How to generate a surface in mefisto

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

In previous paper, see howTo.pdf and manual_quantumDot.pdf, we learn how to specify plane in mefisto, it is very simple, you just

1. Create points.
2. Create straight line by two points.
3. Create a plane by union of several edges.

However if you want a surface, the situation becomes complicated. Most physical problem relates to a surface, not just a plane. If geometry is a plane, then FDM (finite difference method) can work well. When talking about FEM, complicated geometry is essential topic.

In this paper we talk about how to generate a surface by mefisto. According to professor perronnet's opinion, there are two options in mefisto to generate a surface,

1; TRNASFINITE QUADRANGLE
3; SPLINE QUADRANGLE BY INTERPOLATION POINTS
under **3 : SURFACES faces of the objects**

Here we describe basic idea of two methods

1. transfinite quadrangle
   given a 4 line (may not be straight line), mefisto would use these 4 lines to interpolate a surface, the result surface may be quadrilateral mesh or triangular mesh.

2. spline quadrangle
   Given NxN points, mefisto use polynomial approximation to generate a surface

for example, Figure 1 use 25 points to generate the surface, please take care of order of points feed into mefisto.

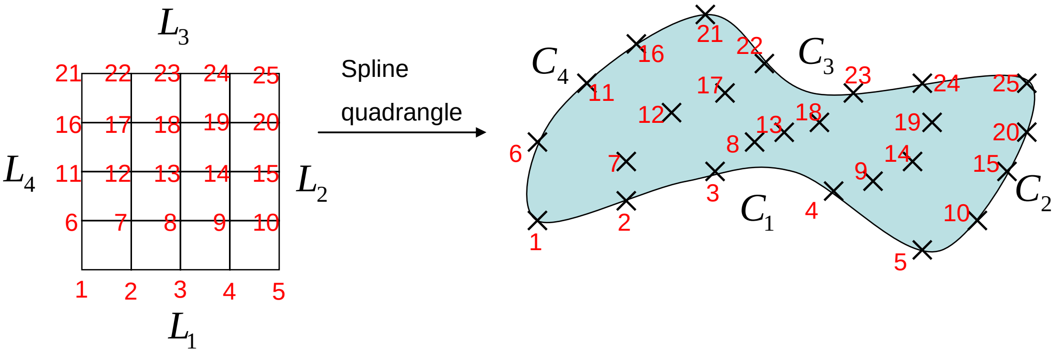

Figure 2: $L_j \to C_j \quad \forall j = 1,2,3,4$ and order of points is very important, just follow the red number.

Example: toast-shaped with saddle-like top surface, see Figure 3.

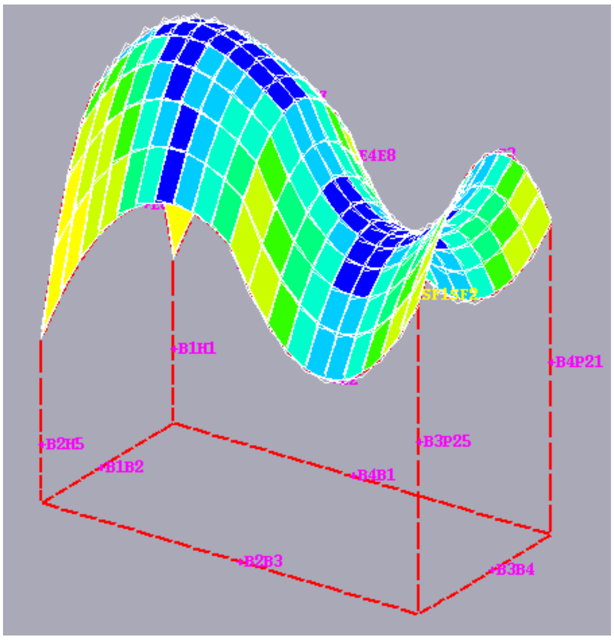

Figure 3: toast-shaped with top saddle-like surface. Here we don't draw color for bottom surface and 4 walls.

We use a graph $z = \varphi(x, y)$ on $\Omega = [-1,1]\times[-3,1]$ to define top surface .

$$\varphi(x,y) = \begin{cases} y^2 - x^2 & \text{if } (x,y)\in[-1,1]\times[-1,1] \\ \left(y^2 - x^2\right)\exp(y+1) + \left[x^2 + \left(\dfrac{y+2}{2}\right)^2\right](y+1) & \text{if } (x,y)\in[-1,1]\times[-3,-1] \end{cases}$$

Step 1: create saddle surface $z = y^2 - x^2$ on $[-1,1]\times[-1,1]$

We divide $x\in[-1,1]$ into 5 points and $y\in[-1,1]$ into 5 points, then mesh plot is shown in Figure 4. It is just piecewise linear approximation.

