reciprocal of element's height aspect ratio is below a valuation-basis value (0.05) and the element have minimum element height.

-d Specifies the nodal density control file. The file name must be the same as the tetrahedral mesh file exclude of extension.

6. File Specifications

The table below presents the files used by this program and their contents

File Name	*Outline of File*
Surface patch data file **(.pcm)**	Data file, which contains the information about node coordinates and triangle patches (domain information, vertex coordinates and triangle connectivity).
Surface mesh data file **(.pcc)**	Temporary data file, which contains the information about node coordinates and triangle meshes (vertex coordinates, triangle connectivity, *etc*.).
Node density data file **(.ptn)**	Data file used for the node density control.
Mesh data file **(.msh)**	Data file, which contains the node coordinates and information on tetrahedral mesh (node coordinates and tetrahedral connectivity) output.
VRML file **(.wrl)**	File containing the surface patch, surface mesh or mesh surface data converted into the VRML format (VRML format Ver1.0).

6.1. Surface Patch Data File

The surface patch data have the following format:

- A vector, normal to the surface patch is faced toward the internal direction of the shape, and, looking from the outside of shape, the connectivity is shown directed clockwise.
- The vertex number starts from **`0`**.
- The file extension is **`.pcm`**.

```
1629 0 2                      ← Number of the vertices   reserved (0)   number of domains
150 -50 50                    ← The first X, Y, and Z coordinates of the vertex
50 -50 50
150 50 50
50 50 50
      ~ omitted ~
50 17.03994 -22.52797
50 20.23377 -15.25734
50 29.21514 -26.66399
50 41.96536 -15.88812         ← Coordinates X, Y, and Z of the 1629th vertex
1598 0 0                      ← Number of surface patches of the first domain  reserved (0)   reserved (0)
158 128 17                    ← The first row of the vertex number, which composes the surface patch
17 128 16
16 160 15
      ~ omitted ~
738 704 799
794 800 731
800 778 731                   ← Row of the vertex number of the 1598th surface patch
1652 0 0                      ← Number of surface patches of the second domain   reserved (0)   reserved (0)
960 958 1035                  ← The first row of the vertex number, which composes the surface patch
841 1025 959
816 817 930
      ~ omitted ~
1566 1627 1621
926 1614 1628
1586 1628 1615                ← Row of the vertex number of the 1652th surface patch
```

(Note) In the case of multiple domain, surface patches need to be closed for every domain. For every domain, each vertex and element is unique and must not be referred to from the different domains. Each vertex and each element also needs to be spatially in agreement in the border plane where two domains touch. Therefore, on the domain boundary, the vertex with the same coordinates value will be referred to from the element with which those with two and each belong to another domain. Each element (triangle) also needs to connect the vertex which serves as a pair, respectively (coordinates are in agreement), and it needs to be overlapped on the border plane. However, in the tetrahedral mesh finally generated by TetMesh_M, it is a share node on the border plane. Please also refer to the user manual of ADVENTURE_TriPatch.

6.2. Node Density Control File

(1) Outline of node density control data

The node density data are classified into the basic node interval and the local node density.

a). Basic node interval

The edge length, which is the basis of the mesh, is specified and the mesh is adjusted to follow this length.

b). Local node density

The local node density is used when the detailed mesh of an arbitrary part of the input shape is used. The local nodal density has two patterns: "Inverse proportion to the distance from the point" and "Inverse proportion to the distance from the segment". Specifying the local node density, the density intensity parameter and the applicable range are set.

(2) Example of nodal density application

Figs. 6.2-1 - 6.2-3 show examples of application of the node density. Three patterns can be seen: “Inverse proportion to the distance from the point” and “Inverse proportion to the distance from the segment (two patterns)”.

- The patterns, application results, and relationships between the density and the distance are shown in the figures.
- Here, 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 the specified point is shown if the option is set to “Inverse proportion to the distance from the point” and the distance from the specified segment is shown if the option is set to “Inverse proportion to the distance from the segment”.

< Example >

Fig. 6.2-1 demonstrates the case “Inverse proportion to the distance from the point”, picked up as an example. The density decreases according to the increasing distance from the point when this density is applied. The node interval grows with moving away from the point.

(Notes)

There are sample data of the node density control in the **sample_data** directory located in a subdirectory one level down from the top directory (**box1**, **box2**).

