

How to use Mefisto

- demo example : Poisson equation

Author : Lung Sheng Chien (簡龍昇)

Department of Mathematics, Tsing Hua university

Email : d947207@oz.nthu.edu.tw

Tel : 03-5715131-33120

Fax : 03-5723888

NCTS invites professor Alain Perronnet¹ to give a short course with topic : **Finite Element Computation with the Mefisto Software** on 9/16 (Saturday) and 9/17 (Sunday).

On Sunday afternoon, professor Chin-Tien Wu (吳金典),² give us a small example to study mefisto. The Author here just integrates the procedures on that afternoon and is trying to get rid of the weed and keep the flower of the leek. The organization of this paper is as follows:

One needs to setup mefisto in cygwin+winxp or linux environment. Then demo example, elliptic problem, is introduced in chapter 1. We start mefisto in chapter 2, the reader can follow the figures and commands step by step to complete geometry setting of the demo problem. In chapter 3, we use Thermal solver provided by mefisto to solve our demo example. Finally some Q&A are listed in Chapter 4. Besides we list command of MAILLER in Appendix A, command of THERMICER in Appendix B, where the commands in this demo example would be highlighted, this may help reader to know the hierarchy of commands in mefisto. We also write some comments for ASCII files generated by MAILLER, for example, file **xyznsef.v.solid3** for volume information, file **xyznsef.s.dirb1** for surface information and file **xyznpef.OB** for object information (how to report these files would be cover in later chapter, we don't mention here), if one needs to mesh information for Finite Element computation, then Appendix C, D, E are suitable for you. (Thanks for professor Perronnet's help to interpret content of ASCII files as shown in Appendix C,D,E).

Remark 1: author collects the commands in this demo example as a file

¹ Laboratoire J. L. Lions, Universite de Pairs VI, FRANCE
[E-mail:perronnet@ann.jussieu.fr](mailto:perronnet@ann.jussieu.fr)

² Department of computer science, National Taiwan Ocean university,
TEL: 02-2462-2192-6649
Email : ctwu@cs.ntou.edu.tw

cube_cylinder.mail, which can be read by MAILLER.

Chapter 0 setup environment

At URL: <http://www.ann.jussieu.fr/~perronnet/pub/>, you can download new version mefisto with source code, mefistocygwin.tgz and mefistolinux.tgz. The author use Cygwin to demonstrate the example, Poisson equation, the following are configuration of mefisto in my machine.

```
$ echo $MEFISTO
```

```
$ /usr/local/mefisto
```

```
$ echo $MEFISTOX
```

```
$ /home/LungShengChien/mefistox
```

```
$ cd /usr/local
```

```
$ ls -l
```

```
  /usr/local/mefisto → /home/LungShengChien/mefisto
```

The location of **\$MEFISTOX** is important since all projects created by mefisto are put into that directory.

Remark 2: if someone cannot start x-term in cygwin, then author strong recommend that you should install mefisto in a Linux machine.

Chapter 1 Problem formulation

Consider the Poisson equation $-10^{-1}\Delta u = 1$ on a solid cube with a hollow cylinder with radius 1, see Figure 1. Here $u = u(x, y, z)$ is a distribution of

temperature and Laplace operator $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$.

$\Omega = \{(x, y, z) : x^2 + y^2 > 1, |x| < 10, |y| < 10, |z| < 10\}$ and boundary of Ω is composite of 5 surface, denoted by $\partial\Omega = T_1 \cup T_2 \cup T_3 \cup T_4 \cup T_5$.

$$T_1 = \{(x, y, z) : y = 10, |x| \leq 10, |z| \leq 10\}$$

$$T_2 = \{(x, y, z) : y = -10, |x| \leq 10, |z| \leq 10\}$$

$$T_3 = \{(x, y, z) : x^2 + y^2 \geq 1, |x| \leq 10, |y| \leq 10, z = 10\}$$

$$T_4 = \{(x, y, z) : x^2 + y^2 \geq 1, |x| \leq 10, |y| \leq 10, z = -10\}$$

$T_5 = \{(x, y, z) : x^2 + y^2 = 1, |z| \leq 10\}$ is a surface of the cylinder inside cube.

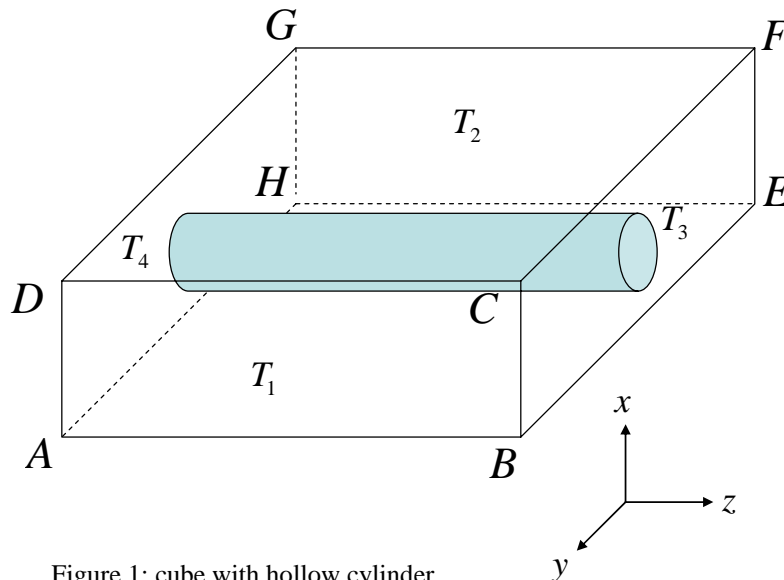


Figure 1: cube with hollow cylinder

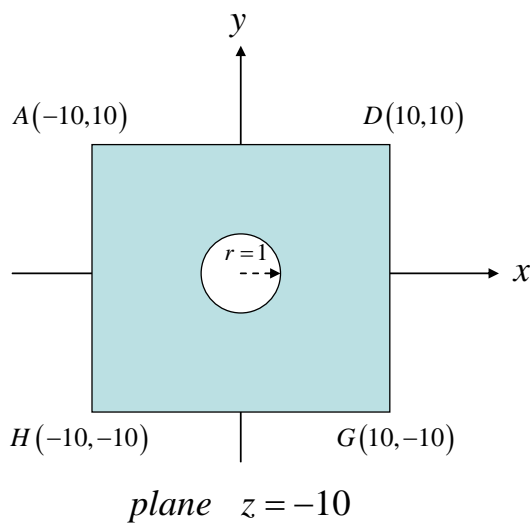


Figure 2: surface T_4 of Figure 1. Note that the z -component of 4 corner points, A, D, G, H are -10.

We set boundary condition on $\partial\Omega = T_1 \cup T_2 \cup T_3 \cup T_4 \cup T_5$ as follows:

Dirichlet boundary condition : $u|_{T_1} = 38^{\circ}C$, $u|_{T_5} = 0^{\circ}C$ and Neumann boundary

condition : $\frac{\partial u}{\partial n}|_{T_1} = 0$ (heat would be reflective).

In order to solve this problem, we have some procedures

1. Describe domain Ω , boundary $\partial\Omega = T_1 \cup T_2 \cup T_3 \cup T_4 \cup T_5$ and tetrahedron triangulation of Ω by MAILLER.

- Specify boundary condition of $\partial\Omega = T_1 \cup T_2 \cup T_3 \cup T_4 \cup T_5$ in THERMICER and compute the Poisson equation with PCG.

Chapter 2 Geometry setting

There are several methods to describe Ω , here our strategy is to define T_4 on x-y plane ($z=0$) and extrude T_4 to positive z-direction with vector (0,0,20) to form a object which is a solid cube with a hollow cylinder, then translate the object with vector (0,0,-10) back.

Open cygwin shell, at prompt symbol \$, start x-term by typing \$startx.

Step 1: create a new project "t1"

Type command \$ **INITIER** and input **t1**, see Figure 3. Mefisto would create a directory named t1 in directory \$MEFISTO.

```

$ INITIER

=====
MEFISTO-INITIER: INIT a PROJECT on MICROSOFT CYGWIN PC
=====

Present state of PC memory
total      used      free      shared      buffers      cached
Mem:      2097151      0      2097151      0      0
-/+ buffers/cache:      0      2097151
Swap:     1572864      60928      1511936

Free spaces on disks
Filesystem      1K-blocks      Used Available Use% Mounted on
C:\cygwin\bin      30716248      15067588      15648660      50% /usr/bin
C:\cygwin\lib      30716248      15067588      15648660      50% /usr/lib
C:\cygwin          30716248      15067588      15648660      50% /
c:                 30716248      15067588      15648660      50% /cygdrive/c
d:                 35840980      12777576      23063404      36% /cygdrive/d
e:                 24579416      5397164      19192252      22% /cygdrive/e
f:                 25109560      12382496      12727064      50% /cygdrive/f
g:                 61440592      25416116      36024476      42% /cygdrive/g
i:                 62798084      4845456      57952628      8% /cygdrive/i

Present Mefisto load modules
total 11316
-rwxr-xr-x 1 LungShengChien None 2531841 Sep 14 12:47 ppelas.exe
-rwxr-xr-x 1 LungShengChien None 2322880 Sep 14 12:47 ppflui.exe
-rwxr-xr-x 1 LungShengChien None 262705 Sep 14 12:47 ppinit.exe
-rwxr-xr-x 1 LungShengChien None 2386285 Sep 14 12:47 pppmail.exe
-rwxr-xr-x 1 LungShengChien None 550698 Sep 14 12:47 pppoba.exe
-rwxr-xr-x 1 LungShengChien None 2934378 Sep 14 12:47 pppther.exe
-rwxr-xr-x 1 LungShengChien None 580128 Sep 14 12:47 pxyz

Present projects in MEFISTOX=/home/LungShengChien/mefistox
total 0
drwxr-xr-x+ 2 LungShengChien None 0 Sep 18 16:04 t1

MEFISTO INITIER: Project (low case) name ?
t1
Files ms10 ms11 ms12 ms13 ms14 and frappe.* are deleted

Execution MEFISTO INITIER in directory /home/LungShengChien/mefistox/t1

-----
| MM MM EEEEEEE FFFFFFF IIII SSSSSS TTTTTT 0000000 |
| MMM MM EE FF II SS TT 00 00 |
| MM M MM EEEEE FFFF II SSSSSS TT 00 00 | FAIT L'EF !
| MM MM EE FF II SS TT 00 00 |
| MM MM EEEEE FF IIII SSSSSS TT 0000000 | Version Aout 2006
-----

Project (low case) name?
t1
DATE      : 18/ 9/ 6 16H 15M 51S
AUTHOR    : LungShengChien
PROJECT   : t1

```

Figure 3: initialize a project t1, input t1 in red rectangle and the directory t1 is created in /home/ LungShengChien/ mefistox/t1 (blue line).

Step 2: invoke module MAILLER

Type command \$ **MAILLER** and input **t1** as project as Figure 4.
After press any key, MAILLER would enter Top level command list, like Figure 5. In mefisto, you can use mouse to select the command you want to execute or just type the number of command after **KEYWORD:** , just remember to add semicolon ‘;’ after the number.

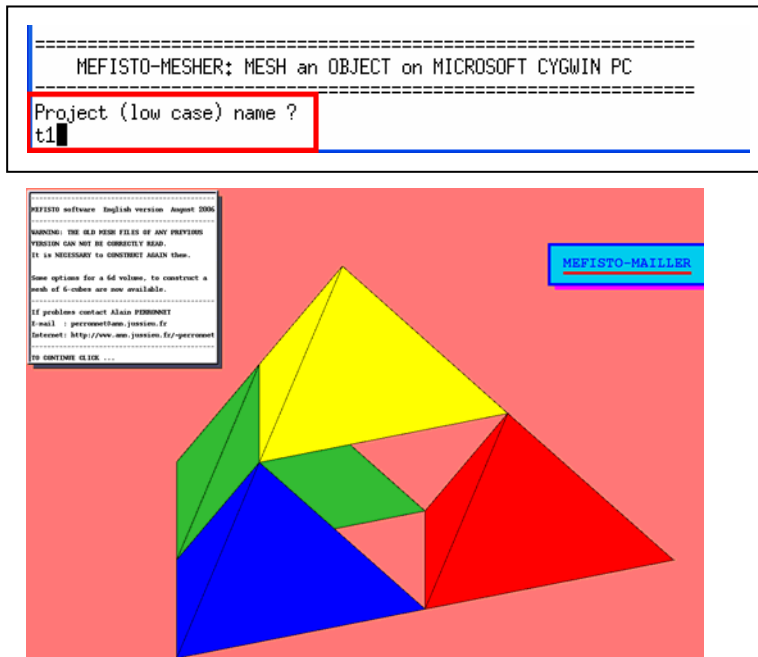


Figure 4: type **t1** in red rectangle to open project t1 and then enter welcome view of MAILLER.

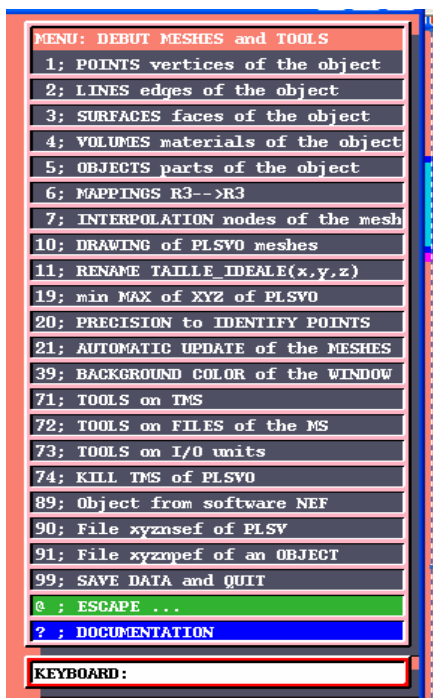


Figure 5: Top level command list, you can use mouse to select the command or type the number of command behind **KEYWORD:** , just remember to type semicolon ‘;’ after the number.

Step 3 : modify the size of view window

When we want to see the object we created. In our problem, the physical size of object is $[-10,10] \times [-10,10] \times [-10,10]$, so we make a bigger one, say

$[-15,15] \times [-15,15] \times [-15,15]$.

(1) Select command **19**; **min MAX of XYZ of PLSV0**

(2) Input **-15** ; as minimum value of x-axis, see Figure 6

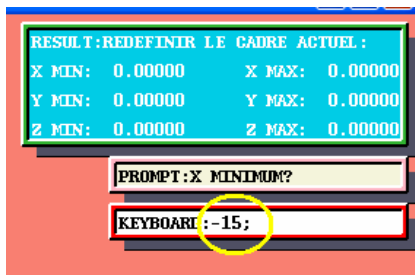


Figure 6: fill minimum of x-axis of viewer, say -15

(3) Input **15**; as maximum value of x-axis, see Figure 7.

(4) Input **-15** ; as minimum value of y-axis, see Figure 8.

(5) Input **15**; as maximum value of y-axis, see Figure 9.

(6) Input **-15** ; as minimum value of z-axis, see Figure 10.

(7) Input **15**; as maximum value of z-axis, see Figure 11.

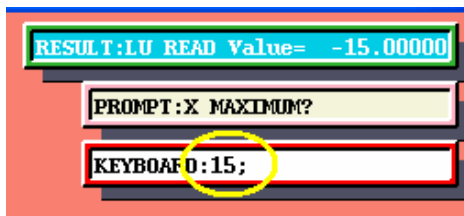


Figure 7: fill maximum of x-axis of viewer, say 15

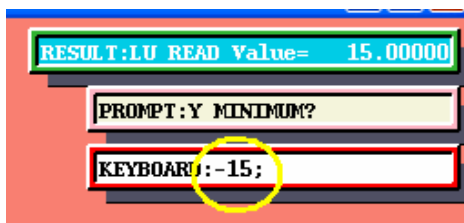


Figure 8: fill minimum of y-axis of viewer, say -15

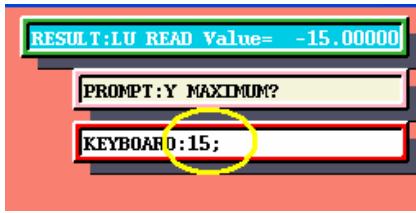


Figure 9: fill maximum of x-axis of viewer, say 15

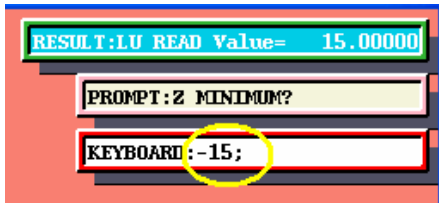


Figure 10: : fill minimum of z-axis of viewer, say -15

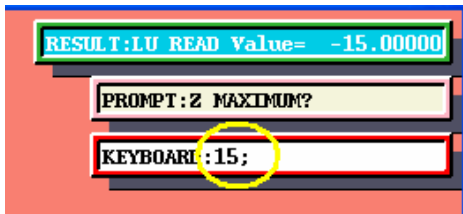


Figure 11: fill maximum of z-axis of viewer, say 15

In order to describe plane T_4 , we have several steps, first setup 6 points $p_1 = H = (-10, -10)$, $p_2 = A = (-10, 10)$, $p_3 = D = (10, 10)$, $p_4 = G = (10, -10)$, $p_5 = (0, 0)$, $p_6 = (1, 0)$. Then 5 lines, \overline{AH} , \overline{AD} , \overline{DG} , \overline{GH} and unit circle C_1 generated by origin $p_5 = (0, 0)$ and a point $p_6 = (1, 0)$ on circumference of C_1 . Third, generate contour $C_2 = \overline{AH} \cup \overline{AD} \cup \overline{DG} \cup \overline{GH}$. Forth, generate a surface enclosed by two contours C_1 and C_2 , then generate triangular mesh.

Step 4: describe 6 points

(1) Select command **1 ; POINTS vertices of the object**, then Figure 12 is shown.

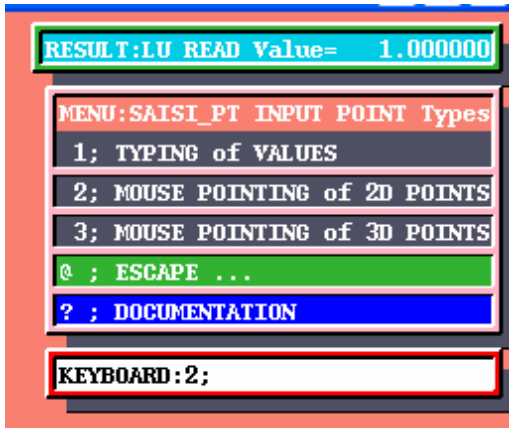


Figure 12: we choose option 2, mouse position of 2D points

(2) Select command **2; MOUSE POINTING of 2D POINTS**, then Figure 13 is shown

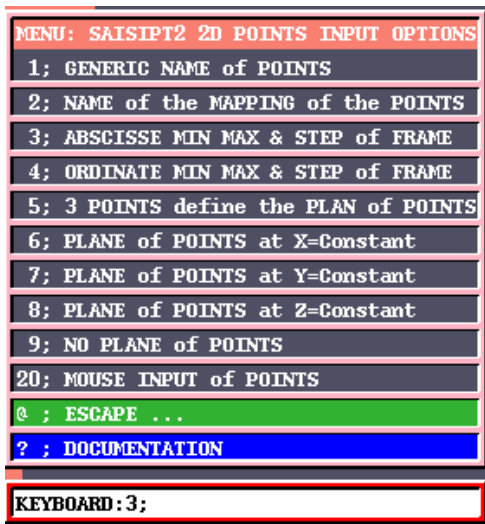


Figure 13: we choose 3; ABSCISSE MIN MAX & STEP OF FRAME to adjust range of x-axis of view window.

In order to click 6 points by using mouse, we must setup frame large enough to enclose all 6 points. We use two commands, one is **ABSCISSE MIN MAX & STEP of FRAME** to determine range of x-axis, the other is **ORDINATE MIN MAX & STEP of FRAME** to determine range of y-axis.

(3) Select **3; ABSCISSE MIN MAX & STEP of FRAME**, then Figure 14 is shown.

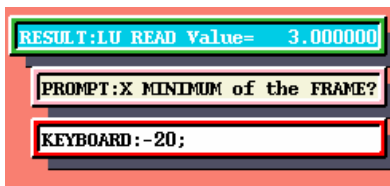


Figure 14: determine minimum of x-axis in the frame. Here we use -20

(4) Input -20; then Figure 15 is shown.

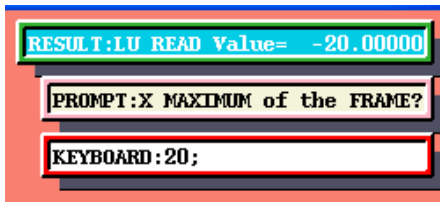


Figure 15: determine maximum of x-axis in the frame. Here we use 20

(5) Input 20; then Figure 16 is shown.

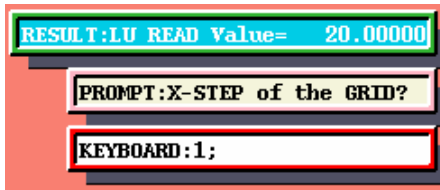


Figure 16: determine step size of x-axis in the frame. Here we use 1

(6) Input 1; then go back to Figure 13, we select command **4; ORDINATE MIN MAX & STEP of FRAME**, then Figure 17 is shown.

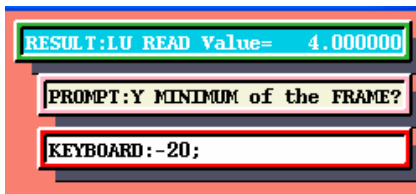


Figure 17: determine minimum of y-axis in the frame. Here we use -20

(7) Input -20; then Figure 18 is shown.

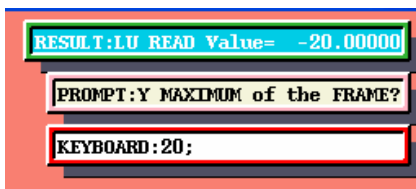


Figure 18: determine maximum of y-axis in the frame. Here we use 20

(8) Input 20; then Figure 19 is shown,.



Figure 19: determine step size of y-axis in the frame. Here we use 1

(9) Input 1; then then go back to Figure 13, we select command **20; MOUSE INPUT of POINTS**, then Figure 20 is shown.