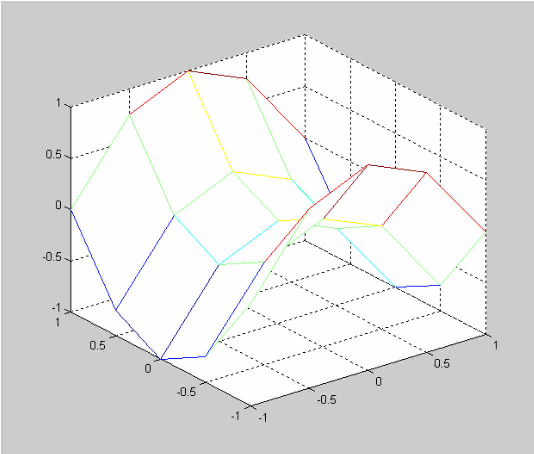

Figure 4: mesh plot of 25 points of saddle plane

Now we use mefisto to construct saddle surface by the same 25 points

Since we have function form, $z = y^2 - x^2$, we can generate 25 points by matlab and output to the mefisto format.

```
1; 1;     {POINTS}
 p1 ; i; 1;   -1.000000 ;   -1.000000 ;    0.000000 ;
 p2 ; i; 1;   -0.500000 ;   -1.000000 ;    0.750000 ;
 p3 ; i; 1;    0.000000 ;   -1.000000 ;    1.000000 ;
 p4 ; i; 1;    0.500000 ;   -1.000000 ;    0.750000 ;
 p5 ; i; 1;    1.000000 ;   -1.000000 ;    0.000000 ;
 p6 ; i; 1;   -1.000000 ;   -0.500000 ;   -0.750000 ;
 p7 ; i; 1;   -0.500000 ;   -0.500000 ;    0.000000 ;
 p8 ; i; 1;    0.000000 ;   -0.500000 ;    0.250000 ;
 p9 ; i; 1;    0.500000 ;   -0.500000 ;    0.000000 ;
 p10 ; i; 1;    1.000000 ;   -0.500000 ;   -0.750000 ;
 p11 ; i; 1;   -1.000000 ;    0.000000 ;   -1.000000 ;
 p12 ; i; 1;   -0.500000 ;    0.000000 ;   -0.250000 ;
 p13 ; i; 1;    0.000000 ;    0.000000 ;    0.000000 ;
 p14 ; i; 1;    0.500000 ;    0.000000 ;   -0.250000 ;
 p15 ; i; 1;    1.000000 ;    0.000000 ;   -1.000000 ;
 p16 ; i; 1;   -1.000000 ;    0.500000 ;   -0.750000 ;
 p17 ; i; 1;   -0.500000 ;    0.500000 ;    0.000000 ;
 p18 ; i; 1;    0.000000 ;    0.500000 ;    0.250000 ;
 p19 ; i; 1;    0.500000 ;    0.500000 ;    0.000000 ;
```

```
 p20 ; i; 1;      1.000000 ;      0.500000 ;    -0.750000 ;
 p21 ; i; 1;    -1.000000 ;      1.000000 ;      0.000000 ;
 p22 ; i; 1;    -0.500000 ;      1.000000 ;      0.750000 ;
 p23 ; i; 1;      0.000000 ;      1.000000 ;      1.000000 ;
 p24 ; i; 1;      0.500000 ;      1.000000 ;      0.750000 ;
 p25 ; i; 1;      1.000000 ;      1.000000 ;      0.000000 ;
```

Step 2: construct surface by polynomial approximation

```
3; { SURFACES faces of the object }
  sf1; { NAME of the SURFACE    }
  i; { NAME of TRANSFORMATION }
  3; { SPLINE QUADRANGLE BY INTERPOLATION POINTS }
  10; { NBAXQB 'number of edges of first parameter U' entier ; }
// this number is independent of initial points, since initial points may not be vetices of
// elements
  10; { NBAYQB 'number of edges of second parameter V' entier ; }
  1; { RAGXQB 'geometrical ratio between the U-vertices' reel ; }
  1; { RAGYQB 'geometrical ratio between the V-vertices' reel ; }
  2; { LDEXSB 'U-degree of polynomials' entier ; }
  2; { LDEYSB 'V-degree of polynomials' entier ; }
  5; { NBPIEX 'number of U interpolation points' entier; }
  5; { NBPIEY 'number of V interpolation points' entier; }
// total number of points to be interpolated is 5x5 = 25
  p1 ; p2 ; p3 ; p4 ; p5;
  p6 ; p7 ; p8 ; p9 ; p10;
  p11 ;p12;p13;p14;p15;
  p16;p17;p18;p19;p20;
  p21;p22;p23;p24;p25;
```

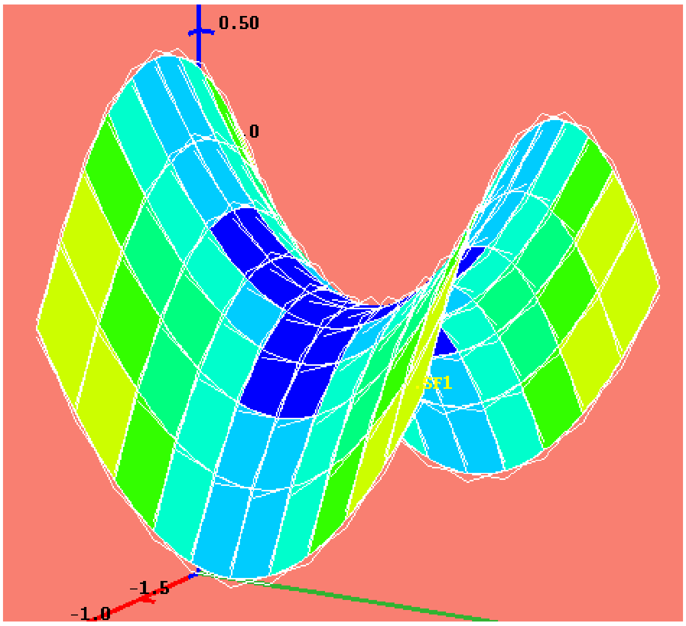

Figure 5: saddle surface is approximated by polynomial of degree 2. It is more smoother than mesh plot of matlab, see Figure 4.