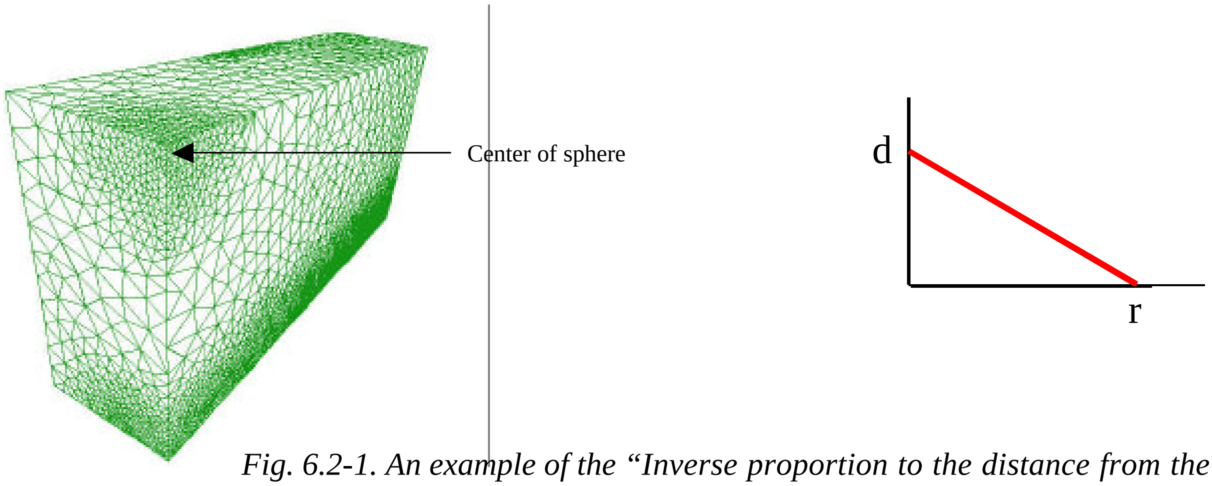


Fig. 6.2-1. An example of the "Inverse proportion to the distance from the point" pattern.　(NodalPatternOnPoint is used)

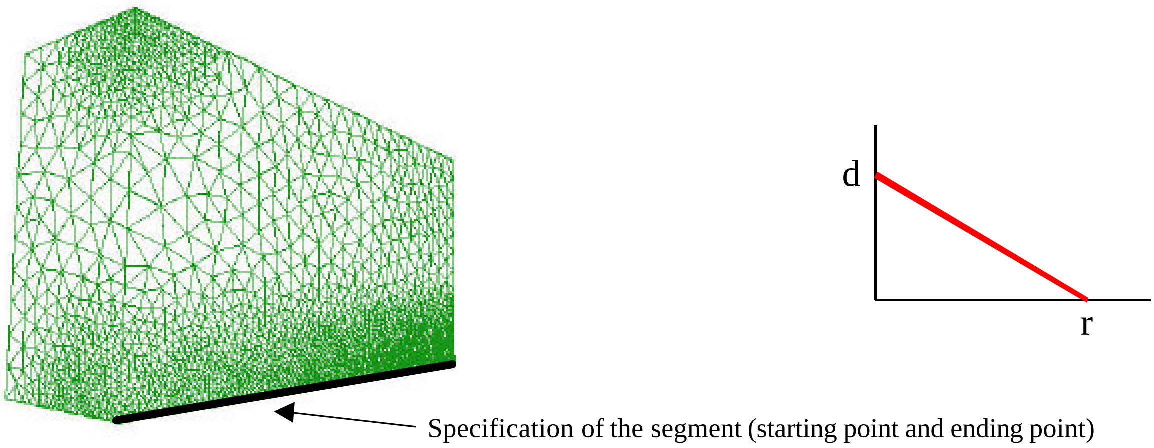


Fig. 6.2-2.　An example of the pattern "Inverse proportion to the distance from the segment". (NodalPatternOnLine is used)

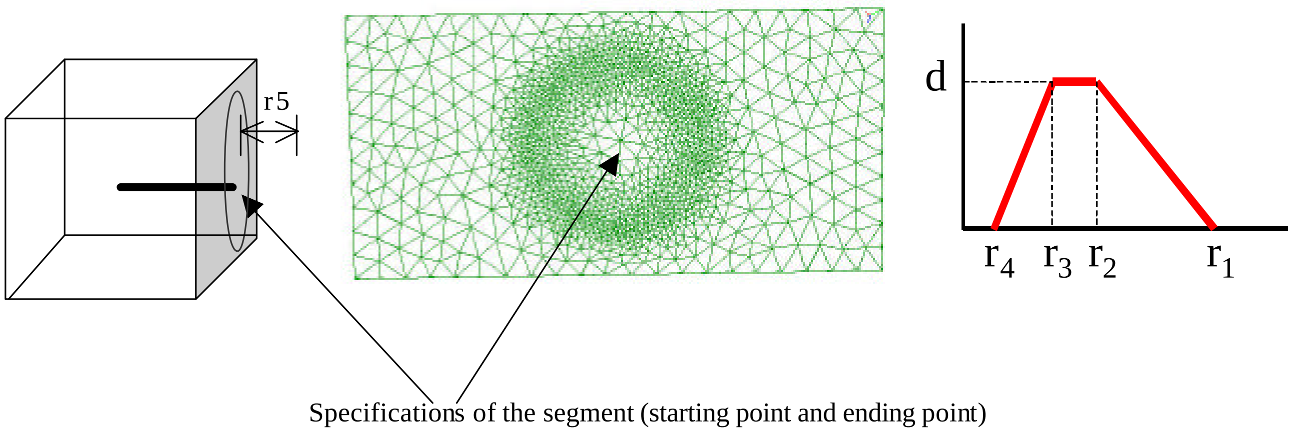


Fig. 6.2-3.　An example of the pattern "Inverse proportion to the distance from the segment". (NodalPatternOnCylinder is used)