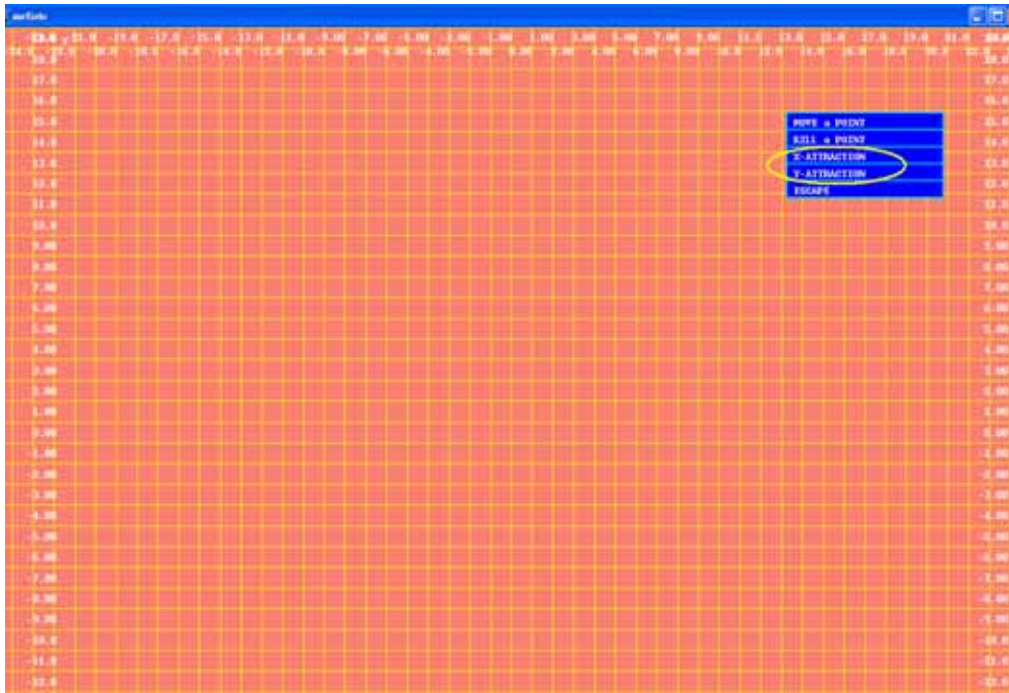


Figure 20: drawing frame, check the option in yellow ellipse, change to **NO X-ATTRACTION** and **NO Y-ATTRACTION** first.

(10) In this frame, use mouse to click options **x-attraction** and **y-attraction** in yellow ellipse, let them become **NO X-ATTRACTION** and **NO Y-ATTRACTION**. see Figure 21. These setting is very important, since our points are of integer, but there must be some error when one uses mouse to click point, Options **NO X-ATTRACTION** and **NO Y-ATTRACTION** would enforce points clicked by mouse to be integer (to nearest grid point).

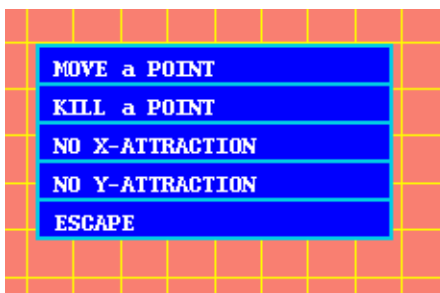


Figure 21: change to no x-attraction and no y-attraction.

(11) use mouse to click points $p_1 = H = (-10, -10)$, $p_2 = A = (-10, 10)$,
 $p_3 = D = (10, 10)$, $p_4 = G = (10, -10)$, $p_5 = (0, 0)$, $p_6 = (1, 0)$, see Figure 22.

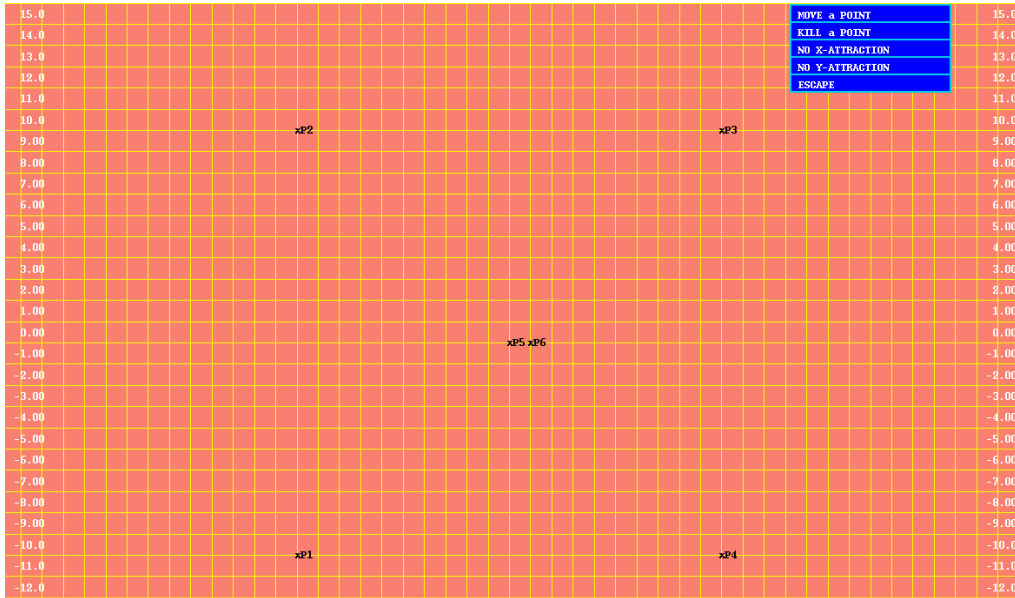


Figure 22: plot 6 points in the frame, the order is indexed by mefisto.

(12) press ESC bottom once, then go back to Top level command list as Figure 5. Then the points are setup completely.

Next we want to construct line segments and contours.

(13) Select command **2; LINES edges of the object**, Figure 23 is shown.

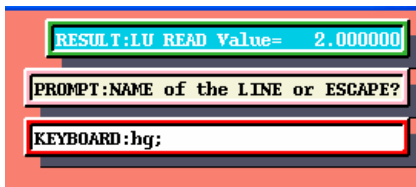


Figure 23: name of line segment \overline{GH} , say hg

We construct \overline{GH} , \overline{DG} , \overline{AD} , \overline{AH} counterclockwisely.

(14) Input **hg;** // name of edge, then Figure 24 is shown.

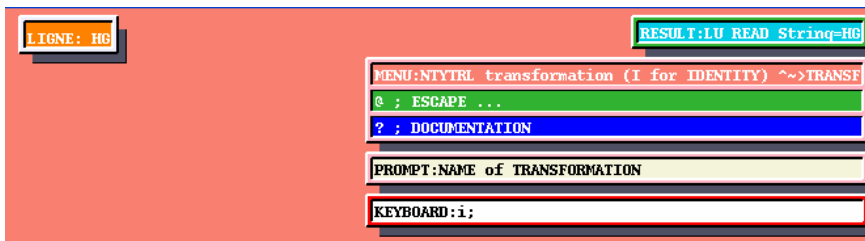


Figure 24: choose Identity transformation

(15) Input **I**; // choose Identity transformation, then Figure 25 is shown.

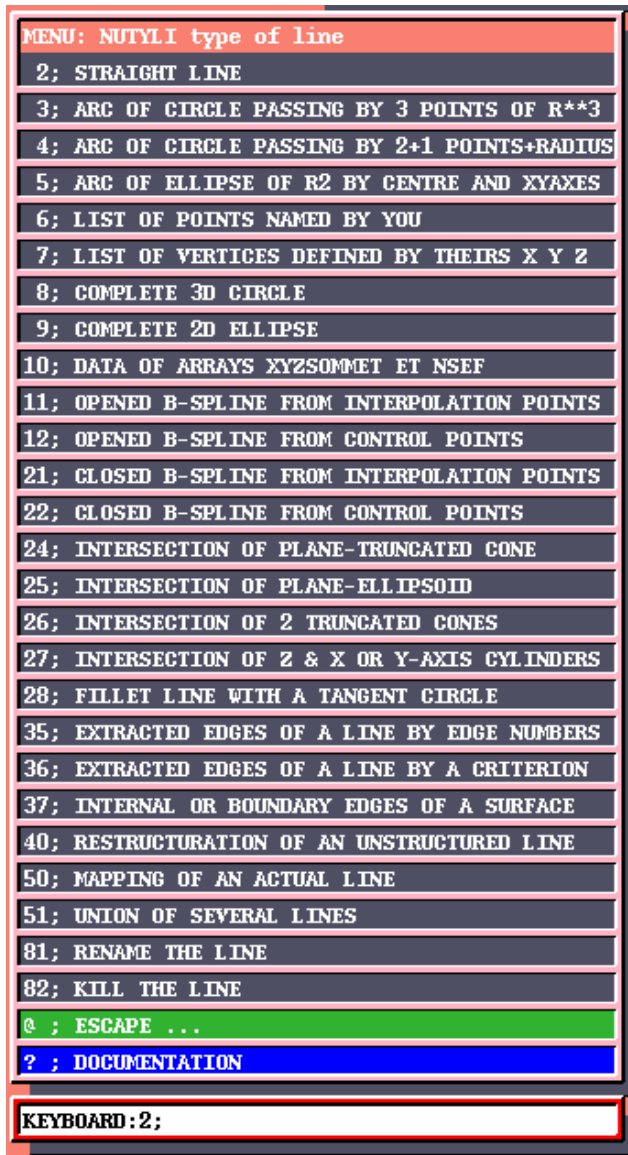


Figure 25: we choose STRAGHT LINE option.

(16) Input **2**; // choose straight line, then Figure 26 is shown.



Figure 26: divide the line to 10 segments

(17) Input **10**; // divide the line into 10 segments, then Figure 27 is shown.

user need only to specify two points to form a line and number of segments of the line, this number would affect the triangulation result.

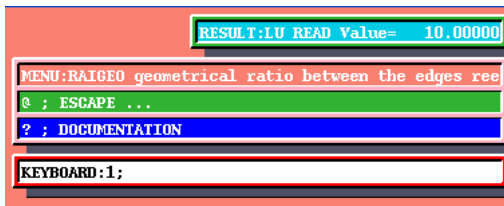
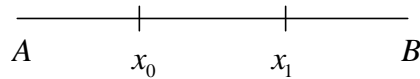
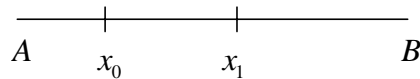


Figure 27: input 1; to choose geometrical ratio as constant.

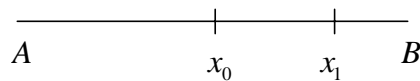
- (18) Input 1; then Figure 28 is shown
 geometrical ratio determine the length of each segment.
 = 1, constant
 >1, ascending
 <1, descending



(a) constant : each segment has the same length



(b) ascending : length of segment increasing geometrically



(c) descending : length of segment decreasing geometrically

(19) From Figure 28, we can choose initial point of the segment by two ways. One is typing name of the point p1; after KEYWORD: the other is to use mouse. Here we use second method. First move the mouse over all frame until 6 points (small white rectangles,) and move mouse to left-bottom one, which is $G = (-10, -10)$, and click left bottom to select it. After you click the mouse, Figure 29 is shown.

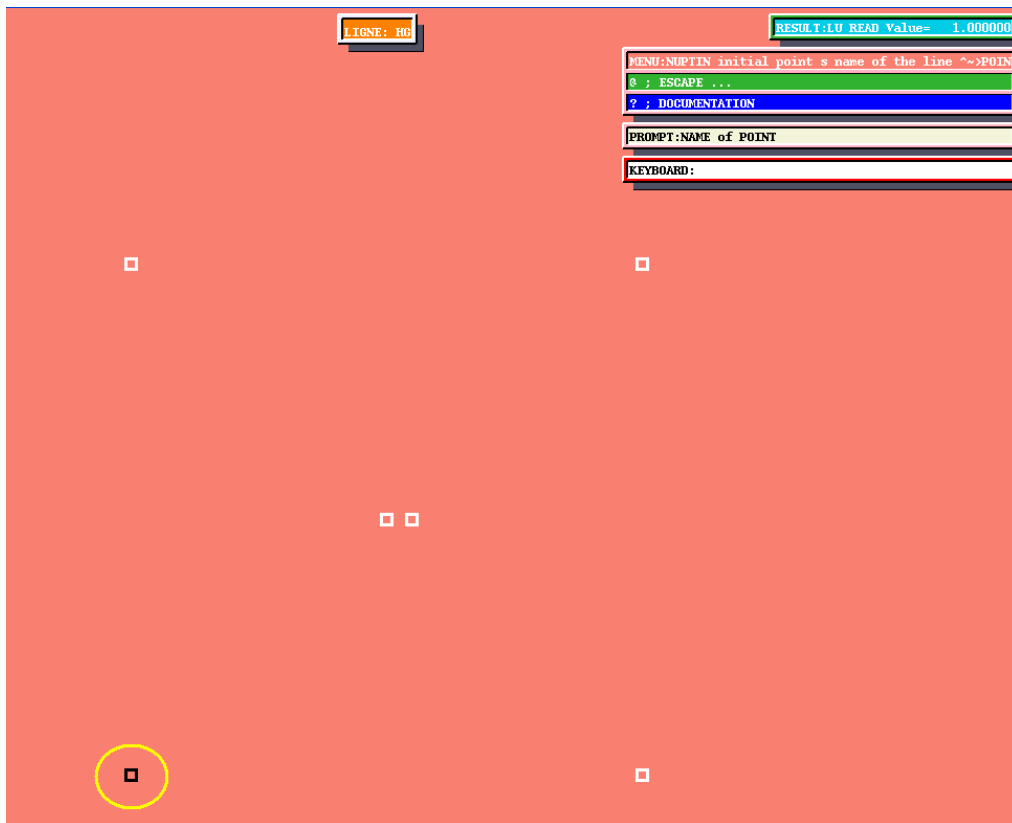


Figure 28: use mouse to choose initial point of the line segment

(20) First, check if text enclosed by blue ellipse in Figure 29 is “final point” or not. If it is, then use mouse to click right-bottom point, which is $H = (10, -10)$. After you click the mouse, the menu of MAILLER would go back to Figure 23.

(21) Repeat procedures (14)~(20), Figure 23 ~ Figure 29 to construct 3 line segments, \overline{DG} , \overline{AD} , \overline{AH} . Name the 3 line segments as gd; da; ah; respectively, configuration of line is the same as hg.

Remark 3: see Figure 30, red line segment is line \overline{GH} in 3D, MAILLER does not erase it when you input the configuration of second line \overline{DG} . Just move the mouse, then 6 points would appear again as small white rectangle, select one that is small black rectangle in Figure 30.

After 4 line segments are configured, the result is shown as Figure 31.

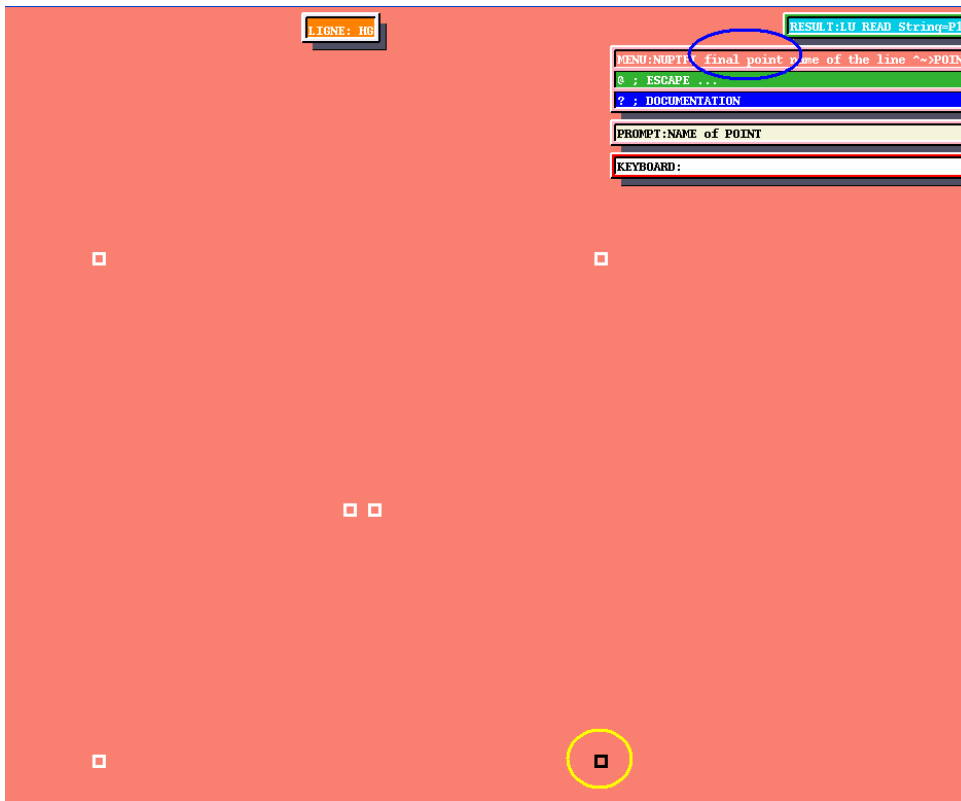


Figure 29: select final point (enclosed by yellow circle) of the line segment. Note that you will see “final point” in blue ellipse.



Figure 30: select initial point of line gd ;

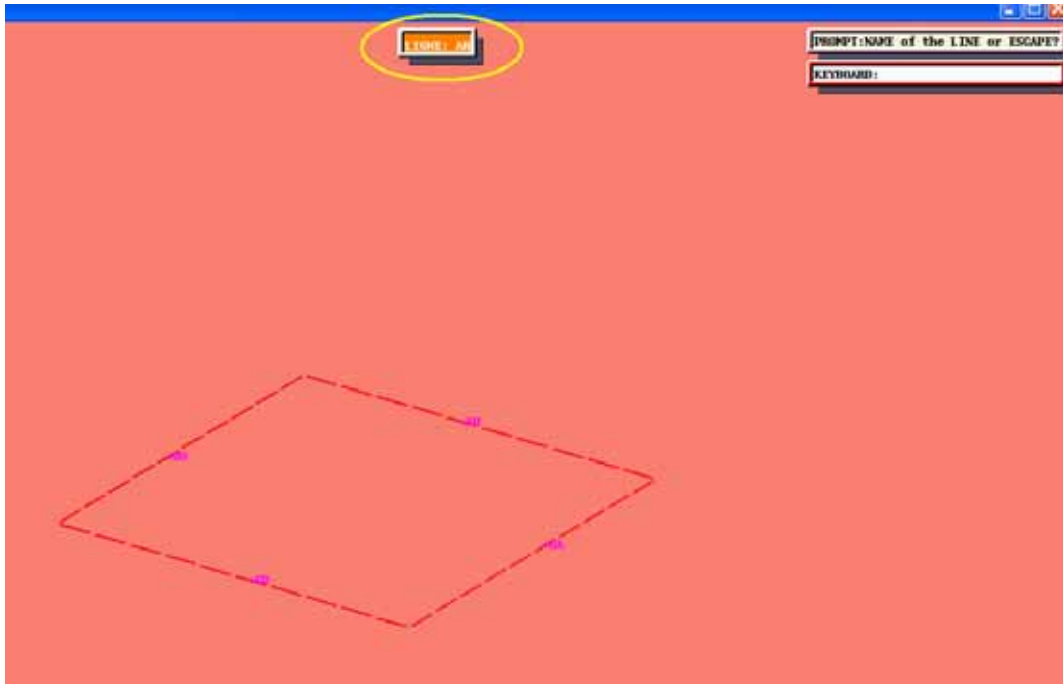


Figure 31: After 4 line segments are configured, they are shown by red lines, you can see each line segment are divided by 10 sub-segments.

(22) Next we want to construct a circle with center $p_5 = (0,0)$ and radius 1, passing through $p_6 = (1,0)$. Set circle's name as **cr1**; (first circle), then Figure 24 is shown.

(23) Input **I**; // choose Identity transformation, then Figure 25 is shown.

(24) Select command **8; COMPLETE 3D CIRCLE**, then Figure 26 is shown.

(25) Input **10**; // divide the line into 10 segments, then Figure 32 is shown.

(26) Input **2**; // circle defined by centre, point on the circle, point of the plane, then Figure 33 is shown.

(27) Input p_5 ; // set $p_5 = (0,0)$ as center of the circle **cr1**, then Figure 34 is shown.

(28) Input p_6 ; // set $p_6 = (1,0)$ be the point on the circle **cr1**. This setting would implicitly determine radius of **cr1**. Then Figure 35 is shown.

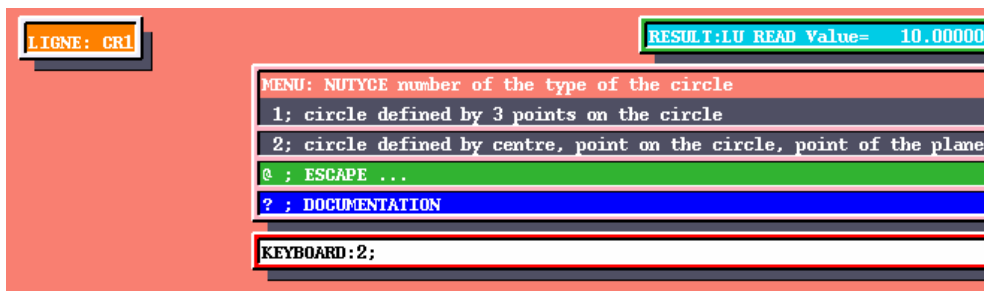


Figure 32: Select option 2;

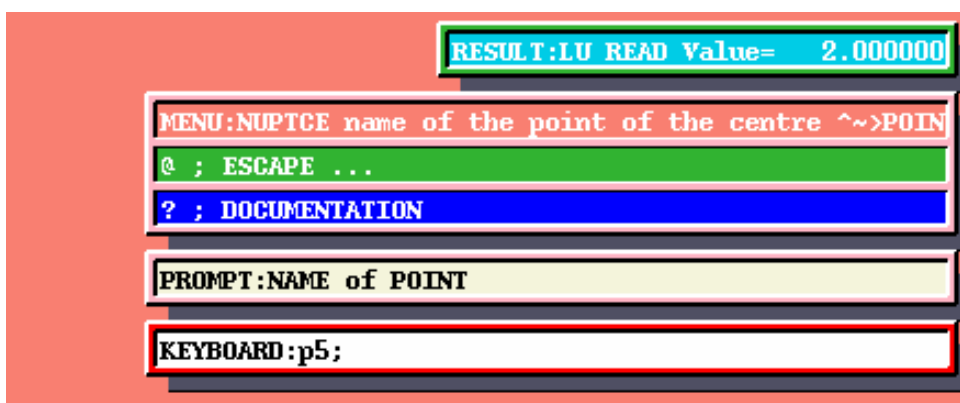


Figure 33: Input p5; // select centre of the circle

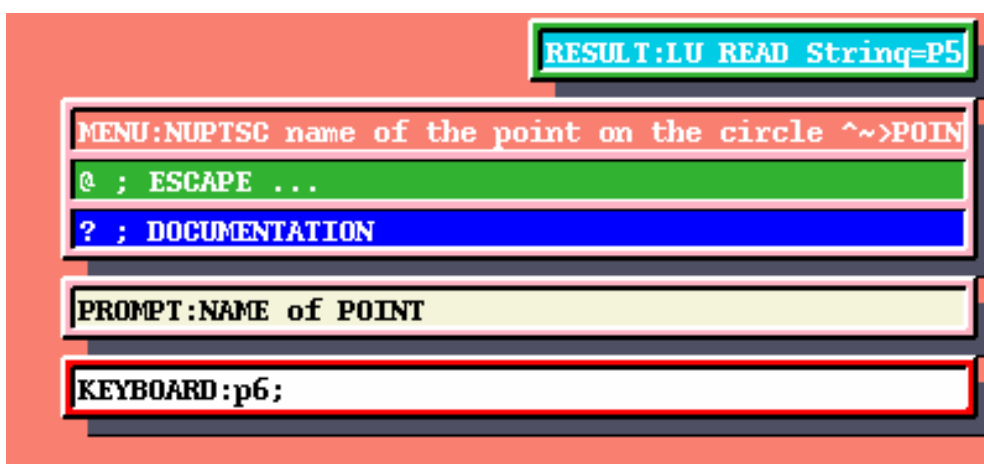


Figure 34: Input p6; // p6 is the point on the circle cr1

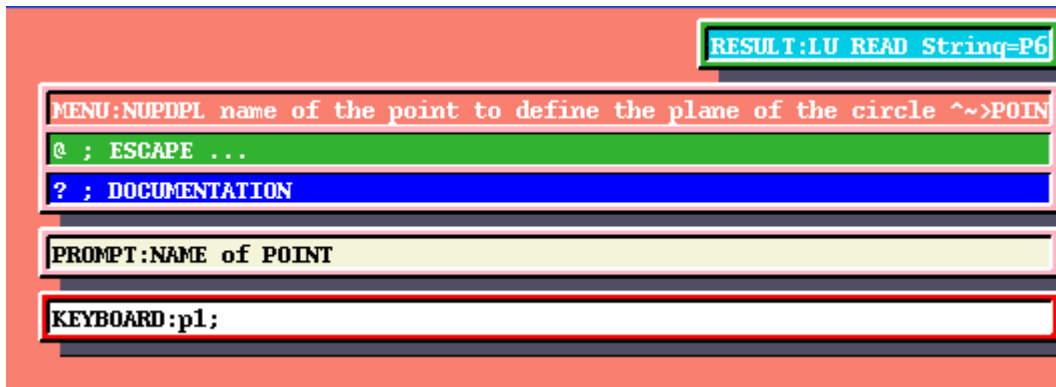


Figure 35: Input $p1$; such that circle $cr1$ falls on the plane generated by $p5, p6$ and $p1$.