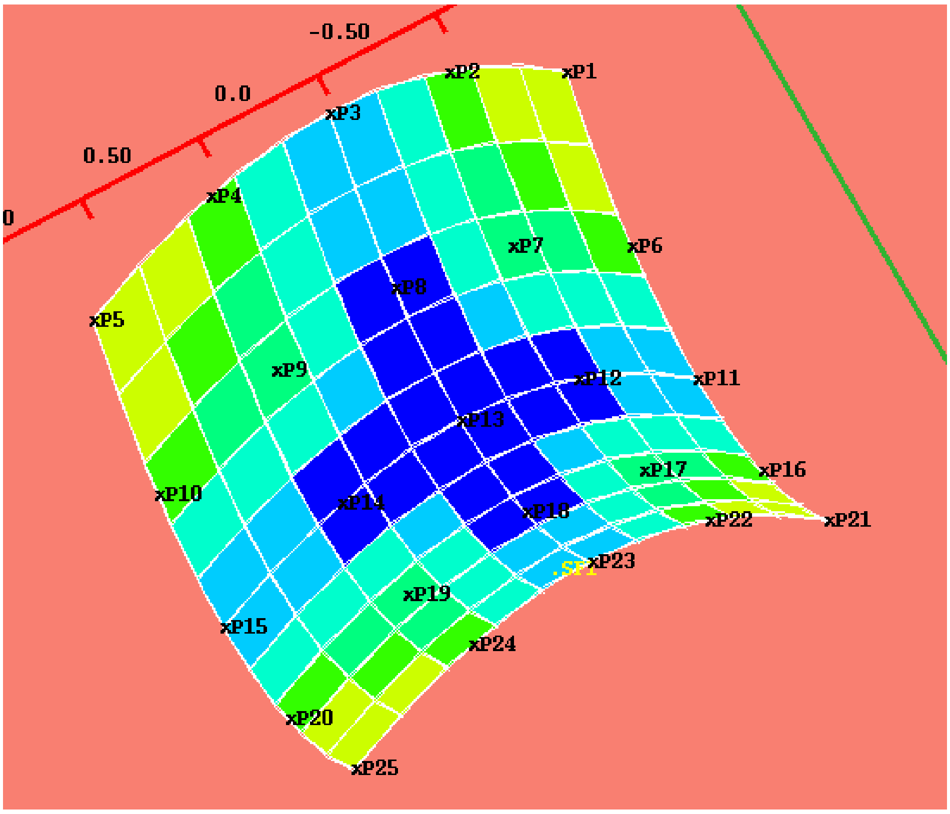

Figure 6: quadrilateral mesh on saddle surface with 10 segment of each direction, you can see that initial 25 points are not vertices of elements. This is good.

Step 3: triangulation of surface sf1 (optional)

3; { SURFACES faces of the object }
sf1_tri; { NAME of the SURFACE or ESCAPE? }
i; { NAME of TRANSFORMATION }
31; { TRIANGULATION OF A QUADRANGULATION }
sf1; { NAME of SURFACE }
4; { max of min of triangle qualities }

Step 4: create second piece of top surface

$z = (y^2 - x^2)\exp(y+1) + \left[x^2 + \left(\frac{y+2}{2}\right)^2\right](y+1)$ on $(x,y) \in [-1,1] \times [-3,-1]$ such that the surface matches previous surface $z = y^2 - x^2$ at $y = -1$. The graph of two surfaces in Matlab is (see Figure 7)

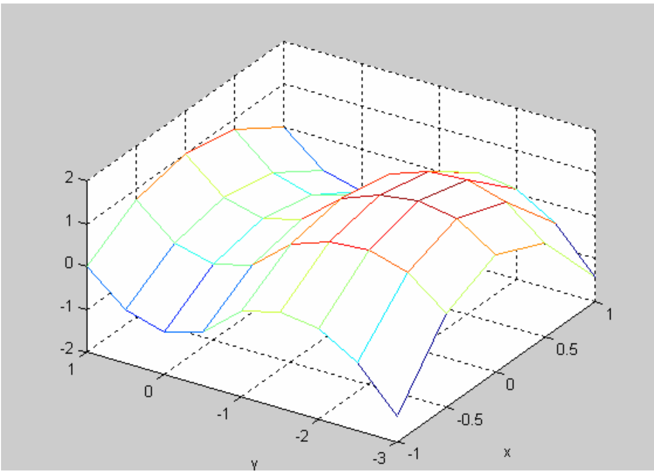

Figure 7: combine two surfaces which are continuous at y = -1

If the two surfaces are interpolated by polynomial of degree 2, and 10 segments for each direction, then mefisto would match interface at y = -1 automatically, amazing !!! (see Figure 8).

Also it is intuitive that if we set one surface with 15 segments for each direction and set the other surface with only 10 segments in each direction, then they cannot match interface at y = -1.