(3) Format of nodal density control file

The format of the node density control data is shown below.

```
BaseDistance                  <----- Base node interval
  1.00E+00

NodalPatternOnPoint           <----- It is in inverse proportion to the distance from the point
  2.00E+01    4.7             <----- Range from the center of sphere (r),    Intensity of density
  1.00000E+01  0.00000E+00  0.00000E+00  <----- Coordinates of the center of the sphere

NodalPatternOnLine            <----- It is in inverse proportion to the distance from the segment
  2.00E+01    4.7             <----- Range from the segment (r),      Intensity of density
  1.00000E+01  0.00000E+00  0.00000E+00  <----- Coordinates of the starting point of the segment
  1.00000E+01  2.00000E+00  0.00000E+00  <----- Coordinates of the end of the segment

NodalPatternOnCylinder            <-----It is in inverse proportion to the distance from the segment
                                        (The range of the nodal density can be specified.)
12.0   10.0   9.0   8.0   3.0   1.5   <--- Range 1 to Range 5 (r1~r5),    Intensity of density
347.1   0.0   100.0               <----- Coordinates of the starting point of the segment
406.1   0.0   100.0               <----- Coordinates of the end of the segment
```

- **`BaseDistance`** is essential to execute the program.
- Other items (**`NodalPatternOnPoint`**, **`NodalPatternOnLine`**, **`NodalPatternOnCylinder`**) are used to make the detailed mesh at an arbitrary position of the input shape.
- The file extension is **`.ptn`**.

6.3. Mesh Data File

The tetrahedron mesh data use the following format:

- Refer to Fig 6.3-1 for the mesh connectivity.
- The node number starts from **`0`**.
- The file extension is .**`msh`**.

```
170776                              ← Number of elements
19900 19890 22150 22160             ← Node row, which composes the first element
24000 23810 23830 23990
30130 30150 32470 32690
730 60 58 61
730 61 58 62
      ~  Omitted ~
38139 38601 38602 38606
38139 38606 38602 38607
38266 38139 38602 38607
38274 38139 38266 38607             ← Node row, which composes the 170776th element
38608                               ← Number of nodes
-31.223900 -3.384220 -5.000000      ← Coordinates of the first node
-31.223900 -3.384220 -4.520000
-31.223900 -3.384220 -3.960000
-31.223900 -3.384220 -3.430000
      ~  Omitted ~
31.308800  2.412930  5.000000
31.280500  2.736690  5.000000
31.252200  3.060460  5.000000
31.223900  3.384220  5.000000       ← Coordinates of the 38608th node
2                                   ← Number of domains
2567                                ← Number of elements of the first domain
0                                   ← The first element number of the first domain
1
2
      ~  Omitted ~
2566                                ← The 2567th element number of the first domain
2052                                ← Number of elements of the second domain
2567                                ← The first element number of the second domain
2568
      ~  Omitted ~
4617
4618                                ← The 2052th element number of the second domain
```

(Note) The case mentioned above is for the linear tetrahedral element. The element's connectivity becomes 10 for the quadratic tetrahedral element.

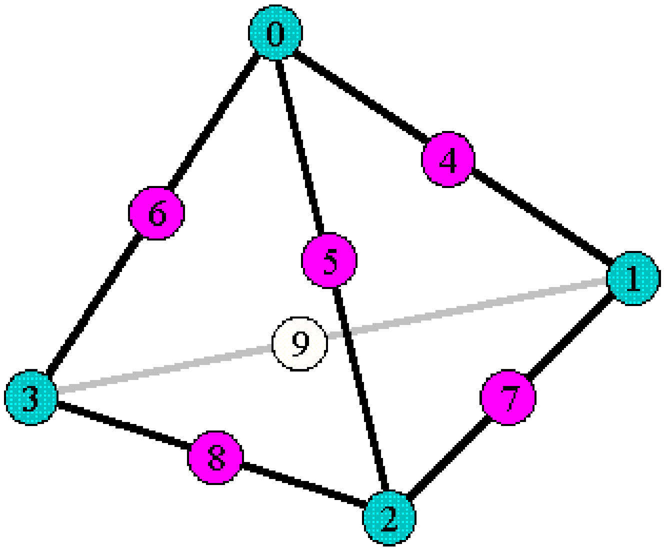


Fig. 6.3-1.　Node connectivity of tetrahedral mesh

References

[1]. Frank J. Bossen, Paul S. Heckbert, “A Pliant Method for Anisotropic Mesh Generation”, 5th Annual Internatonal Meshing Roundtable, (1996).

[2]. Yagawa, G., Yoshimura, S. and Nakao, K., “Automatic Mesh Generation of Complex Geometries Based on Fuzzy Knowledge Processing and Computational Geometry”, Integrated Computer-Aided Engineering 2, pp. 265-282, (1995).