(29) Input $p1$; // to select point H as reference point such that circle $cr1$ falls on the plane generated by 3 points H, $p5$ and $p6$. Note that only $p5$ and $p6$ can not determine a circle uniquely. Then circle $cr1$ is constructed successfully, see Figure 36.

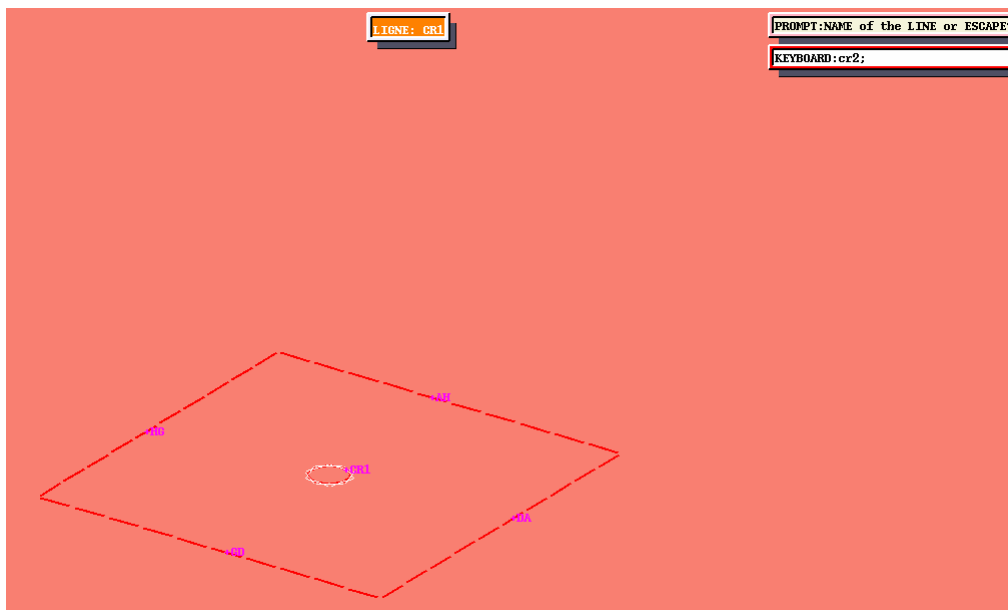


Figure 36: circle $cr1$ is represented by white line.

Finally, we construct a contour (closed loop) $CR_2 = \overline{HG} \cup \overline{GD} \cup \overline{DA} \cup \overline{AH}$

(30) Input $cr2$; // name of the contour, then Figure 24 is shown.

(31) Input I ; // choose Identity transformation, then Figure 25 is shown.

(32) Select command **51; UNION OF SEVERAL LINES**, then Figure 37 is shown.

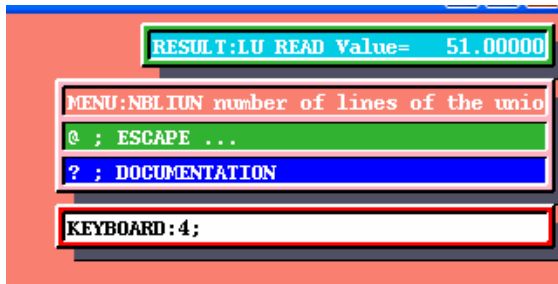


Figure 37: we union 4 line segments.

(33) Input 4; // union 4 line segments, then Figure 38 is shown.



Figure 38: we use mouse to select 4 lines., click 4 smaller rectangles on red line

(34) In Figure 38, using mouse to select 4 lines., click 4 smaller rectangles (white or black) on red line. Then Figure 39

(35) After contour **cr2** is generated (see Figure 39), we cannot see additional information since it overlaps with 4 line segments \overline{GH} , \overline{DG} , \overline{AD} , \overline{AH} . Press ESC to back to Top level command list, Figure 5.



Figure 39: After contour cr2 is generated, we cannot see additional information since it overlaps with 4 line segments \overline{GH} , \overline{DG} , \overline{AD} , \overline{AH} .

Step 5: generate surface T_4

(1) Select command **3 : SURFACES faces of the objects**, then Figure 40 is shown.

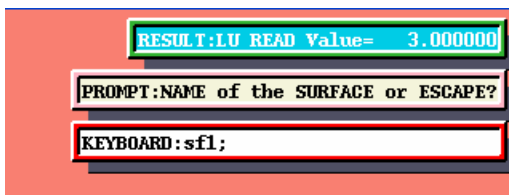


Figure 40: name of surface is **sf1**

(2) Input **sf1**; // first surface, then Figure 24 is shown.

(3) Input **I**; // choose Identity transformation, then Figure 41 is shown.

(4) Select command **9; TRIANGULAITON BETWEEN CLOSED LINES**, then Figure 42 is shown.

(5) Input **2**; // set maximum length of triangle mesh as 2, since for each direction, length of object is 20 and 10 sub-segments, $20/10 = 2$. Then Figure 43 is shown.



Figure 41: select 9, surface is enclosed by two closed curves we have generated.

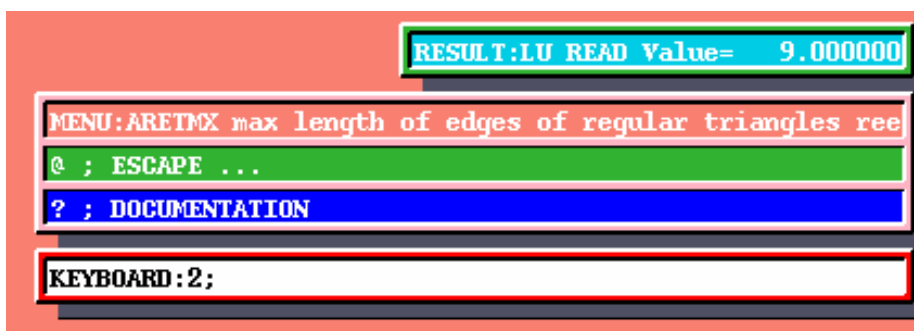


Figure 42: set maximum length of triangle mesh as 2

(6) Input 2; // surface is enclosed by two closed curves, then Figure 44.

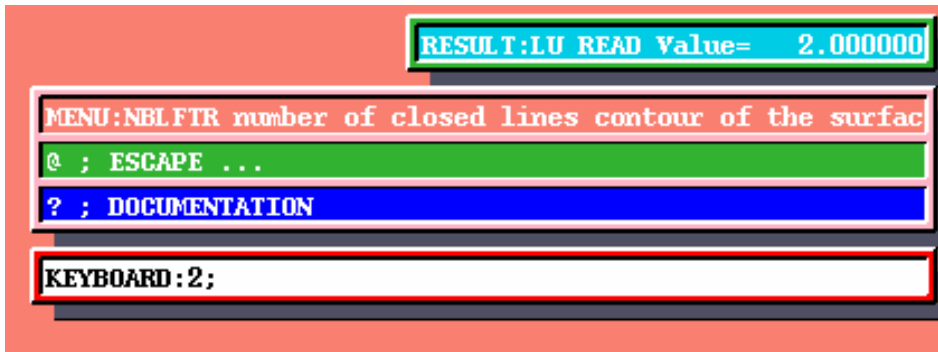


Figure 43: input 2; since surface is enclosed by two closed curves

(7) Input 0; // no internal controlling points, then Figure 45 is shown.

Internal controlling points are set by user, they can be used to control triangulation. Briefly speaking, we can re-define function TAILLE_IDEALE(x,y,z), which affect resolution of mesh near controlling points. Here we don't set any controlling points.

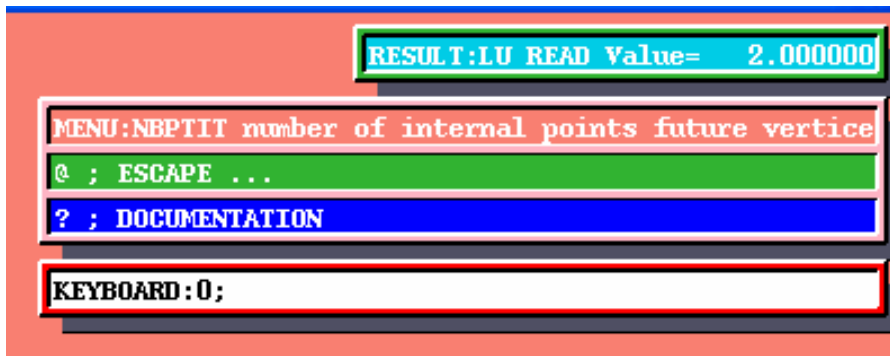


Figure 44: on internal control points, so set 0;

(8) Input cr1 ; [enter] // circle loop and then

Input cr2 ; // rectangle loop

You will see triangular mesh generated as Figure 46.

(9) Press **ESC**, back to Top level command list, see Figure 5.

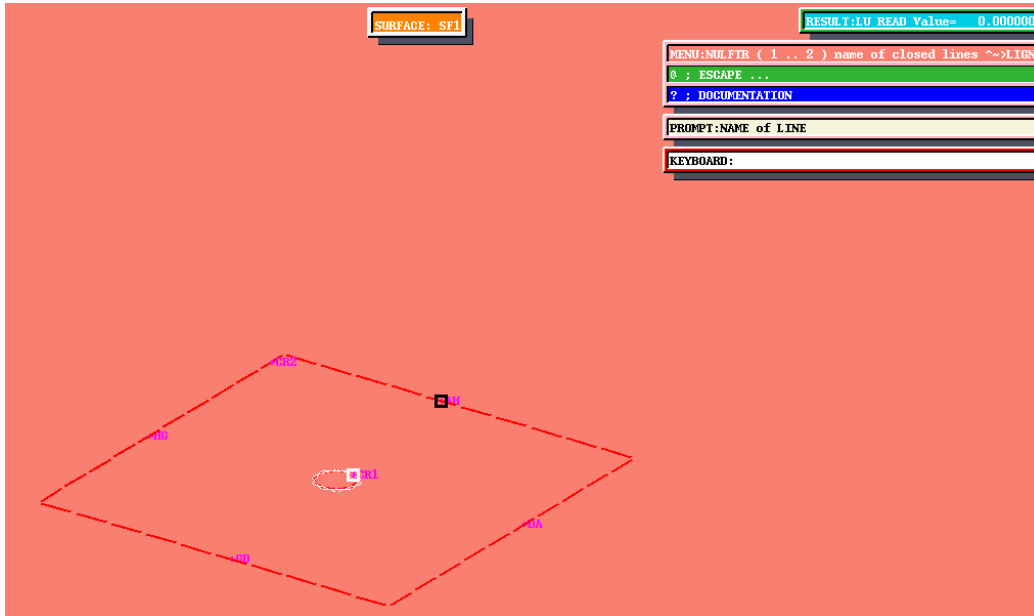


Figure 45: Input **cr1** ; and then **cr2** ;

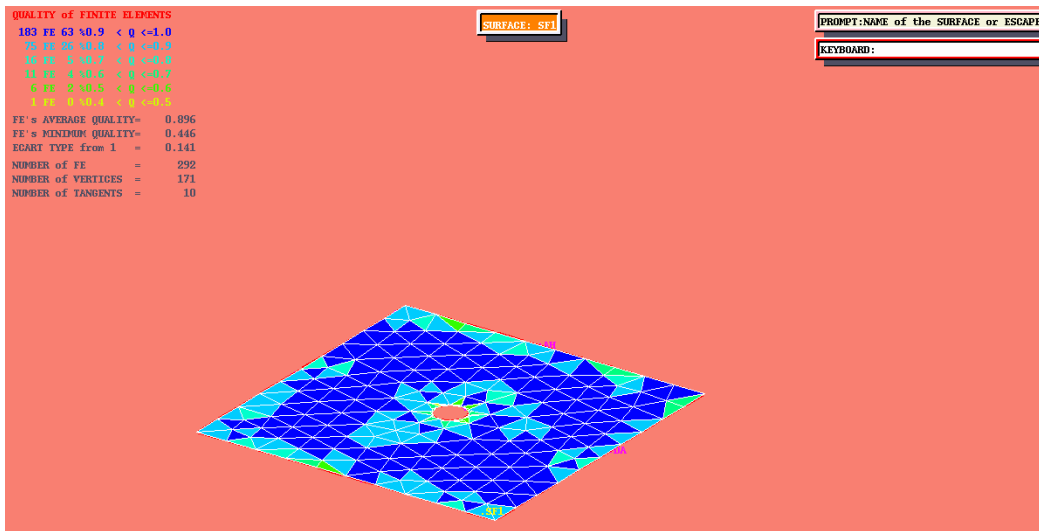


Figure 46: triangular mesh of surface **sf1**

Step 6: extrusion surface T_4 to generate Ω

- (1) Select command **4 : VOLUMES materials of the object**, then Figure 47 is shown.
- (2) Input **solid1**; // first solid, then Figure 24 is shown.

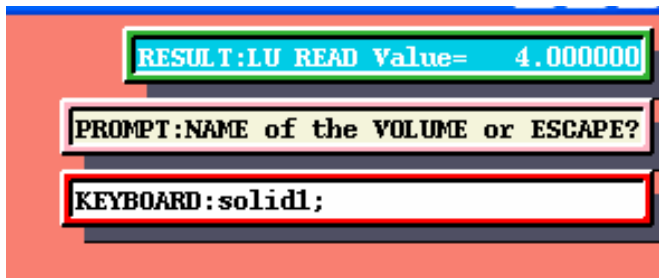


Figure 47: name the volume as **solid1**, means that first solid.

(3) Input **I**; // choose Identity transformation, then Figure 48 is shown.



Figure 48: select 14; slice from a generic surface (extrusion)

(4) Select command **14; SLICES FROM A GENERIC SURFACE (EXTRUSION)**, then Figure 49 is shown.

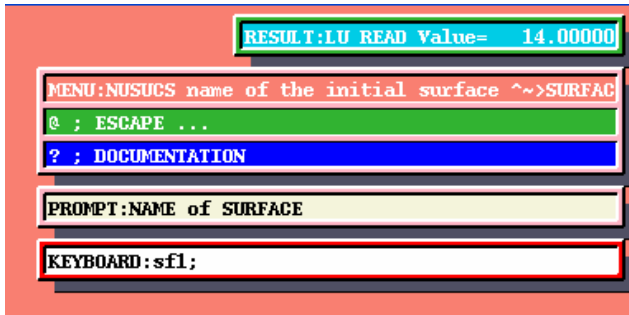


Figure 49: select initial surface which is used to be extended.

(5) Input **sf1;** // initial surface used to be extruded to a volume, then Figure 50 is shown.

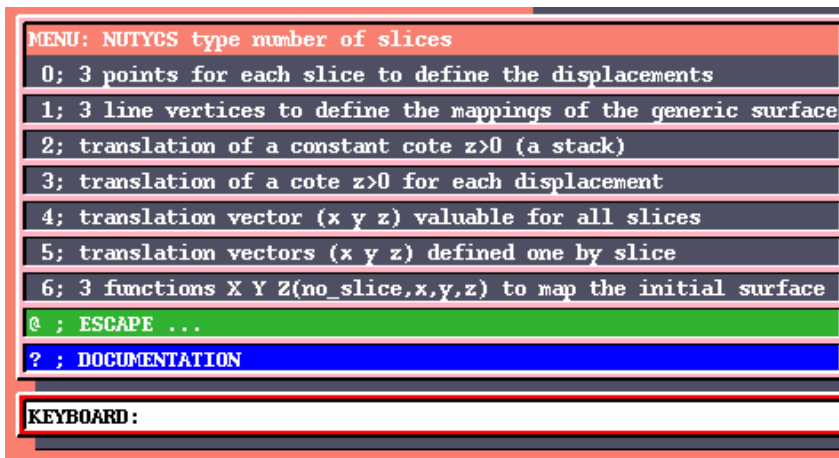


Figure 50: select 2; translation of a constant cote $z > 0$ (a stack)

(6) Select command **2; translation of a constant cote $z > 0$ (a stack)**, then Figure 51 is shown.

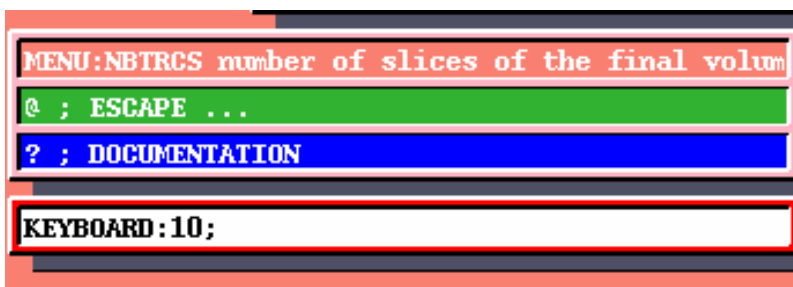


Figure 51: choose 10 slices of the final volume

(7) Input **10;** // 10 slices of the final volume, then Figure 52 is shown.

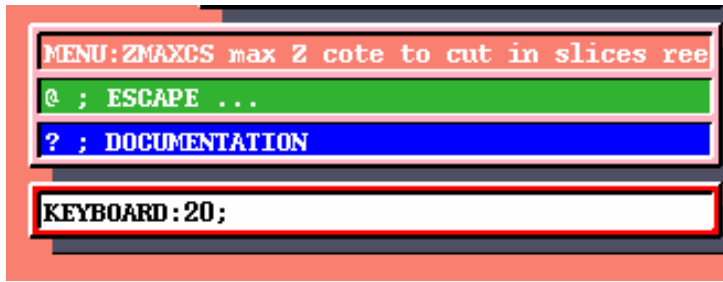


Figure 52: choose 20 as maximum value of z-direction, say length of cube over z-direction is 20.

(8) Input **20;** // maximum value of z-direction, then result is shown in Figure 53.

Our setting for z-direction is $[-10,10]$, but solid1 is configured to be $[0,20]$, later we would define another volume **solid2** which is differ from **solid1** a translation $z = -10$.

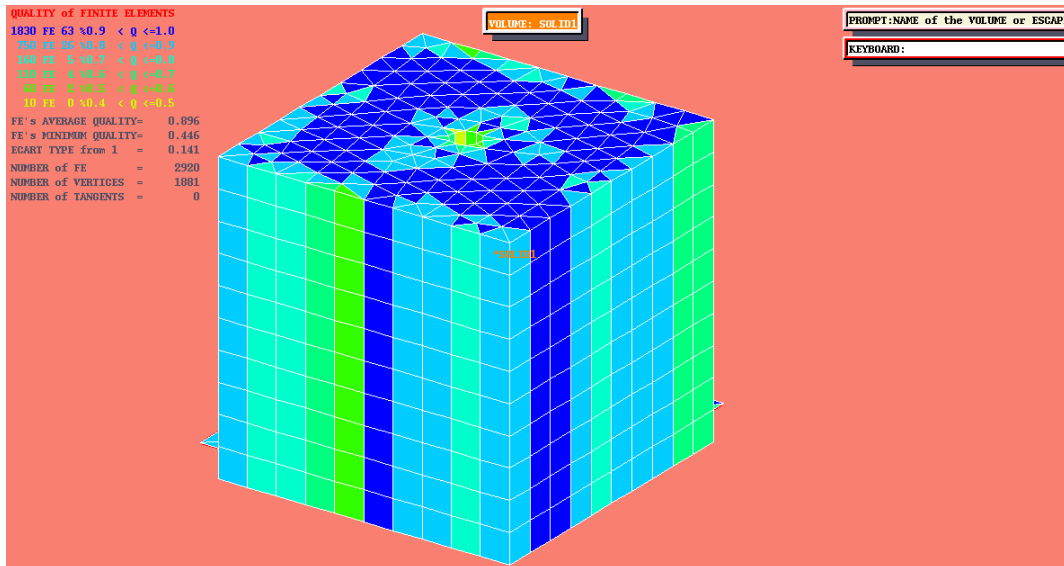


Figure 53: extrusion is complete and each element is pentahedron (not tetrahedron)

(9) Press **ESC**, back to Top level command list, see Figure 5.

Step 7: solid2 = translation $z = -10$ of solid1

First we must define a new transformation, in fact, a translation with vector $(0,0,-10)$, since solid1 has dimension $[-10,10] \times [-10,10] \times [0,20]$, we want to create a solid2 which is translation of solid1 with translation vector $(0,0,-10)$ such that

dimension of solid2 is $[-10,10] \times [-10,10] \times [-10,10]$. Hence we define translation first.

(1) Select command **6 : MAPPINGS $R3 \rightarrow R3$** , then Figure 54 is shown.

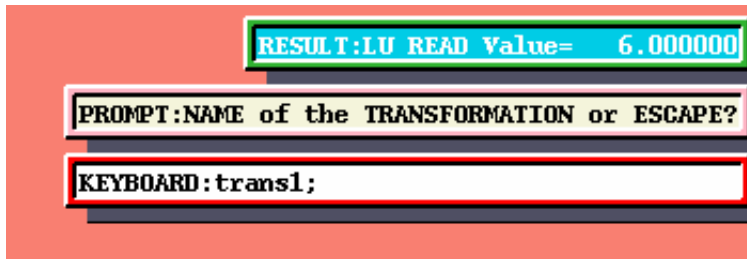


Figure 54: define a transformation, named **trans1**

(2) Input **trans1;** // first transformation, then Figure 55 is shown.

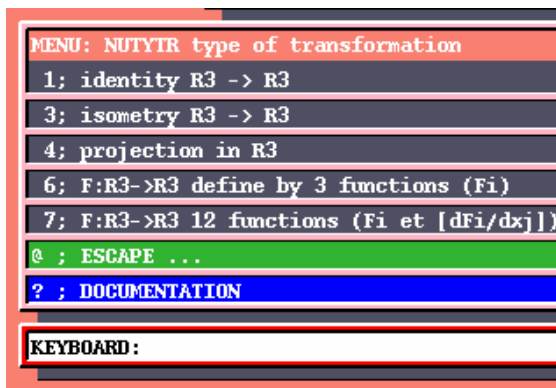


Figure 55: select 3; isometry $R3 \rightarrow R3$

(3) Select command **3; isometry $R3 \rightarrow R3$** , then Figure 56 is shown.