---

h1 ; i; 1; -1.000000 ; -3.000000 ; -1.417318 ;
h2 ; i; 1; -0.500000 ; -3.000000 ; 0.184184 ;
h3 ; i; 1; 0.000000 ; -3.000000 ; 0.718018 ;

h4 ; i; 1;	0.500000 ;	-3.000000 ;	0.184184 ;
h5 ; i; 1;	1.000000 ;	-3.000000 ;	-1.417318 ;
h6 ; i; 1;	-1.000000 ;	-2.500000 ;	-0.422317 ;
h7 ; i; 1;	-0.500000 ;	-2.500000 ;	0.870031 ;
h8 ; i; 1;	0.000000 ;	-2.500000 ;	1.300814 ;
h9 ; i; 1;	0.500000 ;	-2.500000 ;	0.870031 ;
h10 ; i; 1;	1.000000 ;	-2.500000 ;	-0.422317 ;
h11 ; i; 1;	-1.000000 ;	-2.000000 ;	0.103638 ;
h12 ; i; 1;	-0.500000 ;	-2.000000 ;	1.129548 ;
h13 ; i; 1;	0.000000 ;	-2.000000 ;	1.471518 ;
h14 ; i; 1;	0.500000 ;	-2.000000 ;	1.129548 ;
h15 ; i; 1;	1.000000 ;	-2.000000 ;	0.103638 ;
h16 ; i; 1;	-1.000000 ;	-1.500000 ;	0.226913 ;
h17 ; i; 1;	-0.500000 ;	-1.500000 ;	1.056811 ;
h18 ; i; 1;	0.000000 ;	-1.500000 ;	1.333444 ;
h19 ; i; 1;	0.500000 ;	-1.500000 ;	1.056811 ;
h20 ; i; 1;	1.000000 ;	-1.500000 ;	0.226913 ;
h21 ; i; 1;	-1.000000 ;	-1.000000 ;	0.000000 ;
h22 ; i; 1;	-0.500000 ;	-1.000000 ;	0.750000 ;
h23 ; i; 1;	0.000000 ;	-1.000000 ;	1.000000 ;
h24 ; i; 1;	0.500000 ;	-1.000000 ;	0.750000 ;
h25 ; i; 1;	1.000000 ;	-1.000000 ;	0.000000 ;

note that p1=h21, p2 = h22, p3=h23, p4=h24, p5=h25.

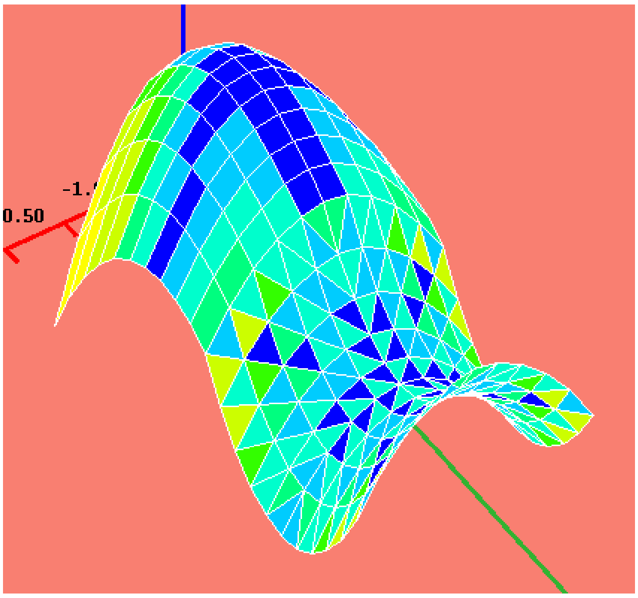

Figure 8: two surface of the same polynomial of degree 2 as interpolation criteria. They match at interface y = -1 vey well.

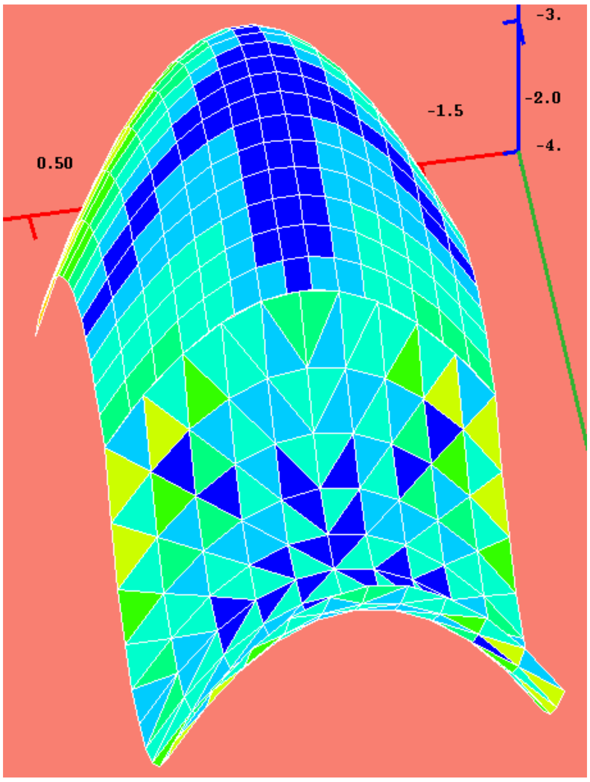

Figure 9: one surface has 15 segments in each direction whereas the other has only 10 segments. They cannot match interface at y = -1.

The same unmatched interface also occurs when degrees of interpolation are different, see Figure 10.