Appendix A. Execution Log of TetMesh_P (*Single domain*)

Explanations of the execution log of surface mesh generation program **`TetMesh_P`** are shown below.

```
ADVENTURE TetMesh_P                                        ← Program name

input patch file:adventure_manual_data01.pcm               ← File name of surface patch input

number of input vertices      =      2213                  ← Number of input vertices
number of volumes             =        1                   ← Number of input domains
number of input elements      =      4422                  ← Number of input elements
range of x-axis               =  -7.1576560E+01  -2.1320670E+01  ← Range of x-axis
range of y-axis               =  -1.6141980E+00   4.8337110E+01  ← Range of y-axis
range of z-axis               =   0.0000000E+00   1.0000000E+01  ← Range of z-axis

input density control file:adventure_manual_data01.ptn ← File name of input node density control
BaseDistance                  =   2.5000000E+00            ← Base node distance
number of density function    =        1                   ← Number of input node density functions
        maximum range         =   2.0000000E+01
        maximum strength      =   3.5000000E+00

number of edges               =       6633                 ← Number of edges
minimum edge length           =   4.1844367E-01     555     737
maximum edge length           =   3.5376171E+00       2     326
average edge length           =   1.3090199E+00

Check Surfaces
number of surface             =        1                   ← Number of surfaces

Edge Correction start                                      ← Beginning of inferior patch deletion
iteration loop, change count =    1       0                   (Repetition)
iteration loop, change count =    2       0                ← Number of corrected elements

Surface Patch Grouping                                     ← Beginning of surface group making
number of Volumes             =        1                   ← Number of domains
number of Bodies              =        1                   ← Number of bodies
number of Surfaces            =        1                   ← Number of surfaces
number of fixed main vertices   =       12                 ←Number of fixed vertices
number of boundary edge groups  =       18                 ←Number of boundary edge groups
  open   edge group           =       18
  closed edge group           =        0
  fixed  edge               =      317
number of face groups         =        8                   ← Number of face groups

Node bucket registration

Delaunay re-triangulation              ← Beginning of Delaunay re-triangulation of the input vertices
```

```
LEPP - Rough vertex density control start                 ← Division of a bad formal element
iteration loop, change count =    1       1                        (Repetition)
iteration loop, change count =    2       1
iteration loop, change count =    3       0
LEPP - Rough vertex density control : iteration converged  ← It is not necessary to converge.

Shape dependent density control .....              ←Density control by shape
Vertex density control start                      ← Rough initial vertices addition and deletion
iteration loop, change count =    1       2             (Repetition)
iteration loop, change count =    2       0
Vertex density control : iteration converged        ← It is not necessary to converge.

Pre-smoothing of boundary edge          ← Only the point of boundary edge is precedence
smoothing.

Pliant Delaunay retriangulation start                      ← Beginning of smoothing
outer/inner iteration, remained =    1    1    1986     ← Convergence loop
outer/inner iteration, remained =    1    2    1997
outer/inner iteration, remained =    1    3    1946
outer/inner iteration, remained =    1    4    1855
      ~  Omitted ~
outer/inner iteration, remained =    1   198      9
outer/inner iteration, remained =    1   199     12
outer/inner iteration, remained =    1   200     10
**** inner iteration not converged ****
outer/inner iteration, remained =    2    1       7
outer/inner iteration, remained =    2    2       2
outer/inner iteration, remained =    2    3       1
outer/inner iteration, remained =    2    4       0
**** inner iteration converged ****                         ←Inner loop convergence
--- outer iteration converged ------- loop    2             ←Outer loop convergence
Laplacian smoothing start                              ← Re-smoothing by Laplacian smoothing
outer/inner iteration, remained =    3    1    2173
outer/inner iteration, remained =    3    2     251
outer/inner iteration, remained =    3    3     161
      ~  Omitted ~
outer/inner iteration, remained =    3    8       3
outer/inner iteration, remained =    3    9       1
outer/inner iteration, remained =    3   10       0
**** inner iteration converged ****                        ←Inner loop convergence
boundary edge protection : outer loop    =          3      ←Boundary edge protection
boundary edge protection : change count  =          0
surface protection       : outer loop    =          3      ←Surface protection
surface protection       : change count  =          0
--- outer iteration converged ------- loop    3            ←Outer loop convergence

number of vertices          =      2185                    ←Number of output vertices
number of elements          =      4366                    ←Number of output elements

open:adventure_manual_data01c.pcc          ← File name of output surface mesh data
open:adventure_manual_data01c.ptn            ← File name of output correction node density
control
open:adventure_manual_data01_c.wrl         ← File name of VRML output
```

```
maximum allocate 3762383 Bytes  3.588 MBytes

start: Thu Mar  6 21:05:18 2003
stop : Thu Mar  6 21:05:30 2003

elapsed      9.13 sec (user)
           0.09 sec (system)
           9.22 sec (total)
```

Appendix B. Execution Log of TetMesh_M (*Single domain*)