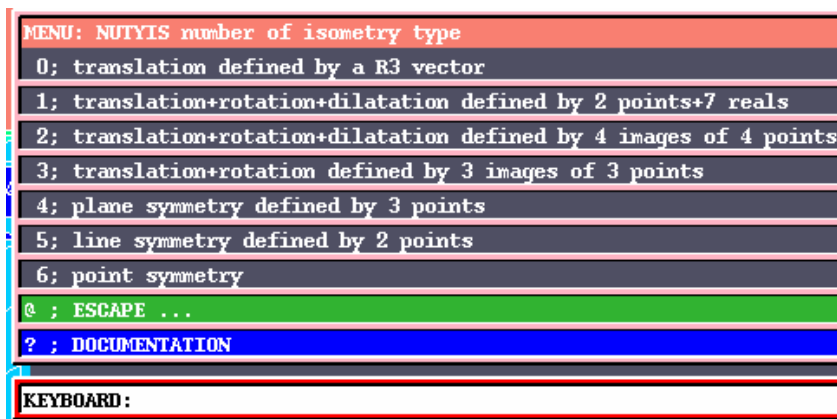


Figure 56: select 0; translation defined by a R3 vector

(4) Select command **0; translation defined by a R3 vector**, then Figure 57 is shown.

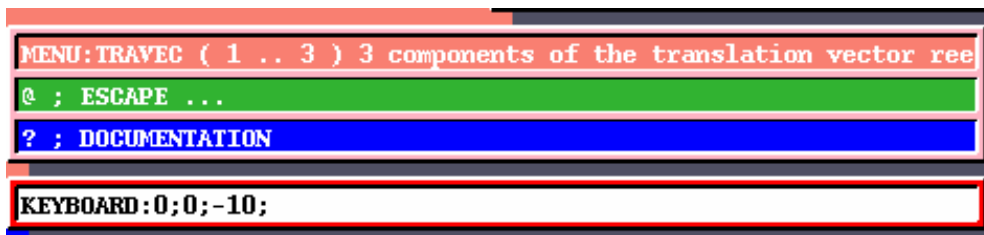


Figure 57: define translation vector (0,0,-10)

(5) Input **0;0;-10;** // translation vector (0,0,-10) , then Figure 54 is shown again, press ESC to top level command list, Figure 5.

(6) Select command **4 : VOLUMES materials of the object**, then Figure 47 is shown..

(7) Input **solid2;** // second solid, then Figure 24 is shown.

(8) Input **trans1;** // choose translation operation we defined in (1)~(5), then Figure 48 is shown.

(9) Select command **50; MAPPING OF A VOLUME**, then Figure 58 is shown.

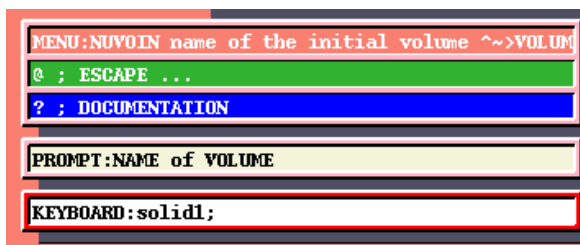


Figure 58: choose **solid1** to do transformation.

(10) Input **solid1;** // choose first volume **solid1**, then **solid2** is shown Figure 59, partially overlap with solid1 since they are differ in translation. Then press **ESC** to Top level command list, Figure 5.

Step 8: solid3 = tetrahedrization of solid2

(1) Select command **4 : VOLUMES materials of the object**, then Figure 47 is shown..

(2) Input **solid3;** // second solid, then Figure 24 is shown.

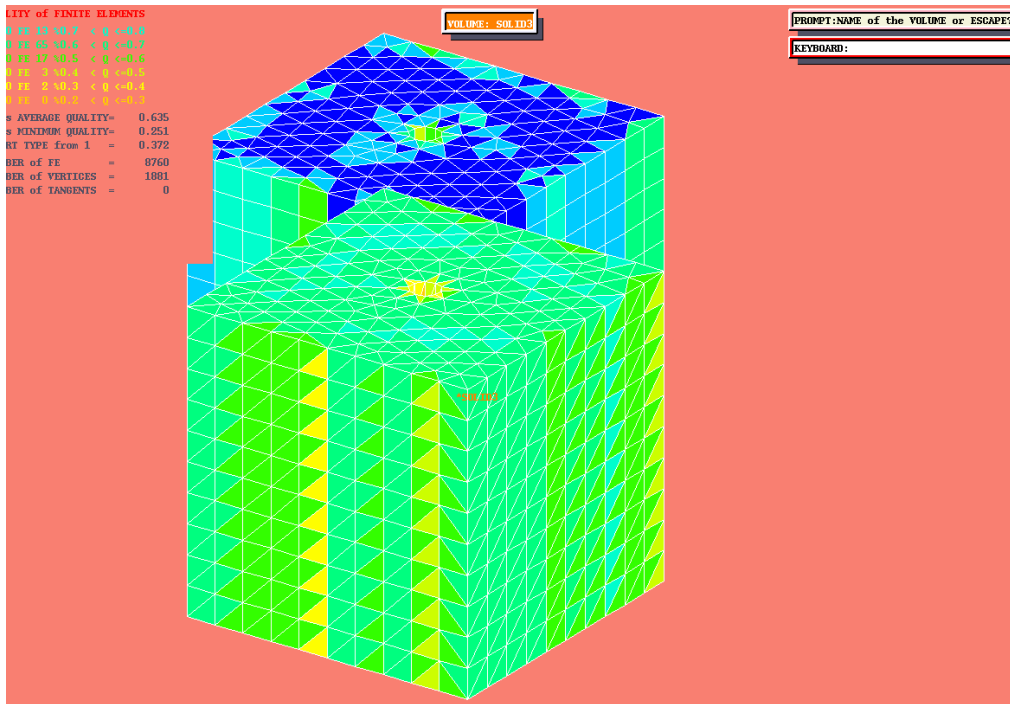


Figure 61: **solid3** is complete with tetrahedron mesh.

Step 9: define boundary surface

Recall boundary condition of our elliptic problem, Dirichlet boundary condition :

$u|_{T_1} = 38^{\circ}C$, $u|_{T_5} = 0^{\circ}C$ and Neumann boundary condition : $\frac{\partial u}{\partial n}|_{T_1} = 0$ (heat would

be reflective). Since Neumann B.C is default in mefisto, we need not to set them, what we need to set is Dirichlet B.C at surface

$$T_1 = \{(x, y, z) : y = 10, |x| \leq 10, |z| \leq 10\} \text{ and}$$

$$T_5 = \{(x, y, z) : x^2 + y^2 = 1, |z| \leq 10\} \text{ which is surface of cylinder inside cube.}$$

The problem is that we don't explicit define surface T_1 and T_5 in our setting.

Remember that we first define T_4 , then extrude the surface to a volume.

(1) Select command **3 : SURFACES faces of the objects**, then Figure 62 is shown.

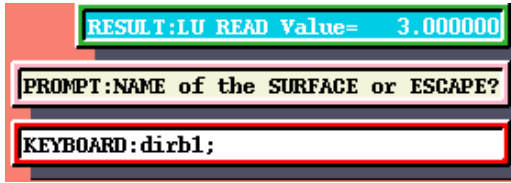


Figure 62: name of T1 is **dirb1**;

- (2) Input **dirb1**; // name of T_1 , then Figure 24 is shown.
- (3) Input **I**; // choose Identity transformation, then Figure 41 is shown.
- (4) Select command **37; EXTRACTED SURFACE FROM FACES OF A VOLUME**, then Figure 63 is shown.

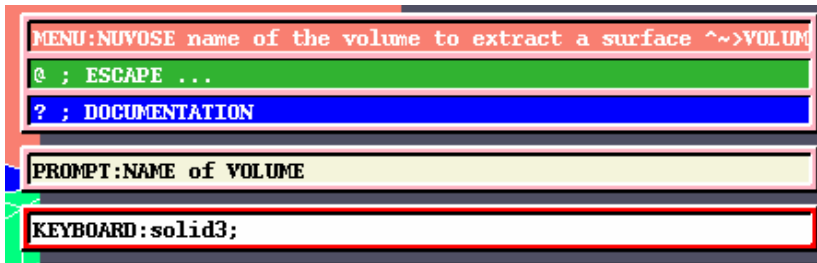


Figure 63: choose volume **solid3** to be extract a surface.

- (5) Input **solid3**; // extract a surface from solid3, then Figure 64 is shown.
- (6) Input **deffunc yp10(x,y,z)**; // function header of fortran-like, then Figure 65 is shown.
- (7) Input **yp10 = y > 9.99** ; // predicate $y > 9.99$, then Figure 66 is shown.

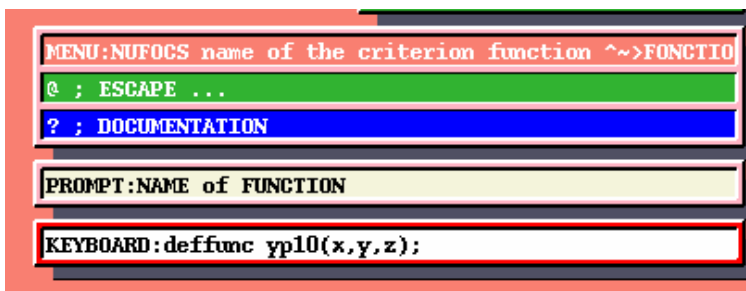


Figure 64: define a function $yp10(x,y,z)$, which is fortran-like

```

MENU:NUFOCS name of the criterion function ^~>FONCTIO
@ ; ESCAPE ...
? ; DOCUMENTATION

READ LINE:DEFFUNC YP10(X,Y,Z);

KEYBOARD:yp10 = y > 9.99 ;

```

Figure 65: predicate

```

MENU:NUFOCS name of the criterion function ^~>FONCTIO
@ ; ESCAPE ...
? ; DOCUMENTATION

READ LINE:DEFFUNC YP10(X,Y,Z);
READ LINE:YP10 = Y > 9.99 ;

KEYBOARD:endfunc ;

```

Figure 66: complete function definition by keyword **endfunc**

- (8) Input **endfunc ;** // complete function definition, then Figure 67 is shown.
- (9) Input **yp10 ;** // use function yp10 as criterion to find surface **dirb1**, then Figure 68 is shown.
- (10) Select command **1; boundary faces**, then Figure 69 is shown.

```

MENU:NUFOCS name of the criterion function ^~>FONCTIO
@ ; ESCAPE ...
? ; DOCUMENTATION

KEYBOARD:yp10;

```

Figure 67: use function **yp10** as criterion to find surface **dirb1** ;

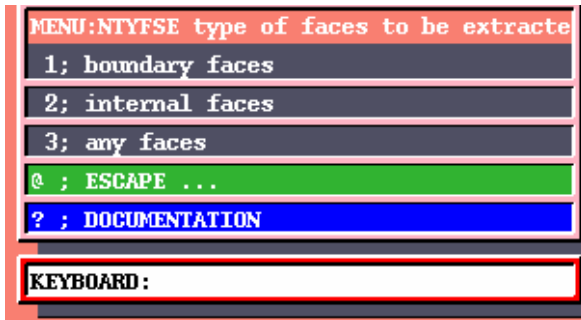


Figure 68: select 1; boundary faces

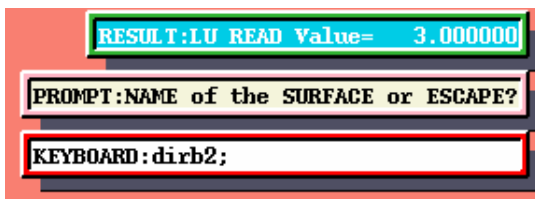


Figure 69: name of T_5 is **dirb2**;

- (11) Input **dirb2**; // name of T_5 , then Figure 24 is shown.
- (12) Input **I**; // choose Identity transformation, then Figure 41 is shown.
- (13) Select command **37; EXTRACTED SURFACE FROM FACES OF A VOLUME**, then Figure 63 is shown.
- (14) Input **solid3**; // extract a surface from solid3, then Figure 70 is shown.
- (15) Input **deffunc cyl(x,y,z)**; // function cyl is first three characters of cylinder, then Figure 71 is shown.
- (16) Input **cyl = x*x + y*y < 1.01**; // predicate function, then Figure 72 is shown.

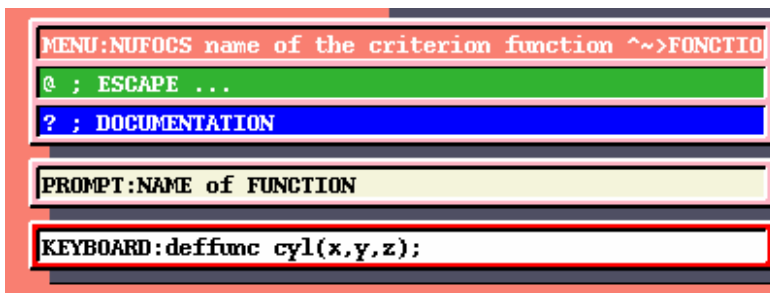


Figure 70: define a function cyl(x,y,z)

```

MENU:NUFOCS name of the criterion function ^~>FONCTIO
@ ; ESCAPE ...
? ; DOCUMENTATION

READ LINE:DEFFUNC CYL(X,Y,Z);

KEYBOARD:cyl = x*x + y*y < 1.01 ;

```

Figure 71: predicate = { $x^2 + y^2 < 1.01$ }

```

MENU:NUFOCS name of the criterion function ^~>FONCTIO
@ ; ESCAPE ...
? ; DOCUMENTATION

READ LINE:DEFFUNC CYL(X,Y,Z);
READ LINE:CYL = X*X + Y*Y < 1.01 ;

KEYBOARD:endfunc ;

```

Figure 72: complete function definition

(17) Input **endfunc** ; // complete function definition, then Figure 73 is shown.

(17) (18) Input **cyl**; // use function cyl as criterion to find surface T_5 . Figure 68 is shown.

(18) Select command **1; boundary faces**, then press **ESC** to Top level command list, Figure 5.

```

MENU:NUFOCS name of the criterion function ^~>FONCTIO
@ ; ESCAPE ...
? ; DOCUMENTATION

KEYBOARD:cyl ;

```

Figure 73: use function cyl as criterion to find surface T_5

Now we want to justify the surfaces we generated.

(19) Select command **10 : DRAWINGS of PLSV0 meshes**, then Figure 74 is shown.

MENU: TRACMAIL Drawings of MESHES
1; Draw ALL the POINTS
2; Draw ALL the LINES
3; Draw ALL the SURFACES without TG
4; Draw ALL the VOLUMES without TG
5; Draw ALL the OBJECTS 1 by 1
6; Draw the POINT
7; Draw the LINE
8; Draw the SURFACE
9; Draw the VOLUME
10; Draw the OBJET
11; SIMPLE VERTICES of a NOT CLOSED LINE
12; SIMPLE EDGES of a NOT CLOSED SURFACE
13; ALL the SURFACES 1 by 1
14; ALL the VOLUMES 1 by 1
21; REDO the PREVIOUS DRAWING
22; GROW by 2 MOUSE clicks at Min MAX
23; REDUCE by 2 MOUSE clicks
24; TRANSLATE by 2 MOUSE clicks
25; 2D WINDOW [Xmin MAX] [Ymin MAX]
26; RE-DEFINITION of EXTREMAL XYZ
27; Total SCENE from EXTREMAL XYZ
30; LONGITUDE & LATITUDE in degrees
31; XYZ of the SEEN POINT & the EYE
32; Half WIDTH & HEIGHT of the SCENE
33; PLANE Behind Ahead of SEEN POINT
34; ZOOM of the Drawing
35; ROTATION around Z-axis in degrees
36; CHANGE the PALETTE of COLORS
37; DEFINE the COLOR of EDGES or FACES
38; BACK at the DEFAULT COLORS
39; BACKGROUND COLOR

Figure 74: select 3; Draw All the SURFACE without TG.

(20) Select command **3; Draw All the SURFACES without TG**, then Figure 75 is shown.



Figure 75: select 90; execute the drawing

(21) Select command **90; EXECUTING the DRAWING**, then Figure 76 is shown. It is easily to see initial surface T_4 and rectangular surface T_1 and cylindrical surface T_5 .

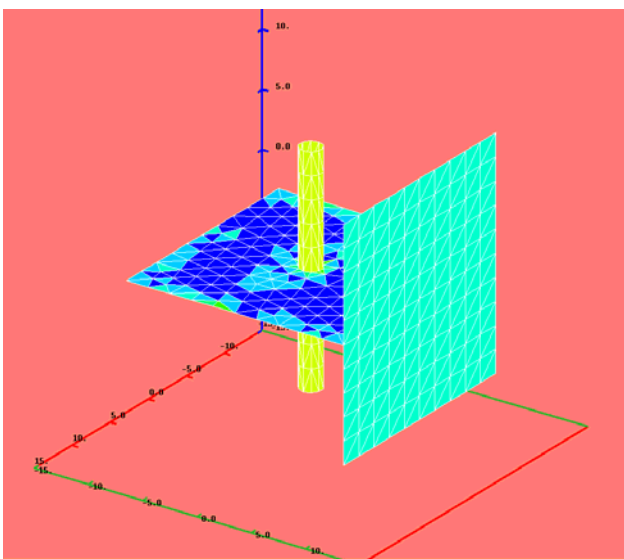


Figure 76: surface T_1 , T_4 and T_5

Step 10: define object = solid3 + boundary surface

In mefisto, boundary condition must be dealt with more carefully. We have extract surface of Dirichlet B.C in step 9. However boundary value is not only set over surface, but also line segment and points. How to deal with that? Recall T_1 is composed by 4 line segments and 4 points, and neighbor surface of T_1 are using Neumann B.C. No two Dirichlet data would be set on line segment or point of T_1 , that is, if we set Dirichlet data on T_1 , then all line segments and points of T_1 would inherit this same value, we don't need to setup them again.

Besides, object need to setup parameters of equation when do computation. There setting are included in volume, hence we need to specify a volume with an object. So in our case, object = 1 volume (solid3) + 2 surface (T_1 (DIRB1) and T_5 (DIRB2)).

(1) Select command **5 : OBJECTS parts of the object**, then Figure 77 is shown.



Figure 77: object name is **ob**;

(2) Input **ob**; // object name, then Figure 78 is shown.

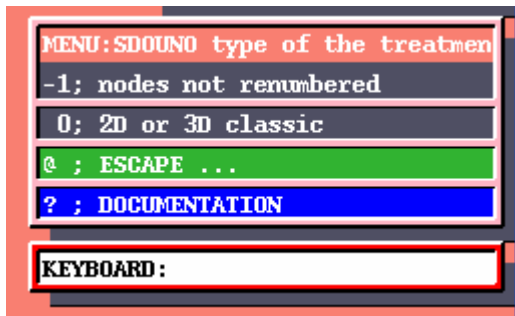


Figure 78: select 0; 2D or 3D classic

(3) Select command **0; 2D or 3D classic**, then Figure 79 is shown.

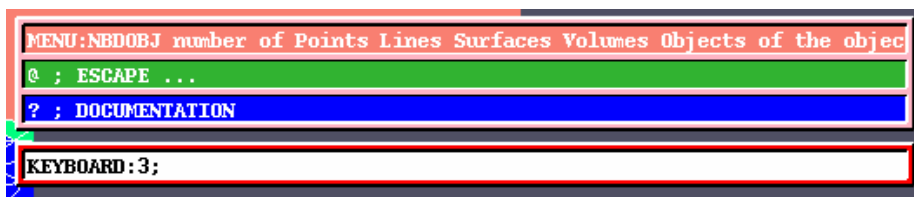


Figure 79: 1 volume **solid3** and 2 surfaces T_1 and T_5 in object **ob**.

(4) Input **3**; // 1 volume **solid3** and 2 surfaces T_1 and T_5 in object **ob**. Then Figure 80 is shown.

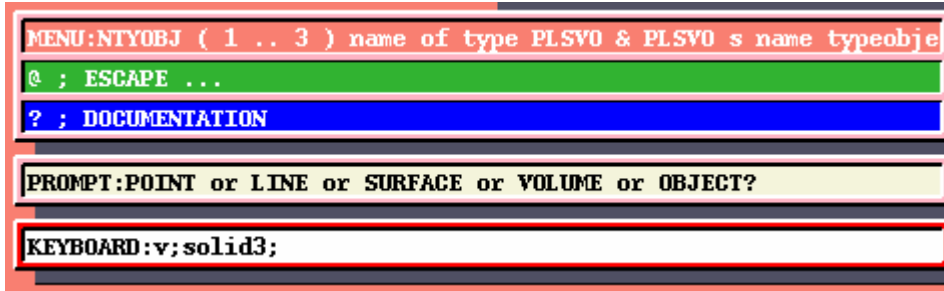


Figure 80: v;solid3; // means volume named solid3

(5) Input **v; solid3**; // means that volume named solid3, then Figure 81 is shown.

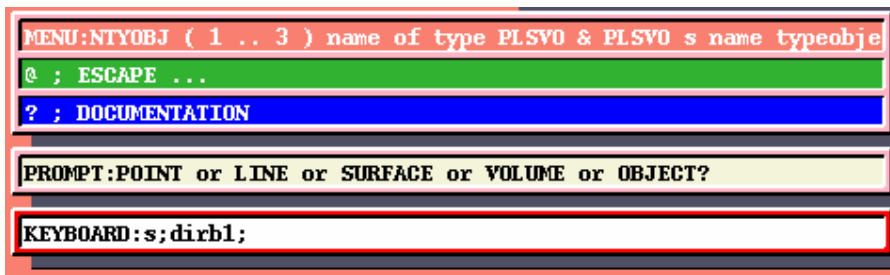


Figure 81: s;dirb1; // mean that surface dirb1

(6) Input **s; dirb1**; // means that surface dirb1, i.e. surface T_1 , then Figure 82 is shown.

(7) Input **s; dirb2**; // means that surface dirb1, i.e. surface T_5 ,

(8) Press **ESC** to top level command list, Figure 5.

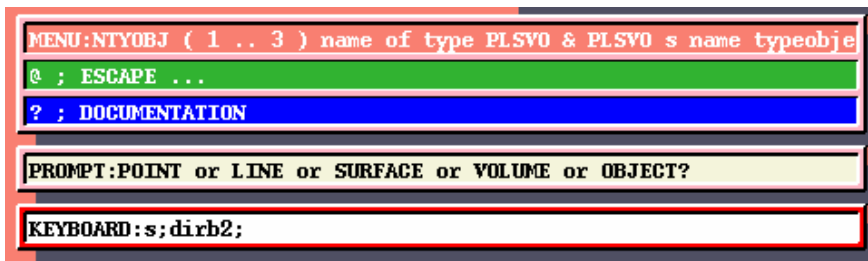


Figure 82: s;dirb2; // mean that surface dirb2

(9) Select command **7 : INTERPOLATION nodes of the mesh**, then Figure 83 is shown.