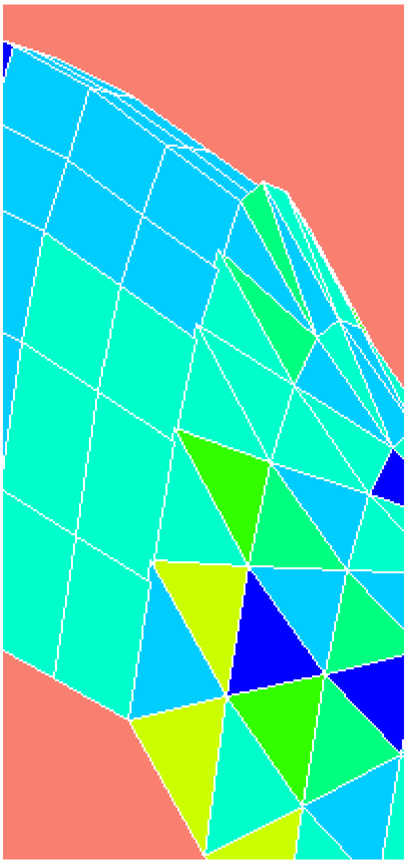

Figure 10: one surface has polynomial of degree 1 whereas the other has polynomial of degree 2. They cannot match interface at y = -1.

Step 5: create bottom surface at z = -3

Figure 11 shows configuration of bottom surface, as usual, we create 4 points b1, b2, b3 and b4 first, then 4 edges b1b2, b2b3, b3b4 and b4b1, then surface sf3 which is enclosed by 4 edges.

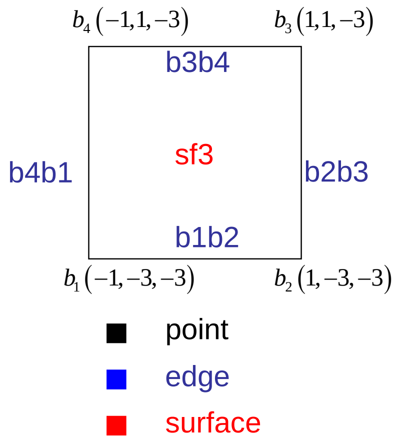

Figure 11: configuration of bottom surface (plane) at z = -3.

```
1; 1;      {POINTS}
b1 ; i; 1;    -1.0 ; -3.0 ; -3    ;
b2 ; i; 1;      1.0 ; -3.0 ; -3    ;
b3 ; i; 1;      1.0 ; 1.0 ; -3 ;
b4 ; i; 1;    -1.0 ; 1.0 ; -3 ;

2; { LINES edges of the object }
b1b2; i; 2; 10; 1; b1; b2;
b2b3; i; 2; 20; 1; b2; b3;
b3b4; i; 2; 10; 1; b3; b4;
b4b1; i; 2; 20; 1; b4; b1;

3; { SURFACES faces of the object }
sf3; { NAME of the SURFACE or ESCAPE? }
i; { NAME of TRANSFORMATION }
1; { TRANSFINITE QUADRANGLE }
b1b2;b2b3;b3b4;b4b1; { NAME of LINE }
1; { regular mesh }
```

Step 6: create 4 walls (plane) between top surface and bottom surface

There are 4 additional edges needed to be set before we construct 4 walls, see Figure 12.

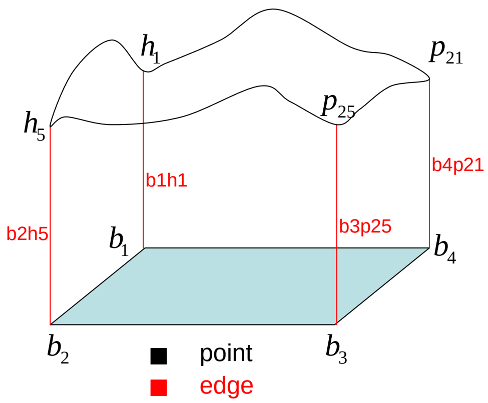

Figure 12: 4 walls between top surface and bottom surface. We need to create additional red edges to construct 4 walls.

2; { LINES edges of the object }

b1h1; i; 2; 6; 1; b1; h1;
b2h5; i; 2; 7; 1; b2; h5;
b3p25; i; 2; 10; 1; b3; p25;
b4p21; i; 2; 10; 1; b4; p21;

Second, create 2 walls which are composed by 4 edges, see Figure 13.

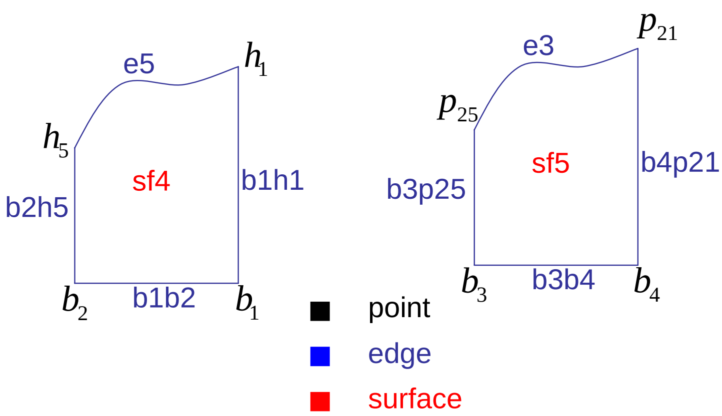

Figure 13: two walls are composed by 4 edges (3 are straight line and 1 is B-spline interpolation).