Explanations of the execution log of tetrahedral mesh generation program **TetMesh_M** are shown below.

```
ADVENTURE TetMesh_M                                        ← Program name
read densityFunction << adventure_manual_data01c.ptn      ← File name of input node density control
read file << adventure_manual_data01c.pcc                  ← File name of surface mesh data input
read domain patch
total vertices              = 2185                         ← Number of input vertices
total number of volume      = 1                            ← Number of domains
set domain data
set interior nodes
set local patches and vertices :: region number = 0
local use total nodes       = 2185                         ← Number of nodes of domain
domain patches              = 4366                         ← Number of input patches of domain
number of vertices          = 2185
number of patches           = 4366
 set duplicate vertices      = 0
bounding box ( -71.5766, -1.6142, 0 )  ( -21.3207, 48.3371, 10 ) ←Range of coordinates
baseMeshSize  = 2.5                                        ← Basic node interval
minInterval   = 0.714286                                   ← Minimum node interval
node Generation on Vertex                                  ← Shape surface node generation
 add vertices      = 2185
 add interior nodes = 0
node Generation in Body                    ← Beginning of shape's internal node generation
bucket --------------------------------- 0/1520   ← Bucket number of final node generation
number of nodes 2185                               ← Number of accumulated nodes
bucket --------------------------------- 76/1520
number of nodes 2185
bucket --------------------------------- 152/1520
number of nodes 2528
bucket --------------------------------- 228/1520
number of nodes 3064
bucket --------------------------------- 304/1520
number of nodes 3756
bucket --------------------------------- 380/1520
     ~  Omitted ~
bucket --------------------------------- 1140/1520
number of nodes 4976
bucket --------------------------------- 1216/1520
number of nodes 4983
bucket --------------------------------- 1292/1520
number of nodes 4989
bucket --------------------------------- 1368/1520
number of nodes 4996
bucket --------------------------------- 1444/1520
number of nodes 4996
bucket --------------------------------- 1520/1520
number of nodes 4996                     ←Final node generated by bucketing method
```

```
Delaunay Triangulation                              ← Beginning of Delaunay tessellation
add Points
remove Outer Tetrahedron                            ← Deletion of external element
correct Sliver Elements                             ← Beginning of sliver element correction
number of additional points for sliver loop-1 = 275  ← Number of nodes added
total number of points = 5271
number of additional points for sliver loop-2 = 6
total number of points = 5277
--------------count On Vertex       = 2185          ← Number of surface nodes
--------------count In Body         = 3092          ← Number of internal nodes
              total 1ry node        = 5277          ← Total number of nodes
number of Elements                  = 25812         ← Number of elements
write .wrl >> adventure_manual_data01c_e.wrl  ← File name of VRML output (surface of element)
write .wrl >> adventure_manual_data01c_n.wrl  ← File name of VRML output (node)
clear all

total ---------------------
number of total nodes               = 5277          ← Total number of nodes
 volume 0  =    5277 :    2185 ( v )      0 ( dv )    3092 ( b )
number of total Elements     = 25812                ← Total number of elements
 volume 0  =   25812

write .msh >> adventure_manual_data01c.msh          ← File name of mesh output

start   : Thu Mar  6 21:07:19 2003
end     : Thu Mar  6 21:07:37 2003
interval      =         18
process time  =         14.78

END advtmesh9m
```

Appendix C. Execution Log of TetMesh_S (*Single domain*)

Explanations of the execution log of tetrahedral mesh generation program **TetMesh_S** are shown below.

```
reading... adventure_manual_data01c.msh              ← File name of input mesh
 linear tetrahedron ---> quadratic tetrahedron
 number of nodes      =        5277                 ← Number of nodes
 number of elements   =       25812                 ← Number of elements

 writing... adventure_manual_data01cs.msh           ← File name of quadratic mesh output
 number of nodes      =       38548                 ← Number of nodes
 number of elements   =       25812                 ← Number of elements
 number of edges      =       33271                 ← Number of edges
 DOF(1ry)             =       15831          ← Degrees of freedum (linear element)
 DOF(2ry)             =         115644              ← Degrees of freedum (quadratic element)
 number of regions    =           1                 ← Number of domains
 range of x-axis      = -7.157660e+01  -2.132070e+01  ← Range of x-axis
 range of y-axis      = -1.614200e+00  4.833710e+01   ← Range of y-axis
 range of z-axis      = 0.000000e+00  1.000000e+01    ← Range of z-axis

 elapsed time = 1.07 sec
```

Appendix D. Execution Log of TetMesh_P (*Multiple domain*)