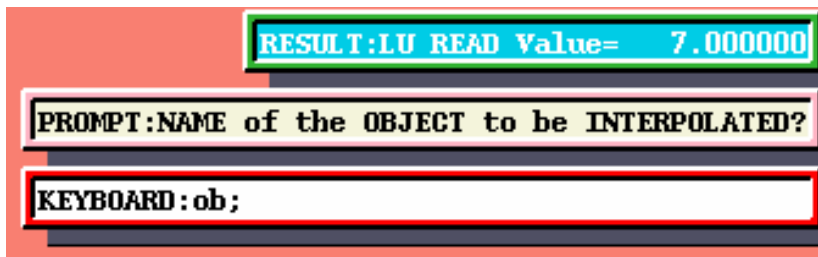


Figure 83: select object “ob” to be interpolated

(10) Input **ob;** // object to be interpolated, then Figure 84 is shown.

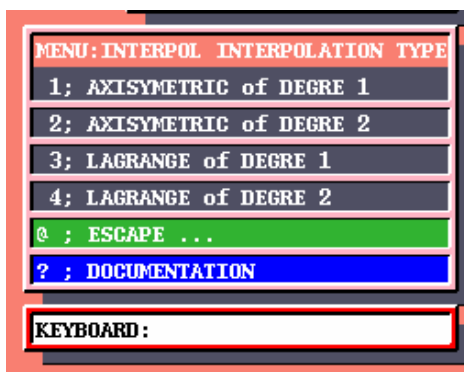


Figure 84: select 4; LAGRANGE of DEGREE 2

(11) Select command **4; LAGRANGE of DEGREE 2**

(12) Press **ESC** to top level command list, Figure 5.

(13) Select command **10 : DRAWINGS of PLSV0 meshes**, then Figure 74 is shown.

(14) Select command **10; Draw the OBJECT**, then Figure 85 is shown.

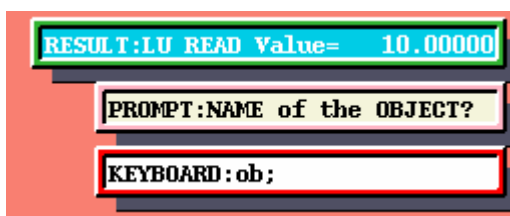


Figure 85: draw object **ob**

(15) Input **ob;** // draw object named ob, then Figure 75 is shown.

(16) Select command **90; EXECUTING the DRAWING**, then Figure 86 is shown. You can see yellow surface named DIRB1, which is T_1 and green surface named DIRB2, which is T_5 . We can translate, rotate and zoom out/in to see the object; we

must ensure that boundary surface is constructed well.

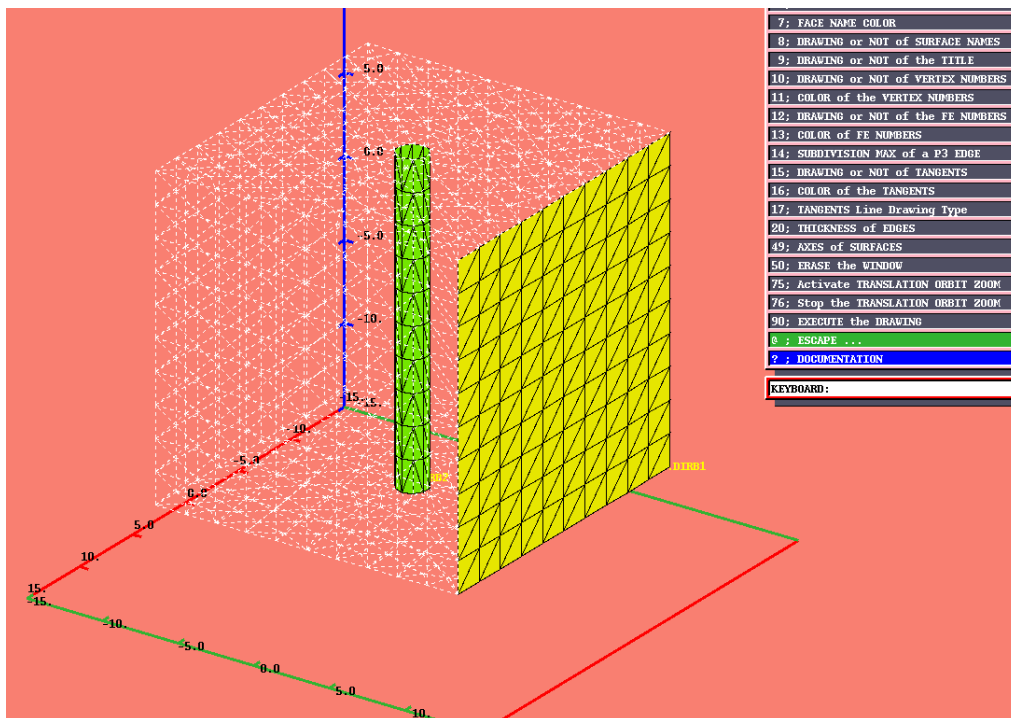


Figure 86: yellow surface is T_1 and green surface is T_5

(17) Select command **75; Active TRANSLATION ORBIT ZOOM**, then

(19) Select command **90; EXECUTING the DRAWING** again, then you can use mouse to do translate, rotate and zoom out/in the object.

- translate: click left bottom of mouse (don't release) and move the mouse
- zoom in/out: click right bottom of mouse (don't release) and move the mouse
- rotate: click wheel (滑鼠滾輪) once, then move the mouse, if you want to stop rotate, just click left bottom of mouse once.

Rotate the object to see top view of the object as Figure 87, there is a closed black line loop on surface of cylinder, it means that surface setting is good.

Remark 4: If the center of cylinder is not $(0,0)$, that is to say $p_5 \neq (0,0)$, then surface of cylinder you construct may not closed.

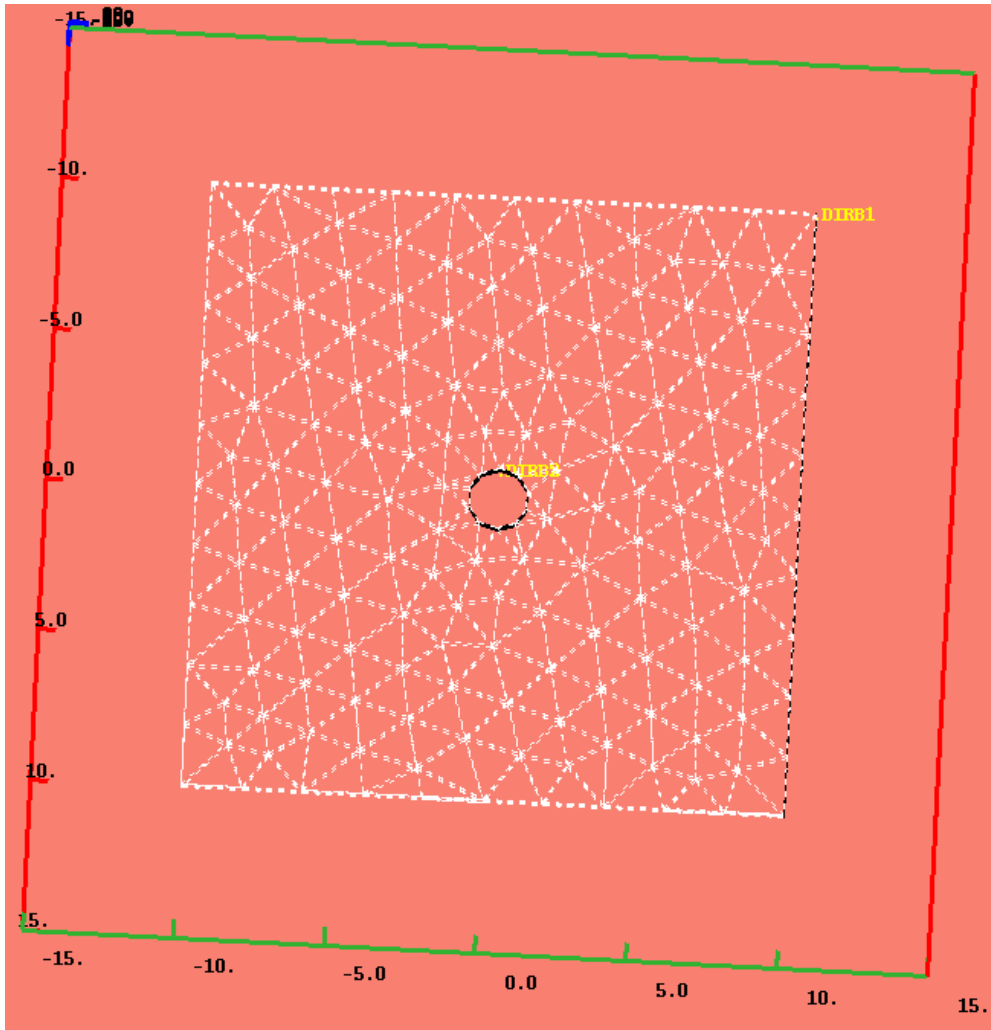


Figure 87: Top view of object, you can see enclose black line on surface of cylinder

So far we have complete geometry setting, next we want to save file

Step 11: save geometry data into ASCII format

(1) In top level command list, Figure 5, select command **90 : File xyznsef of PLSV**, then Figure 88 is shown.

(2) Here you can save point, line, surface or volume independently, we choose **4;VOLUME**, then Figure 89 is shown.



Figure 88: select 4; VOLUME

(3) Input **solid3**; // volume solid3 is correct setting, so we output it. Then Figure 90 is shown.

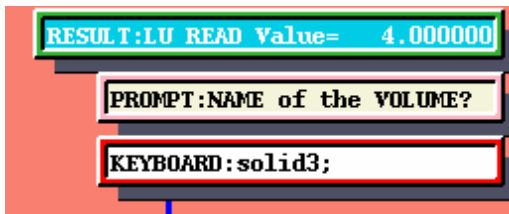


Figure 89: choose volume solid3

(4) Select command **0; WITHOUT TANGENTS**, then file is saved



Figure 90: select 0;WITHOUT TANGENTS

(5) Go into directory \$MEFISTOX/t1, you would see file **xyznsef.v.solid3**, “v” denotes volume and “solid3” denotes name of the volume. See Figure 91.

(6) Similarly we can save object information also, go back to top level command list and select command **91 : File xyznpef of an OBJECT**, then Figure 92 is shown.

(7) Input **ob**; // choose object “ob”, then Figure 90 is shown.

(8) Select command **0; WITHOUT TANGENTS**, then file is saved

```

LungShengChien@math-238dac3744 ~/mefistox/t1
$ echo $MEFISTOX
/home/LungShengChien/mefistox

LungShengChien@math-238dac3744 ~/mefistox/t1
$ cd $MEFISTOX

LungShengChien@math-238dac3744 ~/mefistox
$ ls
t1

LungShengChien@math-238dac3744 ~/mefistox
$ cd t1

LungShengChien@math-238dac3744 ~/mefistox/t1
$ ls
frappe.001  ms10  ms11  ms12  ms13  ms14  xyznsef.v.solid3

```

Figure 91: file xyznsef.v.solid3 is created in \$MEFISTOX/t1

```

RESULT:LU READ Value= 91.00000
PROMPT:NAME of the OBJECT?
KEYBOARD:ob;

```

Figure 92: choose object "ob"

(9) In directory \$MEFISTOX/t1, you would see file **xyznpef.OB**

```

LungShengChien@math-238dac3744 ~/mefistox/t1
$ ls
frappe.001  ms10  ms11  ms12  ms13  ms14  xyznpef.OB  xyznsef.v.solid3

LungShengChien@math-238dac3744 ~/mefistox/t1
$ ls -al
total 3648
drwxr-xr-x+ 2 LungShengChien None          0 Sep 20 14:37 .
drwxr-xr-x+ 3 LungShengChien None          0 Sep 18 15:50 ..
-rw-r--r--  1 LungShengChien None    15360 Sep 18 16:25 frappe.001
-rw-r--r--  1 LungShengChien None   601600 Sep 18 16:15 ms10
-rw-r--r--  1 LungShengChien None    1576 Sep 18 16:15 ms11
-rw-r--r--  1 LungShengChien None    5284 Sep 18 16:15 ms12
-rw-r--r--  1 LungShengChien None   51492 Sep 18 16:15 ms13
-rw-r--r--  1 LungShengChien None         0 Sep 18 16:15 ms14
-rw-r--r--  1 LungShengChien None  2391857 Sep 20 14:37 xyznpef.OB
-rw-r--r--  1 LungShengChien None   656528 Sep 20 14:28 xyznsef.v.solid3

```

Figure 93: file xyznpef.OB has size more than 2MB

(10) In top level command list, select command **99 : SAVE DATA and QUIT** to exist MAILLER.

Chapter 3 Thermo computation

In URL: <http://www.ann.jussieu.fr/~perronnet/mefisto/steadyheat.html>, and URL: <http://www.ann.jussieu.fr/~perronnet/mefisto/unsteadyheat.html> . Professor perronnet write basic idea of mefisto for thermo computation, we extract part of articles and setup relative parameters since our problem is Poisson equation, which can be regarded as steady state of thermal conduction without convection term.

STEADY HEAT TRANSFER
Partial Differential Equations (PDE)

Let

- an opened polyhedral domain Ω in $\mathbb{R}^m (2 \leq m \leq 3)$ and Γ its boundary ;
- $\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_F$ a subdivision of Γ with $measure(\Gamma_D \cup \Gamma_F) > 0$
- find at every point x , the **TEMPERATURE** $\theta(x)$ solution of

$$-\sum_{i=1}^m \frac{\partial}{\partial x_i} \left(\sum_{j=1}^m a_{i,j}(x, \theta) \frac{\partial \theta}{\partial x_j}(x) \right) + \sum_{i=1}^m V_i(x, \theta) \frac{\partial \theta}{\partial x_i}(x) + CT(x, \theta) \theta(x) = F^\Omega(x, \theta) \quad \forall x \in \Omega \quad \forall t \in]T_0, T[$$

Figure 94: equation of steady heat transfer

UNSTEADY HEAT TRANSFER equations

Let

- an opened polyhedral domain Ω in $\mathbb{R}^m (2 \leq m \leq 3)$ and Γ its boundary ;
- $\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_F$, a subdivision of the boundary with $measure(\Gamma_D \cup \Gamma_F) > 0$
- find at every time t , at every point x , the **TEMPERATURE** $\theta(t, x)$ solution of

$$\rho C(t, x, \theta) \frac{\partial \theta}{\partial t}(t, x) - \sum_{i=1}^m \frac{\partial}{\partial x_i} \left(\sum_{j=1}^m a_{i,j}(t, x, \theta) \frac{\partial \theta}{\partial x_j}(t, x) \right) + \sum_{i=1}^m V_i(t, x, \theta) \frac{\partial \theta}{\partial x_i}(t, x) + CT(t, x, \theta) \theta(t, x) = F^\Omega(t, x, \theta) \quad \forall x \in \Omega \quad \forall t \in]T_0, T[$$

with the initial condition

$$\theta(T_0, x) = \theta_0(x) \quad \forall x \in \bar{\Omega}$$

Figure 95: equation of unsteady heat transfer

where

- $\theta(x)$ is the TEMPERATURE at point x of Ω
- $\rho(x)$ is the MASS DENSITY at point x of Ω
- $C(x, \theta)$ is the SPECIFIC MASS HEAT at point x of Ω and eventually $\theta(x)$
- ρC is the HEAT CAPACITY at point x of Ω
- $a_{i,j}(x, \theta)$ are the CONDUCTIVITY tensor coefficients which can be dependent of point x and temperature $\theta(x)$
- $F^\Omega(x, \theta)$ is the HEAT SOURCE (named SOURCE in Mefisto) which can be dependent of point x and eventually the temperature $\theta(x)$

Figure 96: parameters for heat transfer function

our problem is Poisson equation, we can use steady heat transfer solver to solve Poisson equation with some proper modification.

1. set conductivity tensor $a_{ij}(x, \theta) = 0.1$
2. set heat source $F^\Omega(x, \theta) = 1$
3. throw out convection term $V_i(x, \theta)$
4. throw out $CT(x, \theta)$
5. set Dirichlet B.C on T_1 and T_5

(1) in shell, type **THERMICER** to execute it, and enter **t1** as project as in Figure 97, then you will enter welcome frame of THERMICER, like Figure 98. Then press any key, you will see top command list as Figure 99.

```
LungShengChien@math-238dac3744 ~
$ THERMICER
=====
MEFISTO-THERMAL SOLVER: HEAT TRANSFER of an OBJECT on MICROSOFT CYGWIN PC
=====
Project (low case) name ?
t1
Execution MEFISTO THERMAL SOLVER in directory /home/LungShengChien/mefistox/t1

=====
| MM  MM  EEEEE  FFFFF  IIII  SSSSS  TTTTT  0000000  |
| MMM  MM  EE    FF    II  SS    TT  00  00  |
| MM  M  MM  EEEEE  FFFF  II  SSSSS  TT  00  00  | FAIT L'EF !
| MM  MM  EE    FF    II  SS    TT  00  00  |
| MM  MM  EEEEE  FF    IIII  SSSSS  TT  0000000  | Version About 2006
=====

DATE       : 20/ 9/ 6  15H 55M 05S
AUTHOR     : LungShengChien
PROJECT    : t1
```

Figure 97: type THERMICER to start it

(2) Select command **1;OBJECT NAME to treat**, then Figure 100 is shown.

(3) Input **ob**; // select Object, then Figure 101 is shown, notice that object “ob” is the same as we see in MAILLER

(4) Select command **2; HEAT TRANSFER INPUT DATA**, then Figure 102 is shown.

(5) First we set parameter for volume **solid3**, select **2; [A] THERMAL CONDUCTIVITY**, (setup conductivity tensor $a_{ij}(x, \theta) = 0.1$), then Figure 103 is shown.

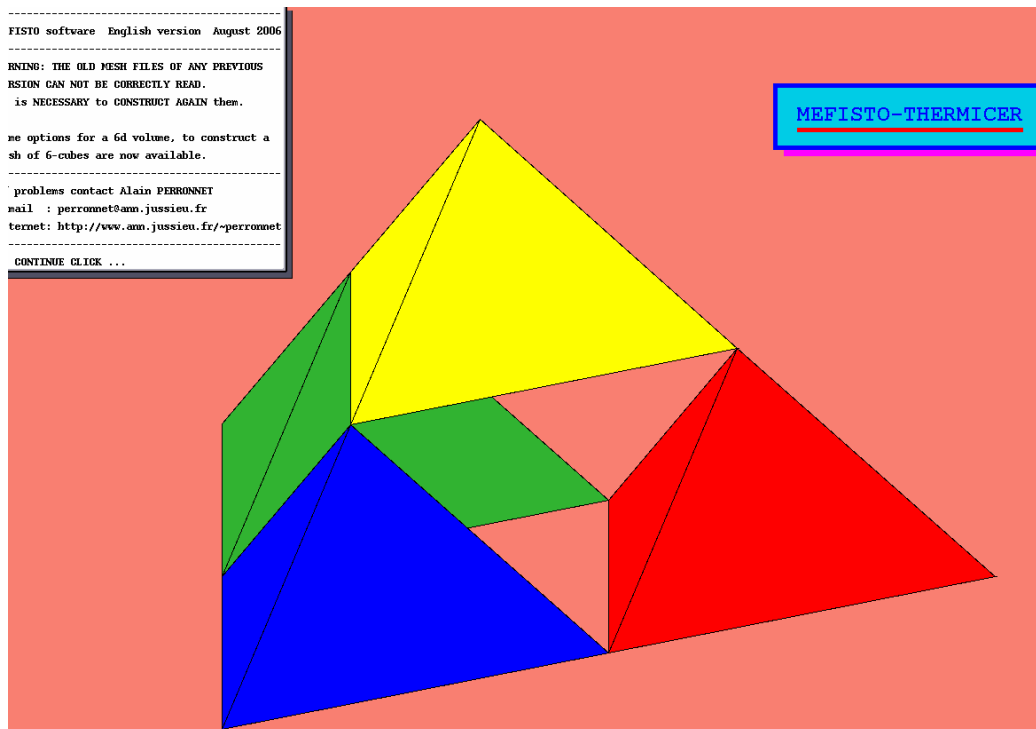


Figure 98: welcome frame of THERMICER

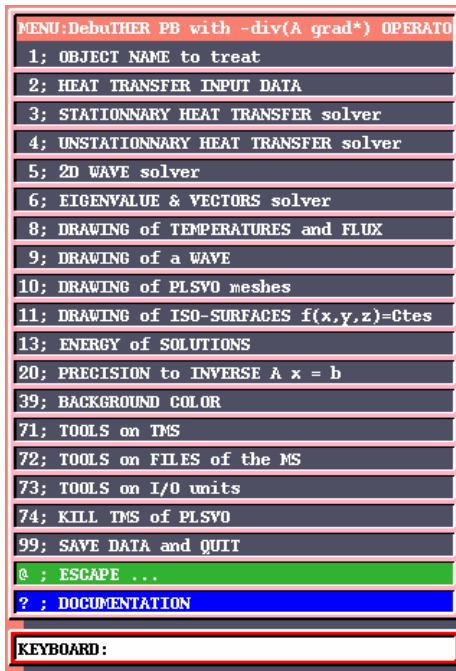


Figure 99: select 2;HEAT TRANSFER INPUT DATA

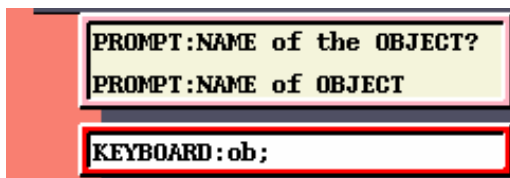


Figure 100: select object "ob"

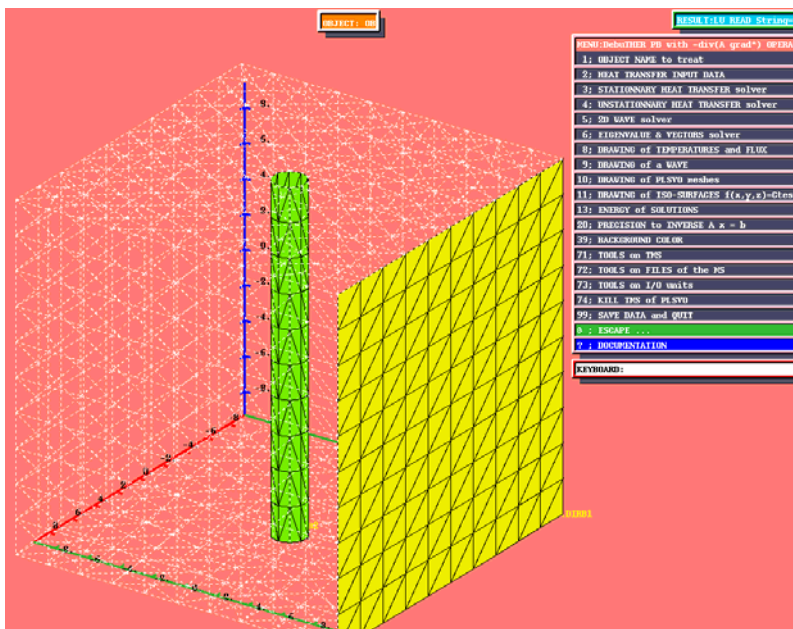


Figure 101: Object "ob" is shown in the frame with surface T_1 (yellow) and T_5 (green)

- (6) Select command **1; 3D homogenous isotropic material**, then Figure 104 is shown.
- (7) Input **0.1; // thermal conductivity**, then Figure 105 is shown.
- (8) Select command **5; vf Fluid speed for TRANSPORT**, then Figure 106 is shown.
- (9) Select command **0; kill the speed of fluid**, (throw out the convection term $V_i(x, \theta)$), then go back to Figure 105.
- (10) Select command **4; FOmega INTERNAL HEAT SOURCE**, then Figure 107 is shown.