3; { SURFACES faces of the object }
sf4; { NAME of the SURFACE or ESCAPE? }

i; { NAME of TRANSFORMATION }
1; { TRANSFINITE QUADRANGLE }
b1b2; b1h1; e5; b2h5; { NAME of LINE }
1; { regular mesh }
sf5; { NAME of the SURFACE or ESCAPE? }
i; { NAME of TRANSFORMATION }
1; { TRANSFINITE QUADRANGLE }
b3b4; b4p21; e3; b3p25; { NAME of LINE }
1; { regular mesh }

Third, remember that top surface is composed of the two surfaces, so the remaining two walls have 5 edges, see Figure 14, If we want to use command TRNASFINITE QUADRANGLE, then it is necessary to merge edges e4 and e8 to e4e8 and merge e2 and e6 into e2e6.

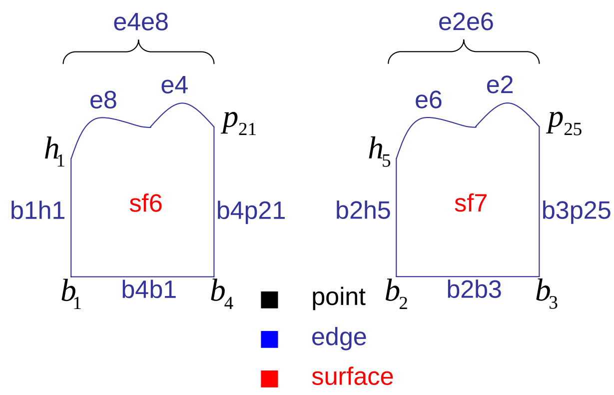

Figure 14: we need to merge edge e4 and e8 to e4e8 and merge edge e2 and e6 to e2e6, then these two walls are the same as Figure 13.

2; { LINES edges of the object }
e4e8; { NAME of the LINE or ESCAPE? }
i; { NAME of TRANSFORMATION }
51; { UNION OF SEVERAL LINES }
2; { NBLIUN 'number of lines of the union' entier ; }
e4;e8; { NAME of LINE }
e2e6; { NAME of the LINE or ESCAPE? }
i; { NAME of TRANSFORMATION }

51; { UNION OF SEVERAL LINES }
2; { NBLIUN 'number of lines of the union' entier ; }
e2;e6; { NAME of LINE }
@;
3; { SURFACES faces of the object }
sf6; { NAME of the SURFACE or ESCAPE? }
i; { NAME of TRANSFORMATION }
1; { TRANSFINITE QUADRANGLE }
b4b1; b4p21; e4e8; b1h1; { NAME of LINE }
1; { regular mesh }
sf7; { NAME of the SURFACE or ESCAPE? }
i; { NAME of TRANSFORMATION }
1; { TRANSFINITE QUADRANGLE }
b2b3; b3p25; e2e6; b2h5; { NAME of LINE }
1; { regular mesh }

Step 7: continuously union 7 surface into a closed surface

We use command 51; C0-CONTINUITY UNION OF SURFACES to union sf1 ~ sf7 to a closed surface.

3; { SURFACES faces of the object }
sf8; i; 51; 7; sf1; sf2; sf3; sf4; sf5; sf6; sf7;

Step 8: triangulation of sf8

The 7 surface are constructed by command either 1; TRNASFINITE QUADRANGLE or 3; SPLINE QUADRANGLE BY INTERPOLATION POINTS, most of them are quadrilateral mesh. We use command 31; TRIANGULATION OF A QUADRANGULATION to convert 7 surface to be triangular mesh.

3; { SURFACES faces of the object }
sf8_tri; i; 31; sf8; 4; {max of min of triangle qualities}

Step 9: create volume enclosed by surface sf8 and do tetrahedrization

We use option 8; 1 CLOSED VOLUME FOR THE OPTION 9 to create volume solid1v8 enclosed by sf8_tri, then use option 9; TETRAHEDRIZATION OF

VOLUMES OF TYPE 8 to do tetrahedrization.

{ /* 8; 1 CLOSED VOLUME FOR THE OPTION 9 */ }
solid1v8; i; 8; 1; sf8_tri;
{ /* 9; TETRAHEDRIZATION OF VOLUMES OF TYPE 8 */ }
solid1t; i; 9; 100000; 2; 5; 0; 1; solid1v8; {Tetrahedrization}

After you execute script mesh, you can plot volume solid1v8 as Figure 15. There are several red color triangles on the surface, it means that the quality of tetrahedrons is not good, for detailed report, you can see message in shell, which is dumped by mefisto when do tetradedrization, see Figure 16.

There is some information in Figure 16

(1) There is one element, numbered as 476, with zero quality, this is unacceptable.
(2) There are 6% elements whose quality factor is less than 0.4, we need to improve these bad tetrahedrons.