Explanations of the execution log of surface mesh generation program **TetMesh_P** are shown below.

```
ADVENTURE TetMesh_P                                        ← Program name

input patch file:mat_in0102.pcm                            ← File name of surface patch input

number of input vertices       =      1629                 ← Number of input vertices
number of volumes            =         2                   ← Number of input domains
Volume         1
  number of input elements     =      1598
Volume         2
  number of input elements     =      1652
total number of input elements  =      3250                ← Number of input elements
range of x-axis              =  -5.0000000E+01  1.5000000E+02  ← Range of x-axis
range of y-axis              =  -5.0000000E+01  5.0000000E+01  ← Range of y-axis
range of z-axis              =  -5.0000000E+01  5.0000000E+01  ← Range of z-axis

input density control file:mat_in0102.ptn                  ← File name of input node density control
BaseDistance                 =  1.0000000E+01              ← Base node distance
number of density function     =         0                 ← Number of input node density functions

number of edges              =      4875                   ← Number of edges
minimum edge length           =  5.7324549E+00     694      770
maximum edge length           =  1.5546799E+01     605      633
average edge length           =  9.3825988E+00

Check Surfaces
Volume          1
  number of surface          =         1                   ← Number of surfaces in first domain
Volume          2
  number of surface          =         1                   ← Number of surfaces in second domain
maximum number of dup.vertex    =         1
maximum number of dup.edge     =         1

Edge Correction start                                      ← Beginning of inferior patch deletion
iteration loop, change count =    1       0                   (Repetition)
iteration loop, change count =    2       0                ← Number of corrected elements

Surface Patch Grouping                                     ← Beginning of surface group making
number of Volumes            =         2                   ← Number of domains
number of Bodies             =         2                   ← Number of bodies
number of Surfaces            =        2                   ← Number of surfaces
number of fixed main vertices   =        16                ← Number of fixed vertices
number of boundary edge groups  =        24                ← Number of boundary edge groups
  open   edge group          =        24
  closed edge group          =         0
  fixed  edge              =       264
number of face groups          =        12                 ← Number of face groups
```

```
Node bucket registration

Delaunay re-triangulation              ← Beginning of Delaunay re-triangulation of the input vertices

LEPP - Rough vertex density control start            ← Division of a bad formal element
iteration loop, change count =    1      0                 (Repetition)
LEPP - Rough vertex density control : iteration converged ← It is not necessary to converge

Shape dependent density control .....              ←Density control by shape
Vertex density control start                     ← Rough initial vertices addition and deletion
iteration loop, change count =    1      0             (Repetition)
Vertex density control : iteration converged       ← It is not necessary to converge

Pre-smoothing of boundary edge        ← Only the point of boundary edge is precedence smoothing

Pliant Delaunay retriangulation start                    ← Beginning of smoothing
outer/inner iteration, remained =    1    1    1422    ← Convergence loop
outer/inner iteration, remained =    1    2    1424
outer/inner iteration, remained =    1    3    1346
      ~ Omitted ~
outer/inner iteration, remained =    1   37       4
outer/inner iteration, remained =    1   38       2
outer/inner iteration, remained =    1   39       0
**** inner iteration converged ****                      ←Inner loop convergence
--- outer iteration converged ------- loop    1          ←Outer loop convergence
Laplacian smoothing start                          ← Re-smoothing by Laplacian smoothing
outer/inner iteration, remained =    2    1    1394
outer/inner iteration, remained =    2    2     110
outer/inner iteration, remained =    2    3      72
outer/inner iteration, remained =    2    4      36
outer/inner iteration, remained =    2    5      16
outer/inner iteration, remained =    2    6       4
outer/inner iteration, remained =    2    7       1
outer/inner iteration, remained =    2    8       0
**** inner iteration converged ****                      ← Inner loop convergence
boundary edge protection : outer loop   =        2       ← Boundary edge protection
boundary edge protection : change count  =        0
surface protection      : outer loop   =        2        ← Surface protection
surface protection      : change count  =       0
duplicate edge protection : outer loop  =        2       ← Duplication edge protection
duplicate edge protection :change count =        0
--- outer iteration converged ------- loop    2           ← Outer loop convergence

number of vertices          =      1395                   ← Number of output vertices
number of elements          =      3088                   ← Number of output elements

open:mat_in0102c.pcc                    ← File name of output surface mesh data
open:mat_in0102c.ptn                    ← File name of output correction node density control
open:mat_in0102_c.wrl                   ← File name of VRML output

maximum allocate 2801176 Bytes  2.671 MBytes
```

```
start: Thu Mar  6 21:06:28 2003
stop : Thu Mar  6 21:06:34 2003

elapsed      5.40 sec (user)
           0.11 sec (system)
           5.51 sec (total)
```

Appendix E. Execution Log of TetMesh_M (*Multiple domain*)