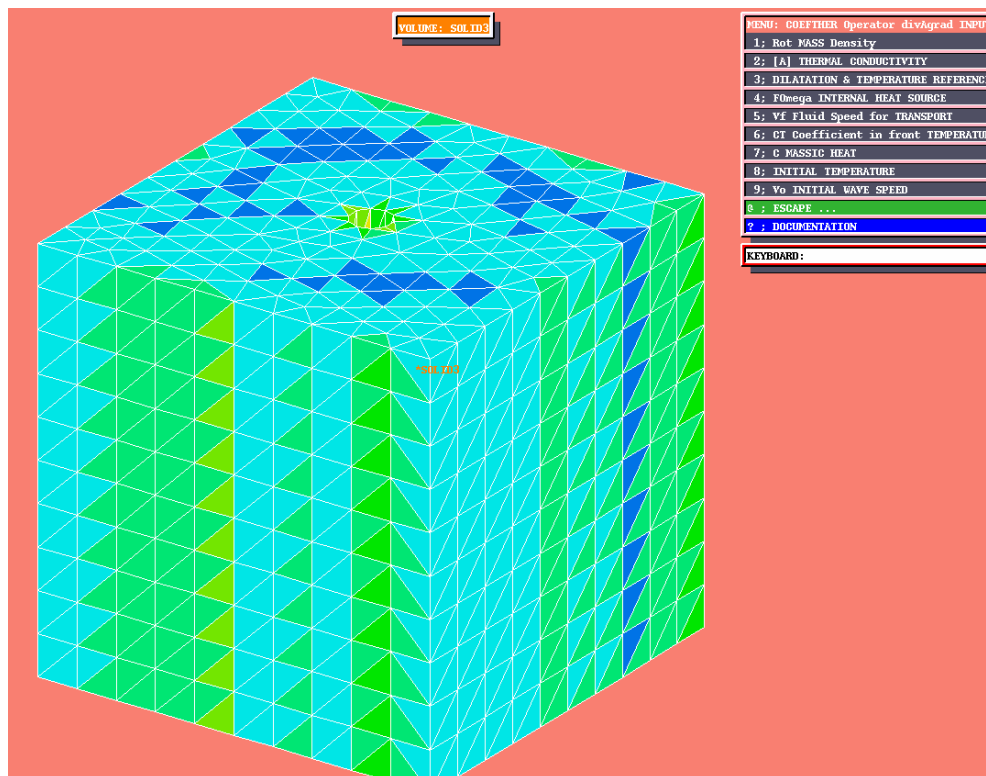


Figure 102: first we set parameter for volume solid3

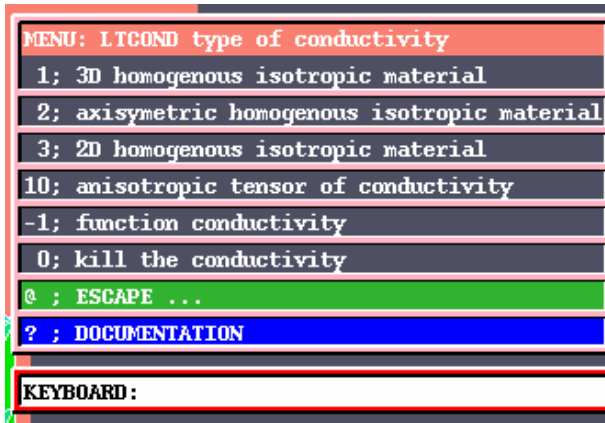


Figure 103: select 1; 3D
homogenous isotropic material

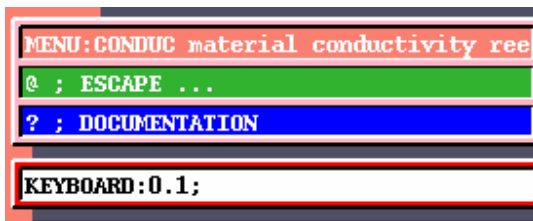


Figure 104: conductivity = 0.1

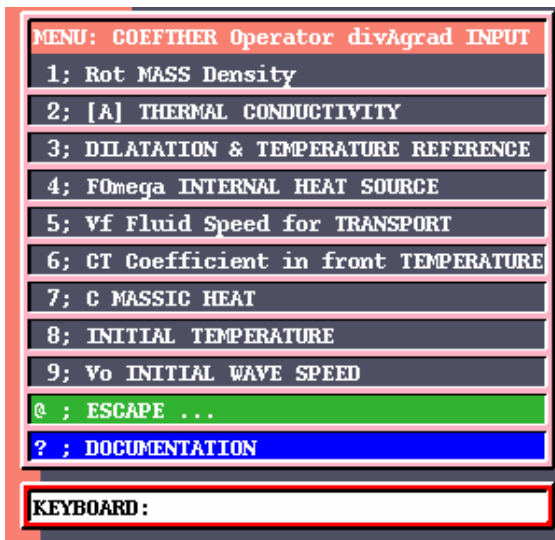


Figure 105: select 5; Vf Fluid Speed
for TRANSPORT

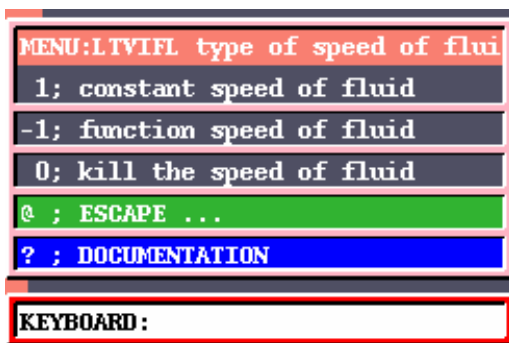


Figure 106: select 1; constant speed
of fluid

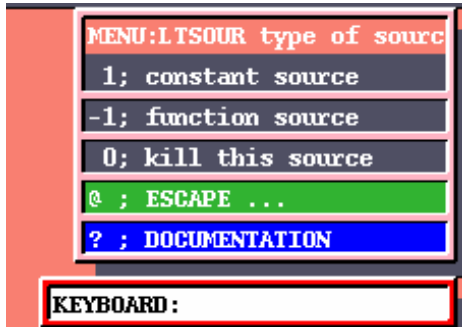


Figure 107: select 1; constant source

(11) Select command **1; constant source**, then Figure 108 is shown.

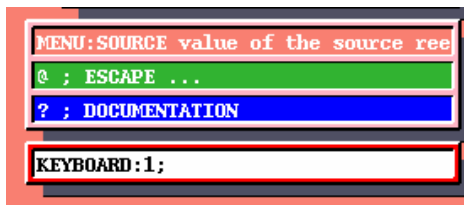


Figure 108: constant source = 1

(12) Input **1; (set heat source $F^\Omega(x, \theta) = 1$) // constant heat source**, then go back to Figure 105.

(13) Select command **6; CT Coefficient in front TEMPERATURE**, then Figure 109 is shown.

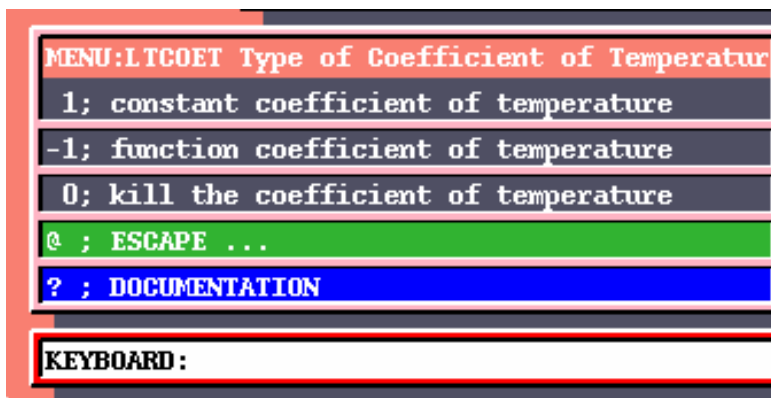


Figure 109: select 0; kill the coefficient of temperature

(14) Select command **0; kill the coefficient of temperature**, (throw out $CT(x, \theta)$), then go back to Figure 105.

(15) Parameters of solid3 is setup completely, press **ESC**, then Figure 110 is shown.

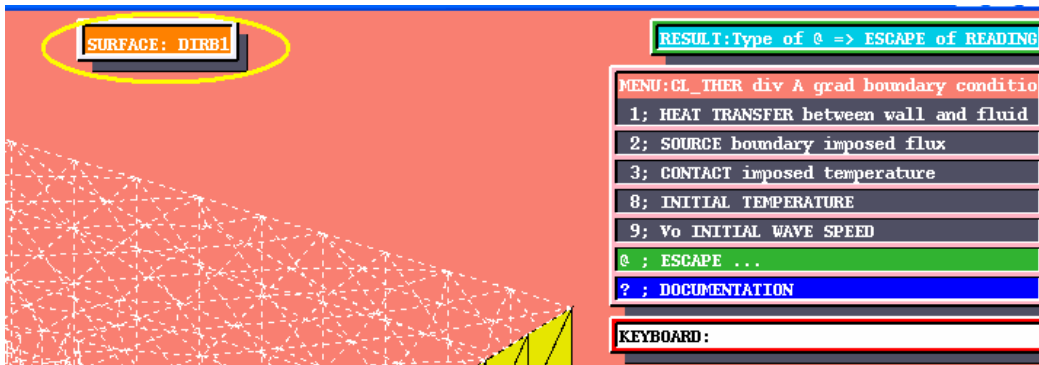


Figure 110: set parameter of surface dirb1, check the text inside yellow ellipse, if it is DIRB1, then you are right

Next we setup irichlet B.C for surface T_1 . Note that in Figure 110, text inside yellow ellipse must be **DIRB1**.

(16) Select command **3; CONTACT imposed temperature**, then Figure 111 is shown.

(17) Select command **1; constant imposed temperature**, then Figure 112 is shown.

(18) Input **38;** // temperature = 38⁰ C, then go back to Figure 110.

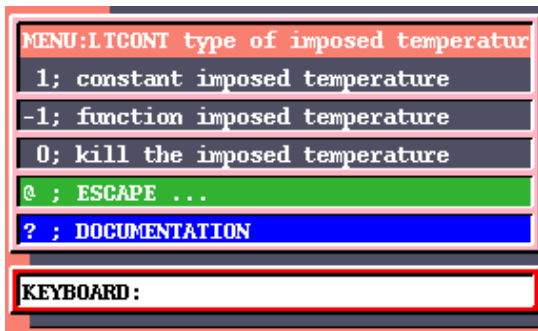


Figure 111: select 1: constant imposed temperature

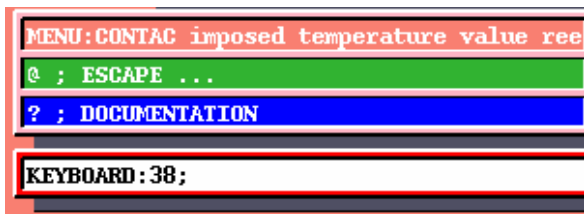


Figure 112: temperature = 38

(19) Press ESC to Figure 113

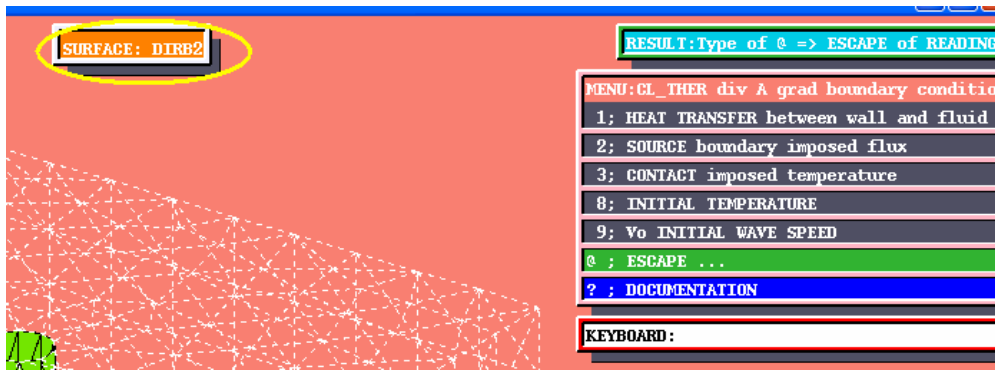


Figure 113: set parameter of surface dirb2, check the text inside yellow ellipse, if it is DIRB2, then you are right

Next setup Dirichlet B.C for for surface T_5 . Note that in Figure 113, text inside yellow ellipse must be **DIRB2**.

(20) Select command **3; CONTACT imposed temperature**, then Figure 111 is shown.

(21) Select command **1; constant imposed temperature**, then Figure 112 is shown.

(22) Input **0**; // temperature = 0°C , then go back to Figure 110, press ESC back to top level command list, Figure 99.

(23) Select command **3; STATIONARY HEAT TRANSFER solver**, then Figure 114 is shown.

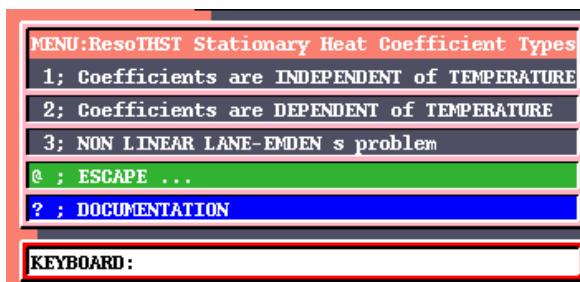


Figure 114:select 1; Coefficients are INDEPENDENT of TEMPERATURE

(24) Select command **1; Coefficient are INDEPENDENT of TEMPERATURE**, then Figure 115 is shown.

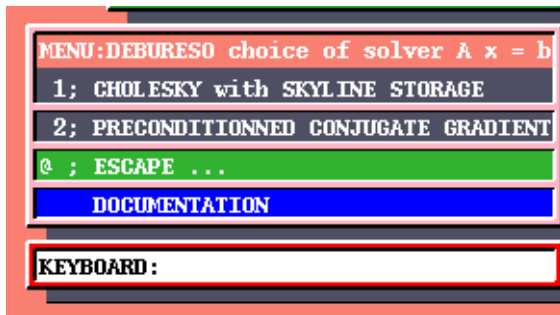


Figure 115: select 1;
CHOLESKY with SKYLINE
STORAGE

(25) Select command **1; CHOLESKY with SKYLINE STORAGE**, then program executes and you can see message output in the origin shell, like Figure 116.

(26) After program executes, press **ESC** back to top level command list. Select command **8; DRAWING of TEMPERATURE and FLUX**, then Figure 117 is shown.

(27) Select command **2; TEMPERATURE**, then Figure 118 is shown.

(28) Select command **1; ISOTHERMAL LINES or SURFACES**, then Figure 119 is shown.

(29) Select command **2; REGULAR ISO between MIN MAX**, then Figure 120 is shown.

(30) Select command **90; EXECUTE the DRAWING**, then result is shown in Figure 121.

(31) Besides, reader can repeat the procedure (1)~(30) again but setting heat source = 0, that is to say, solve Laplace equation $-0.1\Delta u = 0$, result is shown in Figure 122.

```

THDLFX: NUMBER of TEMPERATURES or NODES = 13314
THDLFX: NUMBER of TEMPERATURES FIXED = 861
THDLFX: NUMBER of TEMPERATURES NOT FIXED = 12453

CASE 1 RESULTANT of SOURCES : COMPONENT 1 = 7941.221
CHOLESKY FACTORIZATION. WAIT PATIENTLY...

SOLUTION by L (Lt x) = b

END of SOLUTION by L Lt x = b

CASE 1 TEMPERATURE of 10 FIRST NODES ;
=====
DoF 1: X= -9,0000 Y= -10,000 Z= -7,0000 ; TEMPERATURE = 942,6
219
DoF 2: X= -10,000 Y= -10,000 Z= -7,0000 ; TEMPERATURE = 944,0
438
DoF 3: X= -9,0000 Y= -10,000 Z= -8,0000 ; TEMPERATURE = 942,6
178
DoF 4: X= -9,0000 Y= -10,000 Z= -6,0000 ; TEMPERATURE = 942,6
261
DoF 5: X= -9,2889 Y= -9,3796 Z= -7,0000 ; TEMPERATURE = 941,9
361
DoF 6: X= -9,2889 Y= -9,3796 Z= -6,0000 ; TEMPERATURE = 941,9
410
DoF 7: X= -10,000 Y= -10,000 Z= -8,0000 ; TEMPERATURE = 944,0
384
DoF 8: X= -10,000 Y= -10,000 Z= -6,0000 ; TEMPERATURE = 944,0
482
DoF 9: X= -10,000 Y= -9,0000 Z= -7,0000 ; TEMPERATURE = 940,4
781
DoF 10: X= -10,000 Y= -9,0000 Z= -8,0000 ; TEMPERATURE = 940,4
748

At TIME 0,000000 VECTOR"TEMPERATURE NUMBER = 1
MINIMUM COMPUTED TEMPERATURE = 0,000000 at NODE 4502 : X= -0,80902
Y= -0,58779 Z= -6,0000
MAXIMUM COMPUTED TEMPERATURE = 944,1551 at NODE 4868 : X= 10,000
Y= -10,000 Z= 10,000

The NORMAL HEAT FLUXES on the OBJECT BOUNDARIES
=====
=
SURFACE : DIRB1 FLUX | case 1: -3554,7 | case
SURFACE : DIRB2 FLUX | case 1: -3844,5 | case

MATRIX FORMATION TIME = 0,80 CPU SECONDS
FACTORIZATION MATRIX TIME = 45,52 CPU SECONDS
COMPUTATION TEMPERATURE TIME = 0,25 CPU SECONDS
COMPUTATION HEAT FLUX TIME = 0,89 CPU SECONDS
SOLUTION TOTAL TIME = 47,45 CPU SECONDS

MATRIX SKYLINE STORAGE = 20739384 MEMORY WORDS

```

Figure 116: execution of thermal solver

```

MENU:TEMPGRAD Heat transfer drawing
1; CASE NUMBER of SOLUTION
2; TEMPERATURE
3; TEMPERATURE GRADIENT
4; NORMAL FLUXES
5; ERROR ESTIMATOR
@ ; ESCAPE ...
? ; DOCUMENTATION

KEYBOARD:

```

Figure 117: select TEMPERATURE

MENU: TRACTEMP Drawings of TEMPERATURES	
1;	ISOTHERMAL LINES or SURFACES
2;	ZONES of ISOTHERMAL COLORS
3;	ZONES by CUTTING PLANES X or Y or Z=CT
4;	PROFILES by CUTTING PLANES X or Y or Z=CT
5;	ALONG A LINE DEFINED BY 2 POINTS
7;	2D SURFACE(X,Y,TEMPERATURE(X,Y))
8;	2D ERROR from TEMPERATURE_EXACTE()
9;	PRINTING of TEMPERATURES
@ ;	ESCAPE ...
?	DOCUMENTATION
KEYBOARD:	

Figure 118: select 1;
ISOTHERMAL LINES or
SURFACES

MENU:VALISOTh Types of ISO-TEMPERATURE	
1;	Number of ISO-TEMPERATURES
2;	REGULAR ISO between MIN MAX
3;	DEFINE MIN MAX of ISO
4;	DEFINE VALUES of ISO
5;	COLOR of MESH s EDGES
6;	TYPE of LINES of MESH s EDGES
7;	COLOR of EDGES of ISO
8;	TYPE of LINES of EDGES of ISO
19;	REDUCTION % of FACES

Figure 119: 2; REGULAR ISO
between MIN MAX

MENU: VISEE3D Define a 3D VIEW	
19;	REDUCTION % of FACES
22;	GROW by 2 MOUSE clicks at Min MAX
23;	REDUCE by 2 MOUSE clicks
24;	TRANSLATE by 2 MOUSE clicks
25;	2D WINDOW [Xmin MAX] [Ymin MAX]
26;	RE-DEFINITION of EXTREMAL XYZ
27;	Total SCENE from EXTREMAL XYZ
30;	LONGITUDE & LATITUDE in degrees
31;	XYZ of the SEEN POINT & the EYE
32;	Half WIDTH & HEIGHT of the SCENE
33;	PLANE Behind Front of SEEN POINT
34;	ZOOMING of the Drawing
35;	ROTATION around Z-axis in degrees
36;	CHANGE the PALETTE of COLORS
38;	BACK to the DEFAULT COLORS
39;	BACKGROUND COLOR
44;	DRAW a TEXT on the SCREEN
47;	3 FAREST FACES of GLOBAL HEXAEDRON
48;	3 SEEN EDGES of GLOBAL HEXAEDRON
49;	REFERENCE SYSTEM AXES of PLSVO
50;	ERASE the WINDOW
60;	FRAME a DRAWING
75;	Activate TRANSLATION ORBIT ZOOM
76;	Stop the TRANSLATION ORBIT ZOOM
90;	EXECUTE the DRAWING

Figure 120: select 90; EXECUTE the
DRAWING

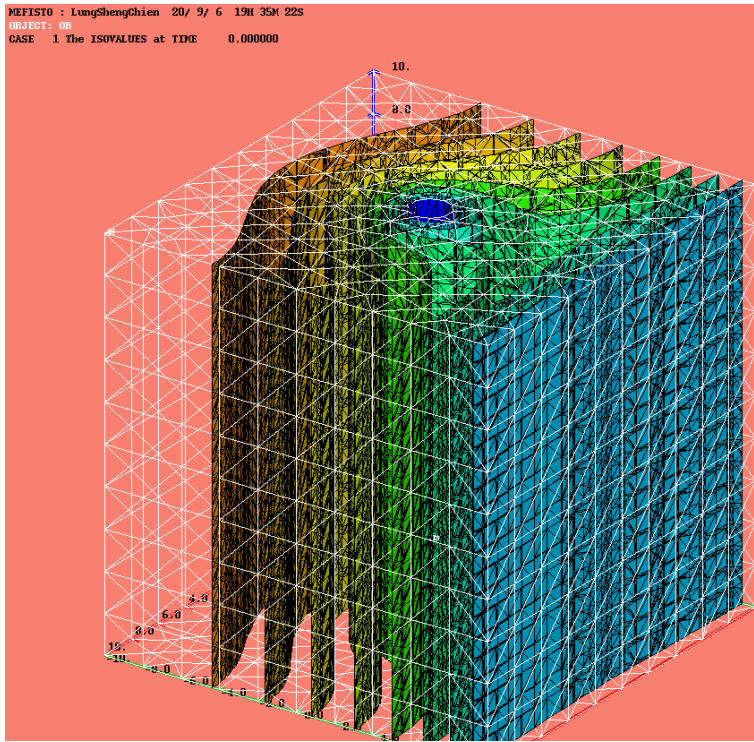


Figure 121: result of temperature

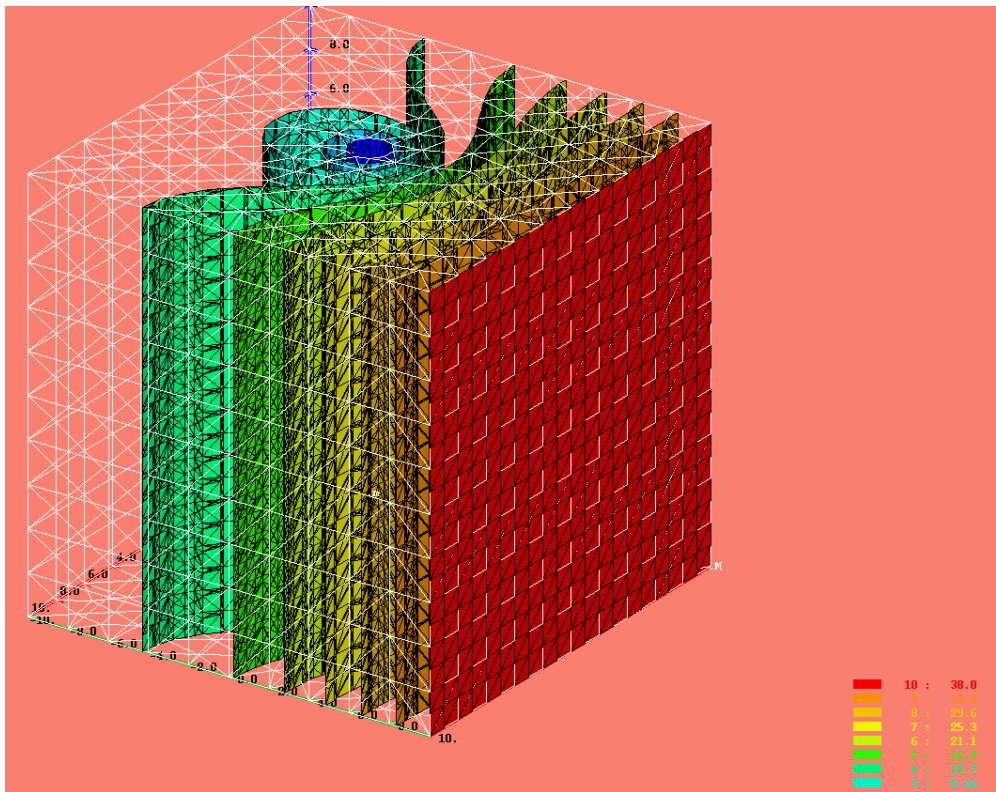


Figure 122: zero heat source, satisfies Maximum principle

Chapter 4 Q & A

Question 1: How to do if some steps are wrong, for example, adding the wrong point, line, surface?