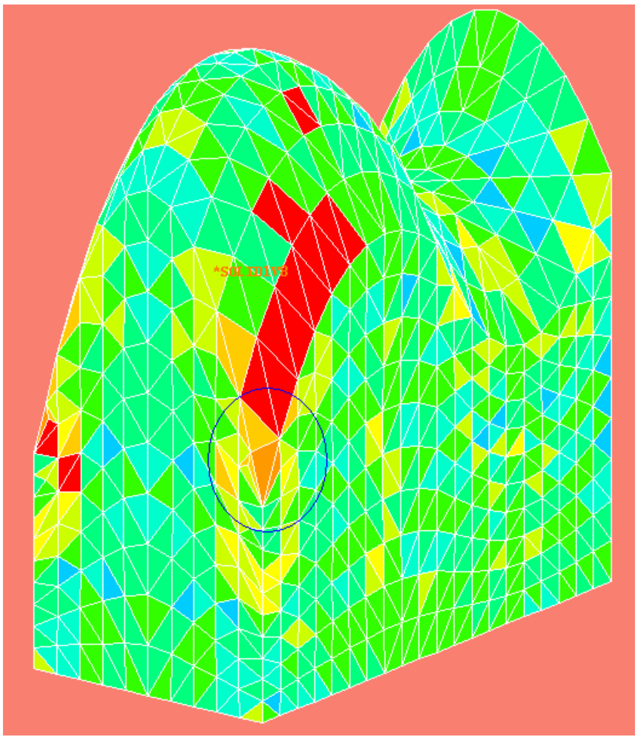

Figure 15: volume solid1v8 has several worse tetrahedron (red color) and bad triangulation on surface (see triangle enclosed by blue line).

```
QUALITY of the MESH : VOLUME SOLID1V8
---------------------
NUMBER of VERTICES         =   1440
NUMBER of TANGENTS         =      0
NUMBER of FINITE ELEMENTS =   5568
NUMBER of FE WITH TANGENTS=      0
AVERAGE QUALITY of FE      = 0.618
MINIMUM QUALITY of FE      =  0.00      for the FE      476
ECART TYPE / 1   of FE     = 0.411

0.900=< QUALITY <1.000 :   292 FE   5 % : **
0.800=< QUALITY <0.900 :   193 FE   3 % : *
0.700=< QUALITY <0.800 :   743 FE  13 % : ******
0.600=< QUALITY <0.700 :  1922 FE  35 % : *****************
0.500=< QUALITY <0.600 :  1382 FE  25 % : ************
0.400=< QUALITY <0.500 :   667 FE  12 % : ******
0.300=< QUALITY <0.400 :   240 FE   4 % : **
0.200=< QUALITY <0.300 :    74 FE   1 % : +
0.100=< QUALITY <0.200 :    28 FE   1 % : +
0.010=< QUALITY <0.100 :    19 FE   0 % : .
0.001=< QUALITY <0.010 :     4 FE   0 % : .
0.000=< QUALITY <0.001 :     4 FE   0 % : .

FE    476 of MINIMUM QUALITY      0.000
VERTEX   1436 X= -0.8719684     Y=  -3.000000     Z=  -1.129834
VERTEX    584 X= -0.7898140     Y=  -3.000000     Z=  -1.410196
VERTEX    595 X= -0.7922392     Y=  -3.000000     Z=  -1.161476
VERTEX    141 X=  -1.000000     Y=  -3.000000     Z=  -1.417318
```

Figure 16: quality report of elements in shell.

What can we do to improve the quality of tetrahedron with minimum effort? Here we propose a method which doesn't change configuration of surface, just modify number of segments of each edge.

Remember that we have set edge b1h1 and b2h2 as

b1h1; i; 2; 10; 1; b1; h1;

b2h5; i; 2; 10; 1; b2; h5;

This setting says that edges b1h1 and b2h5 have 10 segments on them, now we change them to be

b1h1; i; 2; 6; 1; b1; h1;

b2h5; i; 2; 7; 1; b2; h5;

Then we have another result

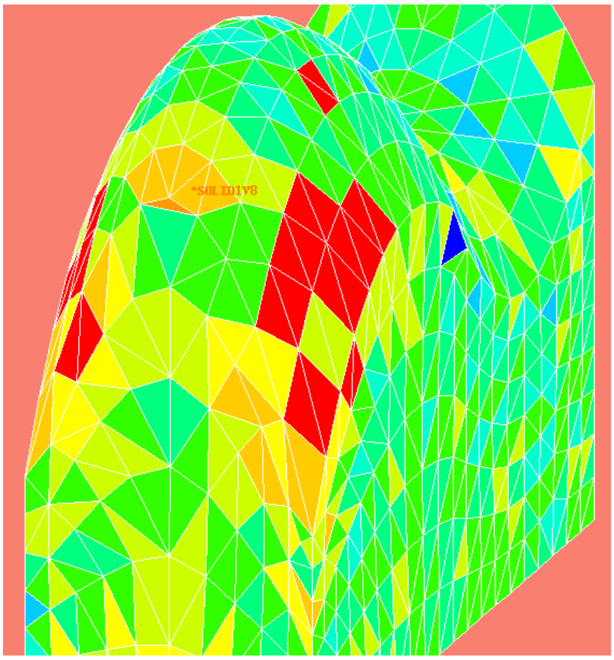

Figure 17: local relaxation of edges.

Second, mefisto provide a quality improvement mechanism for **one volume** (this option is not proper for example quantum-dot, since it is two-volume geometry).

(1) Choose command **4 : VOLUMES materials of the object**
(2) type v; // name of volume
(3) type I; // identity map
(4) 30; QUALITY IMPROVEMENT OF A TETRAHEDRON
(5) type solid1v8; // target volume which is to be improved
(6) type **0.3;** // lower bound of quality factor, see Figure 18

Figure 18: lower bound of quality factor

(7) type **3;** // threshold to determine if two triangles are co-planar. See Figure 19

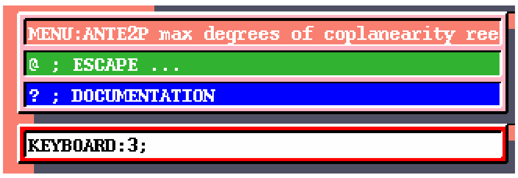

Figure 19: threshold to determine if two triangles are co-planar.