Explanations of the execution log of tetrahedral mesh generation program **TetMesh_M** are shown below.

```
ADVENTURE TetMesh_M                                        ← Program name
read densityFunction << mat_in0102c.ptn                    ← File name of input node density control
read file << mat_in0102c.pcc                               ← File name of surface mesh data input
read domain patch
total vertices         = 1395                              ← Number of input vertices
total number of volume = 2                                 ← Number of domains
set domain data                                            ← Beginning of first domain
set interior nodes
set local patches and vertices :: region number = 0
local use total nodes  = 1395
domain patches         = 1540
number of vertices   = 772                                 ← Number of nodes of domain
number of patches     = 1540                               ← Number of input patches of domain
 set duplicate vertices  = 0
bounding box ( 5, -5, -5 )  ( 15, 5, 5 )                   ← Range of coordinates
baseMeshSize  = 1                                          ← Basic node interval
minInterval   = 1                                          ← Minimum node interval
node Generation on Vertex                                  ← Shape surface node generation
 add vertices       = 772
 add interior nodes  = 0
node Generation in Body                               ← Beginning of shape's internal node generation
bucket --------------------------------- 0/64          ← Bucket number of final node generation
number of nodes 772                                    ← Number of accumulated nodes
bucket --------------------------------- 16/64
number of nodes 897
bucket --------------------------------- 32/64
number of nodes 1101
bucket --------------------------------- 48/64
number of nodes 1300
bucket --------------------------------- 64/64
number of nodes 1396                                  ←Final node generated by bucketing method
Delaunay Triangulation                                 ← Beginning of Delaunay tessellation
add Points
remove Outer Tetrahedron                               ← Deletion of external element
correct Sliver Elements                                ← Beginning of sliver element correction
number of additional points for sliver loop-1 = 59     ← Number of nodes added
total number of points = 1455
number of additional points for sliver loop-2 = 1
total number of points = 1456
--------------count On Vertex  = 772                   ← Number of surface nodes in first domain
--------------count In Body   = 684                    ← Number of internal nodes in first domain
              total 1ry node   = 1456                  ← Total number of nodes in first domain
  number of Elements           = 6491                  ← Number of elements in first domain
write .wrl >> mat_in0102c_e.wrl                   ← File name of VRML output (surface of element)
write .wrl >> mat_in0102c_n.wrl                   ← File name of VRML output (node)
clear all
```

```
set domain data                                          ← Beginning of second domain
set interior nodes                                      (followings are the same
set local patches and vertices :: region number = 1      processing as first domain)
local use total nodes  = 1395
domain patches         = 1548
number of vertices   = 776
number of patches    = 1548
 set duplicate vertices = 0
bounding box ( -5, -5, -5 )  ( 5, 5, 5 )
baseMeshSize  = 1
minInterval   = 1
node Generation on Vertex
 add vertices         = 776
 add interior nodes   = 0
node Generation in Body
bucket --------------------------------- 0/64
number of nodes 776
bucket --------------------------------- 16/64
number of nodes 901
bucket --------------------------------- 32/64
number of nodes 1107
bucket --------------------------------- 48/64
number of nodes 1309
bucket --------------------------------- 64/64
number of nodes 1402
Delaunay Triangulation
add Points
remove Outer Tetrahedron
correct Sliver Elements
number of additional points for sliver loop-1 = 62
total number of points = 1464
number of additional points for sliver loop-2 = 1
total number of points = 1465
--------------count On Vertex  = 776        ← Number of surface nodes in second domain
--------------count In Body    = 689        ← Number of internal nodes in second domain
              total 1ry node   = 1465       ← Total number of nodes in second domain
number of Elements             = 6516       ← Number of elements in second domain
write .wrl >> mat_in0102c_e.wrl
write .wrl >> mat_in0102c_n.wrl
clear all

total  ---------------------
number of total nodes    = 2768                ← Total number of nodes
volume 0  =   1456 :    772 ( v )      0 ( dv )     684 ( b )
volume 1  =   1465 :    623 ( v )    153 ( dv )     689 ( b )
                                   ← Domain no.  Total number of nodes   Number of surface nodes
                                     Number of share node with another domain   Number of inner nodes
number of total Elements = 13007               ← Total number of elements
                     volume 0  = 6491          ← Number of elements in first domain
                     volume 1  = 6516          ← Number of elements in second domain

write .msh >> mat_in0102c.msh                  ← File name of mesh output

start  : Thu Mar  6 21:09:51 2003
```

```
end    : Thu Mar  6 21:09:57 2003
interval      =         6
process time  =         5.52

END advtmesh9m
```

Appendix F. Execution Log of TetMesh_S (*Multiple domain*)

Explanations of the execution log of tetrahedral mesh generation program **TetMesh_S** are shown below.

```
reading... mat_in0102c.msh                              ← File name of input mesh
 linear tetrahedron ---> quadratic tetrahedron
 number of nodes    =        2768                       ← Number of nodes
 number of elements =        13007                      ← Number of elements

 writing... mat_in0102cs.msh                            ← File name of quadratic mesh output
 number of nodes    =       19826                       ← Number of nodes
 number of elements =        13007                      ← Number of elements
 number of edges    =       17058                       ← Number of edges
 DOF(1ry)          =        8304                ← Degrees of freedom (linear element)
 DOF(2ry)             =          59478          ← Degrees of freedom (quadratic element)
 number of regions  =           2                       ← Number of domains
 range of x-axis    = -5.000000e+01   1.500000e+02      ← Range of x-axis
 range of y-axis    = -5.000000e+01   5.000000e+01      ← Range of y-axis
 range of z-axis    = -5.000000e+01   5.000000e+01      ← Range of z-axis

 elapsed time = 0.56 sec
```