Ans: you can press ESC several times, back to top level command list and repeat the procedures again by using the same name, because mefisto would overwrite the configuration if the name is the same.

Question 2: How to remove all geometry and restart ?

Ans: you can execute INITIER again and use the same project name, for example t1, then INITIER would delete file ms11, ms12, ms13 and ms14 under the directory t1. Since MAILLER would read geometry information from ms11, ms12, ms13, ms14 automatically, these 4 files must be remove or you must delete the geometry by using command in MAILLER.

Question 3: Does MAILLER can read file we have generated in chapter 2, for example xyznpef.OB and xyznsef.v.solid3?

Ans: No, these ASCII files just for user, not for MAILLER. In fact, all the command issued by user in MAILLER would be stored in frappe.001, frappe.002,... etc. Frappe.00x record command that user have done. For example, my frapper.001 is

```
{ ===== }
{ Software MEFISTO : Version Aout 2006 }
{ USER's Name : LungShengChien }
{ Project Name: t1 }
{ Date : 18/09/ 6 16h 25m 20s }
{ ===== }
19;
{ min MAX of XYZ of PLSVO }
{ X MINIMUM? }
-15;
{ X MAXIMUM? }
15;
{ Y MINIMUM? }
-15;
```

statement inside brace {...} is comment, mefisto would skip them. If you remember, 19 denotes “command 19; min MAX of XYZ of PLSV0”, this section of commands describe how to resize the view window as we do in chapter 3. you can

```
$ cp frappe.001 t1.mail
```

```
$ xemacs t1.mail
```

```
remove last line 99; {SAVE DATA and QUIT}
```

```
$ MAILLER
```

at top command list, type “**readf t1.mail**” after KEYWORD as Figure 123. Then MAILLER would execute line by line in file **t1.mail**.

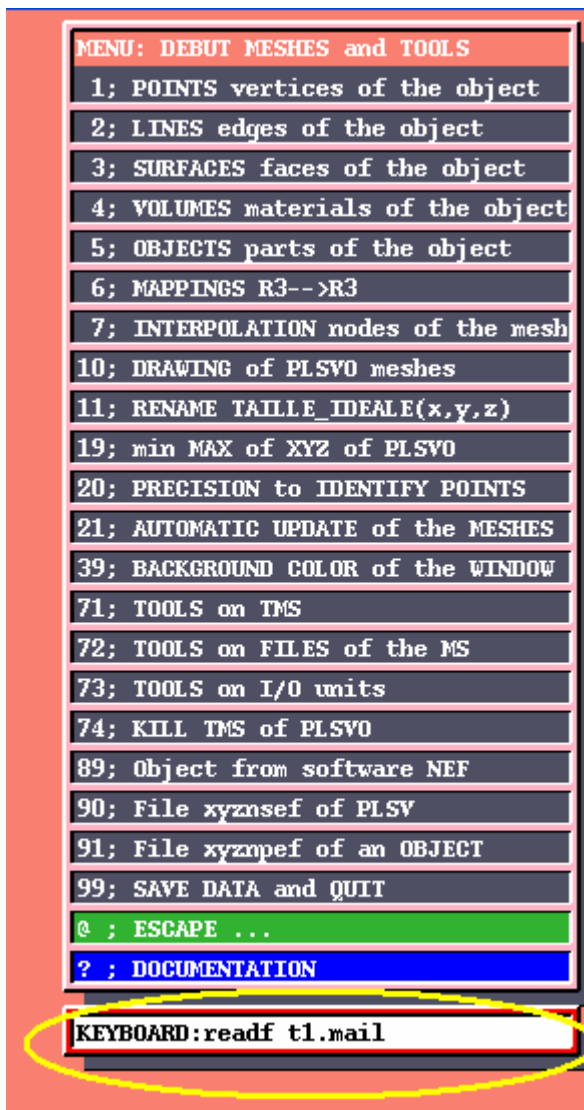


Figure 123: read file t1.mail

Question 4: what is limitation of mefisto?

Ans: In file \$MEFISTO/incl/pp.inc, we can find a statement
PARAMETER (MOTMCN = 128 000 000)

This is fortran language, it means that variable MOTMCN is an integer and denotes 128000000. Professor Perronnet said that this number affect global array allocated by mefisto, all subroutine use this global array. So if your problem's size is too large, then mefisto may not deal with. For example, for demo example, you divide each line into 20 segments (10 segments in this demo) and choose interpolation of Lagrange of degree 2, then MAILLER cannot generate mesh, you can try it.

How to solve this problem? One solution is

1. change MOTMCN to large number, for example MOTMCN = 500 000 000.
2. \$ppinc //recompile the source code
3. execute INITIER and MAILLER

Author suggest that you should do this modification in linux machine, author fails in winxp+cygwin. Besides even in linux machine, it may not work for MOTMCN = 500 000 000, you can try MOTMCN = 300 000 000 or smaller number.

Question 5: Is mefisto case-sensitive?

Ans: No, mefisto is written by fortran language, all the variables are case-insensitive, in fact, mefisto would change the name of variables to upper-case automatically.

Excise: If we want to change dimension of demo problem, for example, size of domain is $[-20, 20] \times [-20, 20] \times [-20, 20]$. How to change it from script

cube_cylinder.mail (This is collection of commands for this demo example, you can modify this file)?

Hint: you can create another project, say t2 and put cube_cylinder.mail into directory t2, then invoke MAILLER and use command “**readf cube_cylinder.mail;**” as Figure 123.

Appendix A : Command of MAILLER

1 ; POINTS vertices of the object

1; TYPING OF VALUES

2; MOUSE POSITION OF 2D POINTS

1; GENERIC NAME of POINTS

2; NAME of the MAPPING of the POINTS

3; ABSCISSE MIN MAX & STEP of FRAME

4; ORDINATE MIN MAX & STEP of FRAME

5; 3 POINTS define the PLAN of points

6; PLANE of POINTS at x=Constant

7; PLANE of POINTS at y=Constant

8; PLANE of POINTS at z=Constant

9; NO PLANE OF POINTS

20; MOUSE INPUT of POINTS

3; MOUSE POSITION OF 3D POINTS

2 : LINES edges of the object

2; STRAIGHT LINE

3; ARC OF CIRCLE PASSING BY 3 POINTS OF R**3

4; ARC OF CIRCLE PASSING BY 2+1 POINTS+RADIUS

5; ARC OF ELLIPSE OF R2 BY CENTRE AND XYAXES

6; LIST OF POINTS NAMED BY YOU

7; LIST OF VERTICES DEFINED BY THERIS X Y Z

8; COMPLETE 3D CIRCLE

1; circle defined by 3 points on the circle

2; circle defined by centre, point on the circle, point of the plane

9; COMPLETE 2D ELLIPSE

10; DATA OF ARRAYS XYZSOMNET ER NSEF

11; OPENED B-SPLINE FROM INTERPOLATION POINTS

12; OPENED B-SPLINE FROM CONTROL POINTS

21; CLOSED B-SPLINE FROM INTERPOLATION POINTS

22; CLOSED B-SPLINE FROM CONTROL POINTS

24; INTERSECTION OF PLANE-TRUNCATED CONE

25; INTERSECTION OF PLANE-ELLIPSOID

26; INTERSECTION OF TWO TRUNCATED CONE

27; INTERSECTION OF Z & X OR Y-AXIS CYLINDERS

28; FINITE LINE WITH A TANGENT CIRCLE

35; EXTRACTED EDGES OF A LINE BY EDGE NUMBER

36; EXTRACTED EDGES OF A LINE BY A CRITERION
37; INTERNAL OR BOUNDARY EDGES OF A SURFACE
40; RECONSTRUCTION OF AN UNSTRUCTURED LINE
50; MAPPING OF AN ACTUAL LINES
51; UNION OF SEVERAL LINES
81; RENAME THE LINE
82; KILL THE LINE

3 : SURFACES faces of the objects

1; TRNASFINITE QUADRANGLE
2; 2D STRUCTURED ELLIPTICAL QUADRANGLE
3; SPLINE QUADRANGLE BY INTERPOLATION POINTS
6; TRANSFINITE TRIANGLE
9; TRIANGULAITON BETWEEN CLOSED LINES
10; INPUT DATA OF XYZSONMET & NSEF TMS
11; 2D STRUCTURED RECTANGLE OF EDGES 1
12; 2D STRUCTURED RECTANGULAR TRIANGLE
13; QUADRANGULATION OF A CONVEX CLOSED CURVE
14; SLOCES FROM A GENERIC LINE (EXTRUCTION)
15; ROTATION OF A LINE AROUND AN AXIS
17; 1/8 OF THE SURFACE OF A SPHERE
18; 1/1 OF THE SURFACE OF A SPHERE
21; ANGULAR SECTOR OF A CYLINDER
28; FILLET SURFACE WITH TANGENT CYLINDER
30; QUALITY IMPROVEMENT OF A TRIANGULATION
31; TRIANGULATION OF A QUADRANGULATION
32; JUNCTION 1 POINT TO EDGES OF A LINE
33; JUNCTION BETWEEN EDGES OF LINES
34; QUADRANGLE FROM A STRUCTURED HEXAHEDRON
35; QUADRANGLE FROM A STRUCTURED QUADRANGLE
36; EXTRACTED SURFACE FROM A SURFACE
37; EXTRACTED SURFACE FROM FACES OF A VOLUME
 1; boundary faces
 2; internal faces
 3; any faces
42; G1-CONTINUITY OF SURFACES WITH TANGENTS
45; LOGICAL OPERATORS ON 2 2D-MESHES
50; MAPPING OF A SURFACE
51; C0-CONTINUITY UNION OF SURFACES

52; G1-CONTINUITY UNION OF SURFACES

81; RENAME A SURFACE

82; KILL THIS SURFACE

4 : VOLUMES materials of the object

1; STRUCTURED TRANSFINITE HEXAHEDRON

2; STRUCTURED ELLIPTICAL HEXAHEDRON

3; STRUCTURED TRANSFINITE TETRAHEDRON

4; STRUCTURED TRANSFINITE PENTAHEDRON

8; 1 CLOSED VOLUME FOR THE OPTION 9

9; TETRAHEDRIZATION OF VOLUMES OF TYPE 8

10; INPUT DATA OF XYZSOMNET & NSEF TMS

11; RECTANGULAR HEXAHEDRON WITH EDGES OF LENGTH 1

12; RECTANGULAR TETRAHEDRON WITH EDGES OF LENGTH 1

13; RECTANGULAR PENTAHEDRON WITH EDGES OF LENGTH 1

14; SLICES FROM A GENERIC SURFACE (EXTRUSION)

0; 3 points for each slice to define the displacements

1; 3 line vertices to define the mapping of the generic surface

2; translation of a constant cote $z>0$ (a stack)

3; translation of a cote $z>0$ for each displacement

4; translation vector (x y z) valuable for all slices

5; translation vectors (x y z) defined one by slice

6; 3 function X Y Z(no slice,x,y,z) to map the initial surface

15; ROTATION OF SURFACE AROUND AXIS (0 VERTEX/AXIS)

21; CYLINDER

22; 1/2 CYLINDER

23; CONE

24; 1/2 CONE

30; QUALITY IMPROVEMENT OF A TETRAHEDRON

31; TETRAHEDRIZATION OF AN HEXA-PENTAHEDRIZATION

35; EXTRACT AN HEXAHEDRON OF A STRUCTURED HEXAHEDRON

36; EXTRACT A VOLUME FROM A VOLUME & CRITERION

50; MAPPING OF A VOLUME

51; UNION OF SEVERAL VOLUMES

61; REGULAR 6-CUBE OF TYPE FINITE DIFFERENCES

62; SIMILARITY OF 6-CUBES

63; 3 FINITE DIFFERENCES AND SIMILARITY OF 6-CUBES

81; RENAME A VOLUME

82; KILL THIS VOLUME

5 : OBJECTS parts of the object

-1; nodes not renumbered

0; 2D or 3D classic

6 : MAPPINGS $R^3 \rightarrow R^3$

1; identity $R^3 \rightarrow R^3$

3; isometry $R^3 \rightarrow R^3$

0; translation defined by a R^3 vector

1; translation+rotation+dilation defined by 2 points+7 reals

2; translation+rotation+dilation defined by 4 images of 4 points

3; translation+rotation defined by 3 images of 3 points

4; plane symmetry defined by 3 points

5; line symmetry defined by 2 points

6; point symmetry

4; projection in R^3

6; $F:R^3 \rightarrow R^3$ define by 3 function (F_i)

7; $F: R^3 \rightarrow R^3$ 12 function (F_i et $[dF_i/dx_j]$)

7 : INTERPOLATION nodes of the mesh

1; AXISYMETRIC of DEGREE 1

2; AXISYMETRIC of DEGREE 2

3; LAGRANGE of DEGREE 1

4; LAGRANGE of DEGREE 2

10 : DRAWINGS of PLSV0 meshes

1; Draw All the POINTS

2; Draw All the LINES

3; Draw All the SURFACES without TG

1; BLACK and WHITE or COLORS

2; DRAWING or NOT of FACE EDGES

3; COLOR of FACE EDGES

4; DRAWING or NOT of FACES

5; REDUCTION % of FACES

6; DARKNESS or NOT

7; FACE NAME COLOR

8; DRAWING or NOT of SURFACE NAMES

9; DRAWING or NOT of the TITLE

10; DRAWING or NOT of VERTEX NUMBERS

11; COLOR of the VERTEX NUMBERS

12; DRAWING or NOT of the FE NUMBERS

13; COLOR of FE NUMBERS

14; SUBDIVISION MAX of a P3 EDGE
15; DRAWING or NOT of TANGENTS
16; COLOR of the TANGENTS
17; TANGENTS Line Drawing Type
20; THICKNESS of EDGES
49; AXES of SURFACES
50; ERASE the WINDOW
75; Active TRANSLATION ORBIT ZOOM
76; Stop the TRANSLATION ORBIT ZOOM
90; EXECUTING the DRAWING

4; Draw All the VOLUMES without TG
5; Draw All the OBJECTS 1 by 1
6; Draw the POINT
7; Draw the LINE
8; Draw the SURFACE
9; Draw the VOLUME
10; Draw the OBJECT
11; SIMPLE VERTICES of a NOT CLOSED LINE
12; SIMPLE EDGES of a NOT CLOSED SURFACE
13; ALL the SURFACE 1 by 1
14; ALL the VOLUMES 1 by 1
21; REDO the PREVIOUS DRAWING
22; GROW by 2 MOUSE clicks at Min Max
23; REDUCE by 2 MOUSE clicks
24; TRANSLATE by 2 MOUSE click
25; 2D WINDOW [Xmin Max] [Ymin Max]
26; RE-DEFINITION f EXTERNAL xyz
27; Total SCENE from EXTERNAL XYZ
30; LONGITUDE & LATTITUDE in degrees
31; XYZ of the SEEN POINT & the EYE
32; Half WIDTH & HEIGHT of the SCENE
33; PLANE Behind ahead of SCENE POINT
34; ZOOM of the Drawing
35; ROTATION around Z-axis in degrees
36; CHANGE the PALETTE of COLORS
37; DEFINE the COLOR of EDGES or FACES
38; BACK at the DEFAULT COLORS
39; BACKGROUND COLOR

- 40; DRAW or NOT the MESH QUALITY
- 41; QUALITY of LAST DRAW SURF or VOLU
- 44; TYPE and DRAW a TEXT
- 45; COLOR of the DRAW POINTS
- 46; LIGHTS in 3D without the QUALITY
- 47; 3 FAR FACES of the GLOBAL HEXAHEDRON
- 48; 3 SEEN EDGES of the GLOBAL HEXAHEDRON
- 49; REFERENCE SYSTEM AXIS of PLSV0
- 50; ERASE THE WINDOW
- 60; FRAME a DRAWING
- 11 : RENAME TAILLE_IDEALE(x,y,z)
- 19 : min MAX of XYZ of PLSV0
- 20 : PRECISION to IDENTIFY POINTS
- 21 : AUTOMATIC UPDATE of the MESHES
- 39 : BACKGROUND COLOR of the WINDOW
- 71 : TOOLS on TMS
- 72 : TOOLS on FILES of the MS
- 73 : TOOLS on I/O units
- 74 : KILL TMS of PLSV0
- 89 : Objects from software NEF
- 90 : File xyznsef of PLSV**
 - 1; POINT
 - 2; LINE
 - 3; SURFACE
 - 4; VOLUME
 - 0; WITHOUT TANGENTS
 - 1; WITH TANGENTS
- 91 : File xyznpef of an OBJECT**
- 99 : SAVE DATA and QUIT**

Appendix B. Command of THERMICER

1; OBJECT NAME to treat

2; HEAT TRANSFER INPUT DATA

for volume

1; Rot MASS density

2; [A] THERMAL CONDUCTIVITY

1; 3D homogenous isotropic material

2; axisymmetric homogenous isotropic material

3; 2D homogenous isotropic material

10; anisotropic tensor of conductivity

-1; function conductivity

0; kill the conductivity

3; DILATATION & TEMPERATURE REFERENCE

4; FOMEGA INTERNAL HEAT SOURCE

1; constant source

-1; function source

0; kill this source

5; vF Fluid speed for TRANSPORT

1; constant speed of fluid

-1; function speed of fluid

0; kill the speed of fluid

6; CT Coefficient in front TEMPERATURE

1; constant coefficient of temperature

-1; function coefficient of temperature

0; kill the coefficient of temperature

7; C MASSIC HEAT

8; INITIAL TEMPERATURE

9 V0 INITIAL WAVE SPEED

for surface

1; HEAT TRANSFER between wall and fluid

2; SOURCE boundary imposed flux

3; CONTACT imposed temperature

1; constant imposed temperature

-1; function imposed temperature

0; kill the imposed temperature

8; INITIAL TEMPERATURE

9; V0 INITIAL WAVE SPEED

3; STATIONARY HEAT TRANSFER solver

1; Coefficient are INDEPENDENT of TEMPERATURE

1; CHOLESKY with SKYLINE STORAGE

2; PRECONDITIONED CONJUGATE GRADIENT

2; Coefficient are DEPENDENT of TEMPERATURE

3; NON LINEAR LANE-EMDEN s problem

4; UNSTATIONARY HEAT TRANSFER solver

5; 2D WAVE solver

6; EIGENVALUE & VECTOR solver

8; DRAWING of TEMPERATURE and FLUX

1; CASE NUMBER of SOLUTION

2; TEMPERATURE

1; ISOTHERMAL LINES or SURFACES

1; Number of ISO-TEMPERATURE

2; REGULAR ISO between MIN MAX

3; DEFINE MIN MAX of ISO

4; DEFINE VALUES of ISO

5; COLOR of MESH s EDGES

6; TYPE of LINES of MESH s EDGES

7; COLOR of EDGES of ISO

8; TYPE of LINES of EDGES of ISO

19; REDUCTION % of FACES

2; ZONES of ISOTHERMAL COLORS

3; ZONES by CUTTING PLANES X or Y or Z=CT

4; PROFILES BY CUTTING PLANES X or Y or Z=CT

5; ALONG A LINE DEFINED BY 2 POINTS

7; 2D SURFACE(X,Y,TEMPERATURE(X,Y))

8; 2D ERROR from TEMPERATURE_EXACTE()

9; PRINTING of TEMPERATURE

3; TEMPERATURE GRADIENT

4; NORMAL FLUXES

5; ERROR ESTIMATOR

9; DRAWING of a WAVE

10; DRAWING of PLSV0 meshes

11; DRAWING of ISO-SURFACE $f(x,y,z)=ctes$

13; ENERGY of SOLUTIONS

20; PRECISION to INVERSE $A x = b$

39; BACKGROUND COLOR

71; TOOLS on TMS
 72; TOOLS on FILES of the MS
 73; TOOLS on I/O units
 74; KILL TMS of PLSV0
 99; SAVE DATA and QUIT

Appendix C format of xyznsef.v.solid3

Note that the statement after double backslash // is additive interpretation for last line above the line of //. For example,

```
solid3      { PLSV's NAME }
// name of volumn, it can be any string, depending on user
interpret that solid3 is name of the volume.
```

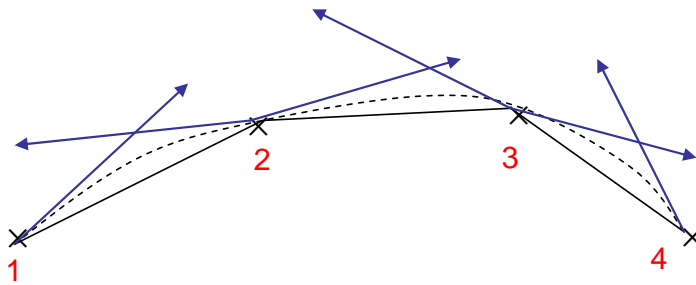
```
solid3      { PLSV's NAME }
// name of volume, it can be any string, depending on user
      3      {NBCOOR PLSV's VERTICE COORDINATES NUMBER}
// 3D surface, since we input points as 3-tuple (x,y,z)
      1881   {NBSOM PLSV's VERTICES NUMBER}
// total number of vertices = 24, see Figure 126
      0      {NBTGS PLSV's TANGENT VECTORS NUMBER}
// in mefisto, tangent vectors are generated as Figure 124, but these tangent vectors are
// useless when we just need geometry information.
      0.1000000E+01  0.0000000E+00  -0.1000000E+02      // x1 y1 z1
      0.8090170E+00  -0.5877852E+00  -0.1000000E+02      // x2 y2 z2
      .....
      0.5377303E+00  0.1645961E+01  0.1000000E+02      // x1881 y1881 z1881

4      {NUTYOB type of the PLSV(1 2 3 4)}
// type 1: point , type 2: line , type 3: surface , type 4: volume
-1     {NUTFMA type: CLOSED(1) or NOT(0) or UNKNOWN(-1)}
// see Figure 125, this number is useless in fact, you can ignore it.
8      {NBSOEF number of vertices for 1 FE}
// although element is tetrahedron, it should be 4 vertices, the implementation issue
// results in this number, see Remark 5.
0      {NBTGEF number of tangent vectors for 1 FE}
// useless
```

```

8760 {NBEFOB  number of FE of the PLSV}
0   {NBEFTG  number of FE with tangent vectors}
// useless
0   {NBEFAP  number of FE with POINTER on FE with TG}
// useless
0   {NUTYMA  structured type(1,...,7) or NOT(0)}
// useless
    22 306 135 73  0 0 0 0  // FE1
// see Remark 5
    22 306 73 244 0 0 0 0
....
    1545 1878 1716 1717 0 0 0 0  // FE8760

```



- Boundary of object passing through points 1,2,3 and 4
- Tangent line of boundary curve at points 1,2,3 and 4
- Each point has two tangent vectors

Figure 124: from point 1 to point 2, we have a tangent vector at point 1. Similarly, from point 2 to point 1, we still have a tangent vector. Hence each point has two tangent vectors.