Appendix G. Execution Log of TetMesh_E

Explanations of the execution log of tetrahedral mesh generation program **TetMesh_E** are shown below.

```
input mesh file : mati_in0102cs.msh              ← File name of input mesh
quadratic element                                ← Element type
number of elements   =      13007                ← Number of elements
number of nodes             =      19826         ← Number of nodes
number of volume            =          2         ← Number of domains
             volume         1 =       6491       ← Number of elements in first domain
             volume         2 =       6516       ← Number of elements in second domain
X coordinates range         =  -5.0000000e+01     1.5000000e+02 ← Range of x-axis
Y coordinates range         =  -5.0000000e+01     5.0000000e+01 ← Range of x-axis
Z coordinates range         =  -5.0000000e+01     5.0000000e+01 ← Range of x-axis

read densityFunction : mati_in0102cs.ptn          ← File name of input node density control
base mesh size       =   1.0000000e+01            ← Basic node interval
number of 1ry nodes     =      2768               ← Number of primary nodes
number of 2ry nodes     =     17058               ← Number of secondary nodes
number of total triangles      =      27298       ← Total number of triangles
number of surface triangles     =      2568       ← Number of surface triangles
number of illegal elements     =         0        ← Number of illegal elements
number of surface nodes         =      5138       ← Number of surface nodes
  1ry surface nodes            =      1286        ← Number of surface primary nodes
  2ry surface nodes            =      3852        ← Number of surface secondary nodes

number of total edges     =      17058            ← Total number of edges
number of interior edges   =      13206           ← Number of inner edges
number of surface edges    =      3852            ← Number of surface edges

Edge Length ---------   ratio to local size --    ← Edge length distribution
  1   0.01      0    0.01      0
  2   0.10      0    0.10      0
  3   0.20      0    0.20      0
      ~  Omitted ~
 20  10.00    2028   1.90      77
 21  20.00   13429   2.00      22
 22  30.00       1   3.00       1

minimum edge length                         =      4.7845943
maximum edge length                         =     20.4460551
average edge length                         =     11.7784436
minimum edge length ratio to local size   =       0.4784594
maximum edge length ratio to local size   =       2.0446055
average edge length ratio to local size   =       1.1778444
All of 2ry nodes are on the edge middle point.

Dihedral angle -- minimum    maximum       ← Minimum and Maximum dihedral angle distribution
  1     5        0        0
```

```
  2     10        1        0
  3     15       63        0
  4     20      155        0
      ~  Omitted ~
 32    160        0       73
 33    165        0        4
 34    170        0        0
 35    175        0        0
 36    180        0        0

minimum in minimum dihedral angle       =     9.2923577
maximum in minimum dihedral angle       =    70.1424314
minimum in maximum dihedral angle       =    71.9159013
maximum in maximum dihedral angle       =   164.2051316
average of minimum dihedral angle       =    46.8098209
average of maximum dihedral angle       =   102.6634915
average of dihedral angle               =    69.5994465
number of sliver elements               =             0
  max > 175 & min < 5                   =             0
  max > 175                             =             0
  min < 5                               =             0
```

```
Regular tetrahedral edge length
```
← Regular tetrahedral edge length of equivalent element volume distribution

```
   distribution  of equivalent element volume --   ratio to local size –
  1    0.01       0                                  0.01       0
  2    0.10       0                                  0.10       0
  3    0.20       0                                  0.20       0
  4    0.30       0                                  0.30       0
  5    0.40       0                                  0.40       0
  6    0.50       0                                  0.50       0
  7    0.60       0                                  0.60       0
  8    0.70       0                                  0.70      14
  9    0.80       0                                  0.80     193
 10    0.90       0                                  0.90     645
 11    1.00       0                                  1.00    2009
 12    2.00       0                                  1.10    4362
 13    3.00       0                                  1.20    4028
 14    4.00       0                                  1.30    1386
 15    5.00       0                                  1.40     312
 16    6.00       0                                  1.50      50
 17    7.00      14                                 1.60       8
 18    8.00     193                                  1.70       0
 19    9.00     645                                  1.80       0
 20   10.00    2009                                  1.90       0
 21   20.00   10146                                         2.00       0
 22   30.00       0                                         3.00       0

minimum edge length                        =      6.2320287
maximum edge length                        =     15.8144499
average edge length                        =     10.8046704
minimum edge length ratio to local size  =      0.6232029
maximum edge length ratio to local size  =      1.5814450
average edge length ratio to local size  =      1.0804670
number of illegal elements               =             0
```

```
Inverse of Element Height Aspect Ratio --      ← Reciprocal of element's height aspect ratio distribution
  1   0.05       0
  2   0.10       0
  3   0.15       1
  4   0.20      42
  5   0.25      85
      ~  Omitted ~
 13   0.65    2017
 14   0.70    1669
 15   0.75    1002
 16   0.80     548
 17   0.85     246
 18   0.90     138
 19   0.95      30
 20   1.00       7

minimum inv. element height aspect ratio       =     0.1266485
maximum inv. element height aspect ratio       =     0.9763302
average inv. element height aspect ratio       =     0.5775726
number of lower than regulation( 0.050 )        =           0

Minimum Element Height --  ratio to local size – ← Minimum element height distribution
  1   0.01       0        0.01      0
  2   0.10       0        0.10      0
  3   0.20       0        0.20      0
  4   0.30       0        0.30     23
  5   0.40       0        0.40    124
      ~  Omitted ~
 18   8.00    4522        1.70      0
 19   9.00    1923        1.80      0
 20  10.00     193        1.90      0
 21  20.00      33        2.00      0

minimum element height                   =      1.6518785
   element number                         =            675
   inv. element height aspect ratio       =      0.1266485
   minimum dihedral angle                 =      9.2923577
   maximum dihedral angle                 =    164.2051316
maximum element height                    =     11.6131432
average element height                    =      6.8353189
minimum element height ratio to local size =      0.2023130
maximum element height ratio to local size =      1.4223138
average element height ratio to local size =      0.8371522

write chk.wrl >> mati_in0102cs_chk.wrl                  ← File name of VRML output
write har.wrl >> mati_in0102cs_har.wrl                  ← File name of VRML output

elapsed time = 2.050 sec
END advtmesh9e
```