Remark 5: in 3D, we have 3 kinds of element, tetrahedron, pentahedron and hexahedron. Logically speaking we need 3 arrays to record the information of each type of element, say

NV_tetrahedron(1:4, NOTETRA) : 4 points

NV_pentahedron(1:6, NOPENTA) : 6 points

NV_hexahedron(1:8, NOHEXA) : 8 points

However the author use a single array to store 3 types of element, say

NV_3DFE(1:8, NOFE).

Hence if element is tetrahedron, then we have p1,p2,p3,p4,0,0,0,0. if element is pentahedron, then we have p1,p2,p3,p4,p5,p6,0,0.

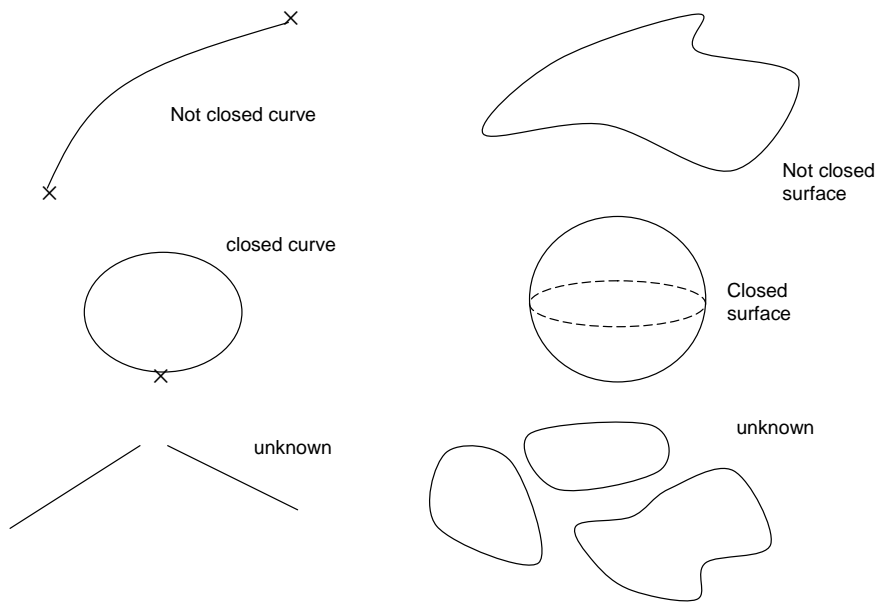


Figure 125: type of NUTFMA

Appendix D format of xyznsef.s.abcd

Example 1: Two dimensional quadrilateral mesh

Consider rectangle abcd in Figure 126 (from lecture note of NCTS short course), it has 15 elements, each element is rectangle with 4 points. The content from MAILLER is

```

abcd      { PLSV's NAME }
          3  {NBCOOR PLSV's VERTICE COORDINATES NUMBER}
          24 {NBSOM PLSV's VERTICES NUMBER}
           0  {NBTGS PLSV's TANGENT VECTORS NUMBER}
0.0000000E+00  0.0000000E+00  0.0000000E+00
// p1 = (x1, y1, z1)
0.2000000E+00  0.0000000E+00  0.0000000E+00
// p2 = (x2, y2, z2)
.....
0.6000000E+00  0.1000000E+01  0.0000000E+00
// p24 = (x24, y24, z24)

```

```

3 {NUTYOB  type of the PLSV(1 2 3 4)}
// type 1: point , type 2: line , type 3: surface , type 4: volume
0 {NUTFMA  type: CLOSED(1) or NOT(0) or UNKNOWN(-1)}
// ???
4 {NBSOEF  number of vertices for 1 FE}
// each element has 4 vertices since it is rectangle mesh
0 {NBTGEF  number of tangent vectors for 1 FE}
// no tangent vectors specified in element
15 {NBEFOB  number of FE of the PLSV}
// total number of FE = 15
0 {NBEFTG  number of FE with tangent vectors}
// no tangent vectors
0 {NBEFAP  number of FE with POINTER on FE with TG}
// ???
4 {NUTYMA  structured type(1,...,7) or NOT(0)}
// ???
1      2      6      5
//  $FE_1 = (p_1, p_2, p_6, p_5)$  counterclockwise
2      3      7      6
3      4      8      7
5      6      10     9
6      7      11     10
7      8      12     11
9      10     14     13
10     11     15     14
11     12     16     15
13     14     18     17
14     15     19     18
15     16     20     19

```

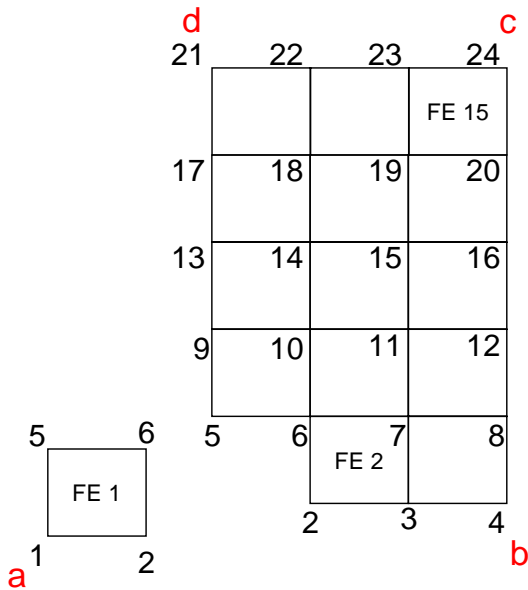


Figure 126: 15 elements of rectangle in rectangle abcd.

Example 2: surface T_1 of demo example

We output surface T_1 by choosing with tangent, but tangent vector is zero in this case.

```

dirb1      { PLSV's NAME }
          3  {NBCOOR PLSV's VERTICE COORDINATES NUMBER}
        121 {NBSOM PLSV's VERTICES NUMBER}
          0  {NBTGS PLSV's TANGENT VECTORS NUMBER}
0.1000000E+02  0.1000000E+02 -0.1000000E+02
0.8000000E+01  0.1000000E+02 -0.1000000E+02
.....
-0.1000000E+02  0.1000000E+02  0.1000000E+02
    3 {NUTYOB  type of the PLSV(1 2 3 4)}
    0 {NUTFMA  type: CLOSED(1) or NOT(0) or UNKNOWN(-1)}
    4 {NBSOEF  number of vertices for 1 FE}
// why each element has 4 vertices
    0 {NBTGEF  number of tangent vectors for 1 FE}
  200 {NBEFOB  number of FE of the PLSV}
    0 {NBEFTG  number of FE with tangent vectors}
    0 {NBEFAP  number of FE with POINTER on FE with TG}
    0 {NUTYMA  structured type(1,...,7) or NOT(0)}
    1      2      13      0
// what does '0' mean?

```

```

2      3      14      0
.....
4      5      16      0

```

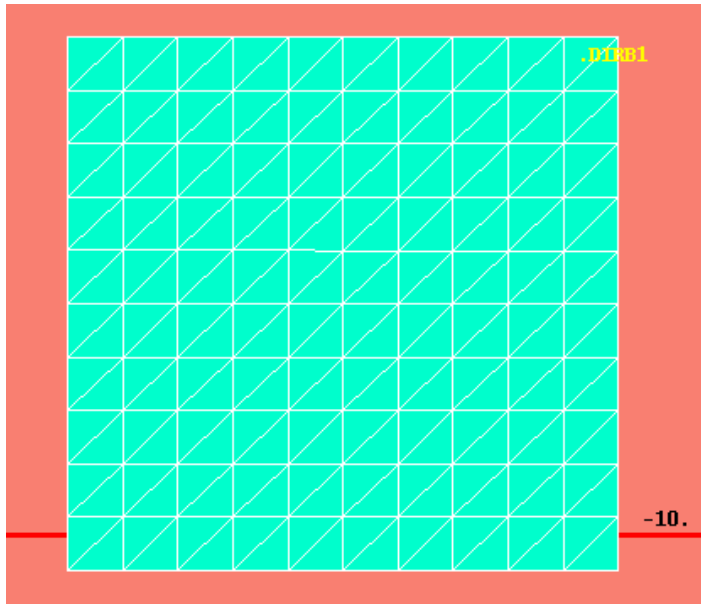


Figure 127: triangular mesh, total number of element = 200, 2 triangle of each rectangle.

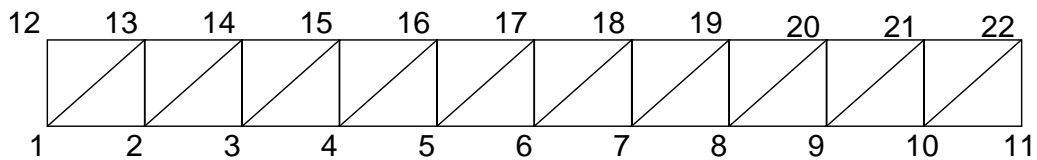


Figure 128: part of elements in surface T_1

Appendix E format of xyznpef.OB

Example 3: object of T4 in short course

Consider rectangle abcd in Figure 126 (from lecture note of NCTS short course), store object as file xyznpef.ob.

```

OB                                {NAME of the OBJECT}
3    {NBCOOR COORDINATE NUMBER of a NODE (3 or 6) }
// 3D setting
2    {SPACE's DIMENSION of the OBJECT (2 3 or 6)}
// z-component is 0 for all points, mefisto can detect this and say the problem is 2D
62   {NBNOEUDS NUMBER of NODES of the OBJECT}

```

```

// we use interpolation of Lagrange of degree 2, the number of points per element = 8,
// these points are used to construct basis function, see Figure 129. The number of grid
// points + the points marked by 'x' = 62
      0    {NBTG NUMBER of TANGENTS of the OBJECT}
      0.0000000E+00  0.0000000E+00  0.0000000E+00
      0.1000000E+00  0.0000000E+00  0.0000000E+00
.....
      0.5000000E+00  0.1000000E+01  0.0000000E+00
      1    {NBOBIN NUMBER of MATERIALS}
// thermal conductivity = 52.0W / m0C
      1    {is the NUMBER in SURFACE of }
ABCD
      6    {NBOBCL NUMBER of PL(S) at the BOUNDARY}
// number of boundary nodes are 6, including 4 edges and 2 points. Here PL(S) inside
// { } denotes that P = points, L = line, S = surface
      4    {NUMBER of LIGNE      }
      1    {is the NUMBER in LIGNE of }
AB
      4    {is the NUMBER in LIGNE of }
DA
      2    {is the NUMBER in LIGNE of }
BC
      3    {is the NUMBER in LIGNE of }
CD
      2    {NUMBER of POINT      }
      1    {is the NUMBER in POINT of }
A
      2    {is the NUMBER in POINT of }
B
      1    {NBTYEL NUMBER of TYPES of FE of the OBJECT}
// we only use single type of elements, say quadrilateral, if we have two types of
// element in Figure 130, then the value is 2. However we only have two kinds of
// elements in 2D, quadrilateral and triangular, so the value is 1 or 2 in 2D. In 3D we
// have 3 kinds of elements, tetrahedron, pentahedron and hexahedron, so the value is
// 1 or 2 or 3 in 3D.
      15   {NBELEM NUMBER of FE of THIS TYPE}
// total number of elements = 15.
      8    {NBNOE  NUMBER of NODES of a FE of THIS TYPE}

```

// there are 8 nodes if using Lagrange of degree 2. This number depends on what kind
 // of interpolation you do.

4 {NBSOE NUMBER of VERTICES of a FE of THIS TYPE}

// element has 4 points (4 nodes), this number is independent of interpolation.

4 {NBARET NUMBER of EDGES of a FE of THIS TYPE}

// element has 4 edges which is generated by 4 points.

1 {NBFACE NUMBER of FACES of a FE of THIS TYPE}

// element has 1 surface which is union of 4 edges.

0 {NBVOL NUMBER of VOLUMES of a FE of THIS TYPE}

// element is 2D, no volume

23	25	36	34	24	31	35	30	0	0
0	0	0	0	0	0	4	1		

// first 8 numbers are vertices of FE_1 , the remaining values (9-th ~17-th) are

// boundary nodes 4 points (red), 4 lines (blue) and 1 surface (black), ordered as in

// Figure 129. In this case, boundary points are A,B,C and D, boundary lines are \overline{AB} ,

// \overline{BC} , \overline{CD} and \overline{DA} . Hence FE_1 has no boundary points, first 4 numbers are 0, 0,

// 0, 0. However line ordered 8 is part of \overline{DA} which is number of 4, so 5-th ~ 8-th

// numbers are 0, 0, 0, 4. Finally, FE_1 belongs to only one surface abcd, which is

// number 1, so last number is 1.

34	36	45	42	35	43	44	41	0	0
0	0	0	0	0	0	4	1		

.....

12	14	25	23	13	20	24	19	0	0
0	0	0	0	0	0	4	1		

15 {TOTAL NUMBER of FE of the OBJECT OB

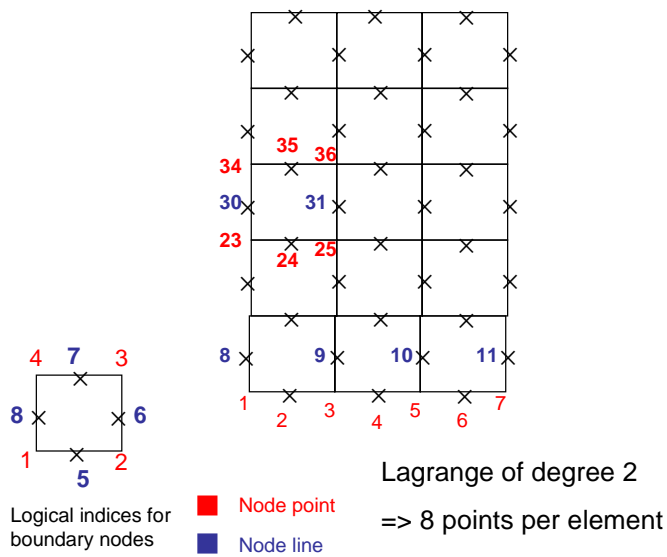


Figure 129: interpolation of Lagrange of degree 2 results in more points

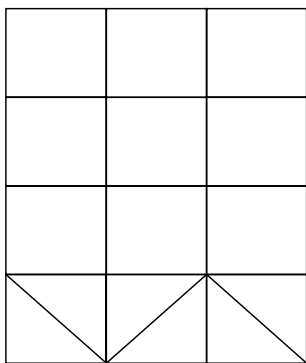


Figure 130: Two type of elements, one is quadrilateral and the other is triangular.

Example 4: object of demo example

```

OB                                {NAME of the OBJECT}
    3    {NBCOOR COORDINATE NUMBER of a NODE (3 or 6) }
    3    {SPACE's DIMENSION of the OBJECT (2 3 or 6)}
// actual 3D object, different from Example 3.
13314  {NBNOEUDS NUMBER of NODES of the OBJECT}
    0    {NBTG  NUMBER of TANGENTS of the OBJECT}
-0.9000000E+01  -0.1000000E+02  -0.7000000E+01
-0.1000000E+02  -0.1000000E+02  -0.7000000E+01
.....
    0.1000000E+02  0.7000000E+01  0.2000000E+01

```

```

1    {NBOBIN NUMBER of MATERIALS}

// it is thermal conductivity  $a_{i,j} = 10^{-1}$ 

3    {is the NUMBER in VOLUME      of }
SOLID3
// Recall we have created solid1 and solid2 before solid is created, so number of
// solid3 in MAILER is 3
2    {NBOBCL NUMBER of PL(S) at the BOUNDARY}
// boundary nodes = 2 surfaces, dirb1 + dirb2
2    {NUMBER of SURFACE      }
// number of sf1 = 1, number of dirb1 = 2, number of dirb2 = 3
2    {is the NUMBER in SURFACE    of }
DIRB1
3    {is the NUMBER in SURFACE    of }
DIRB2
0    {NUMBER of LIGNE      }
// no boundary node of lines in our setting
0    {NUMBER of POINT      }
// no boundary node of points in our setting
1    {NBTYEL NUMBER of TYPES of FE of the OBJECT}
// element is tetrahedron
8760 {NBELEM NUMBER of FE of THIS TYPE}
10   {NBNOE  NUMBER of NODES of a FE of THIS TYPE}
// red + blue in Figure 131.
4    {NBSOE  NUMBER of VERTICES of a FE of THIS TYPE}
// 4 points of a tetrahedron
6    {NBARET NUMBER of EDGES of a FE of THIS TYPE}
// 6 edges of a tetrahedron
4    {NBFACE NUMBER of FACES of a FE of THIS TYPE}
// 4 surfaces of a tetrahedron
1    {NBVOL  NUMBER of VOLUMES of a FE of THIS TYPE}
// 1 volume of a tetrahedron
5464 5468 6812 6546 5466 6192 6193 5878 5897 6810
0    0    0    0    0    0    0    0    0    0
3    0    0    0    3
// see Figure 131, first 10 numbers denote 10 nodes of  $FE_1$ , there are 15 numbers left,
// first 4 number (red) denote boundary node of points, 5-th ~ 10-th (blue) denote
// boundary node of lines, 11-th ~14th (purple) denote boundary node of surfaces,

```

// 15-th (orange) denotes boundary node of volume.

.....

```

2192 1427 1715 1691 1811 1712 1833 1812 1713 1714
  0   0   0   0   0   0   0   0   0   0
  0   0   0   0   3
8760 {TOTAL NUMBER of FE of the OBJECT OB  }

```

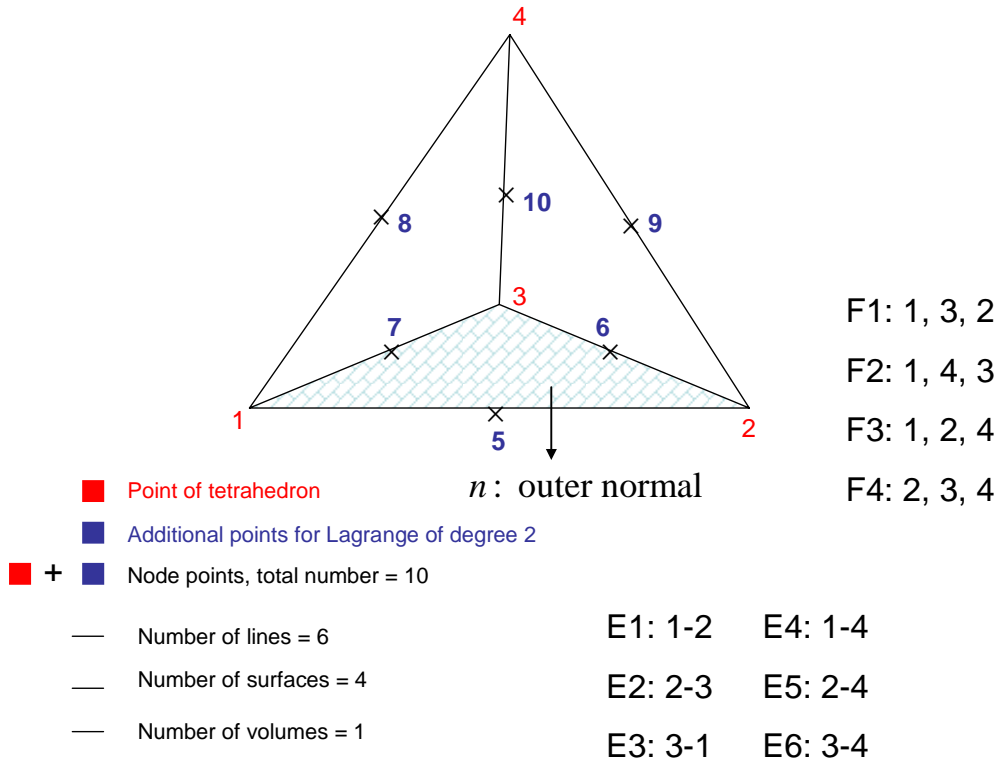


Figure 131: configuration of a tetrahedron element, including numbering of points, edges, and surface. Note that this numbering follows the code

Remark 6: In Figure 131, the numbering of points, edges and surfaces are adapted from the code in \$MEFISTO/util/soarfa.f. Part of code is shown as

```

C    FACE 1 = p1, p3, p2
500 NOSOFA(1,1) = 1
    NOSOFA(2,1) = 3
    NOSOFA(3,1) = 2
C    FACE 2 = p1, p4, p3
    NOSOFA(1,2) = 1
    NOSOFA(2,2) = 4

```

```

      NOSOFA(3,2) = 3
C     FACE 3 = p1, p2, p4
      NOSOFA(1,3) = 1
      NOSOFA(2,3) = 2
      NOSOFA(3,3) = 4
C     FACE 4 = p2, p3, p4
      NOSOFA(1,4) = 2
      NOSOFA(2,4) = 3
      NOSOFA(3,4) = 4

```

Face k has first point = NOSOFA(1, k), second point = NOSOFA(2, k) and third point = NOSOFA(3, k). The order of points of a surface depends on normal direction, for example F1 = 1,3,2 since $\vec{n} = \overline{p_1 p_3} \times \overline{p_1 p_2}$.

```

300 NBSOM = 3
310 DO 350 I=1,NBSOM
      NOSOAR(1,I) = I
      NOSOAR(2,I) = I + 1
      NOSOFA(I,1) = I
350 CONTINUE
      NOSOAR(2,NBSOM) = 1
      RETURN
C     LES ARETES
      DO 510 I=1,3
          NOSOAR(1,I+3) = I
          NOSOAR(2,I+3) = 4
510 CONTINUE
      GOTO 300

```

Edge k has first point = NOSOAR (1, k), second point = NOSOAR (2, k).