This number 3 denotes angle $\theta = 3^0$ between two normal vectors of adjacent triangular mesh on the surface. Consider two elements FE_1 and FE_2 in Figure 20, if angle $\theta \le 3^0$, then mefisto would regard them as co-plane, then re-split them.

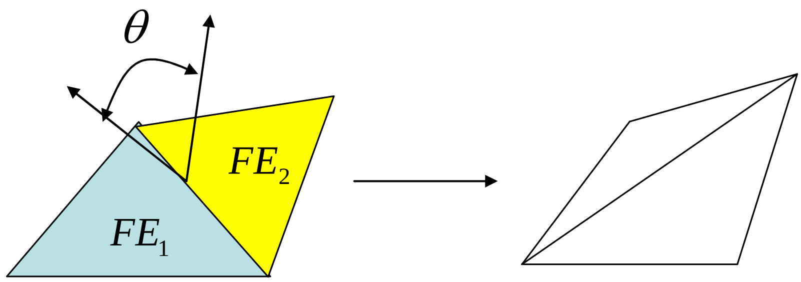

Figure 20: merge two elements FE_1 and FE_2, then re-split into 2 elemetns.

(8) type **10;** // number of iterations, see Figure 21.

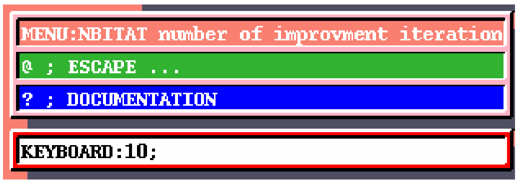

Figure 21: number of iterations, mefisto cannot guarantee desired result in finite iterations.

Then we have better result as Figure 22, moreover if we look at the quality factor of volume v, then it shows evidence that volume v is better than volume solid1v8

(1) Although solidv8 has better average quality 0.618 > 0.609 of volume v, volume v has lower bound of quality 0.2 > 0 of solidv8

(2) If we consider threshold of quality as 0.3, then volume solidv8 and v both have the same ratio 6% of elements whose quality is less then 0.3.

These two methods show us that if the quality of tetrahedrons are not good, we don't need to re-configure our geometry but just change some parameters and then do quality improvement if only one volume in your setting, then we may obtain better result. However this process cannot be automatic execution by mefisto. The users must determine the parameters by themselves.

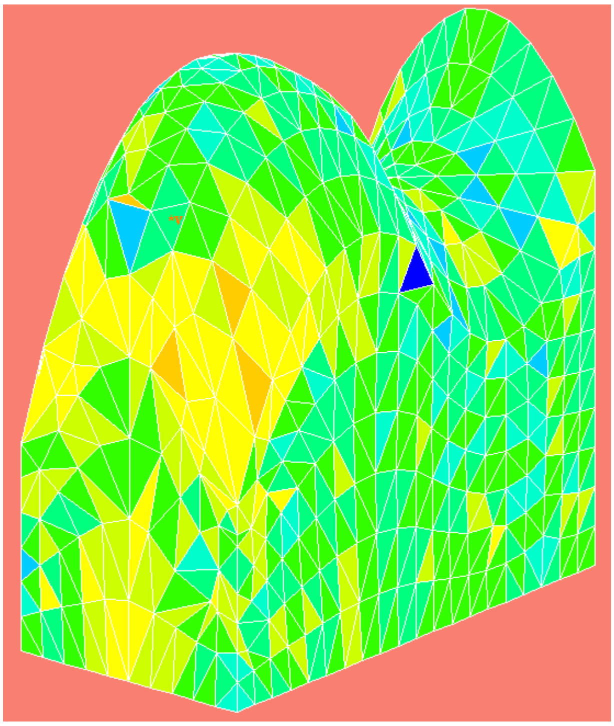

Figure 22: volume v has better result since we don't see any red color.

```
QUALITY of the MESH : VOLUME V
---------------------
NUMBER of VERTICES        =   1168
NUMBER of TANGENTS        =      0
NUMBER of FINITE ELEMENTS =   4373
NUMBER of FE WITH TANGENTS=      0
AVERAGE QUALITY of FE      = 0.609
MINIMUM QUALITY of FE      = 0.288     for the FE    2850
ECART TYPE / 1  of FE      = 0.416

0.900=< QUALITY <1.000 :   193 FE   4 % : **
0.800=< QUALITY <0.900 :   132 FE   3 % : *
0.700=< QUALITY <0.800 :   517 FE  12 % : ******
0.600=< QUALITY <0.700 :  1423 FE  33 % : ****************
0.500=< QUALITY <0.600 :  1166 FE  27 % : *************
0.400=< QUALITY <0.500 :   665 FE  15 % : *******
0.300=< QUALITY <0.400 :   271 FE   6 % : ***
0.200=< QUALITY <0.300 :     6 FE   0 % : .

FE   2850 of MINIMUM QUALITY     0.288
VERTEX     563 X=  0.4292184     Y=  -3.000000     Z= -0.1979911
VERTEX     551 X=  0.2036884     Y=  -3.000000     Z= -0.5655373
VERTEX     160 X=  0.7999998     Y=  -2.800000     Z= -0.2547446
VERTEX     564 X=  0.5314252     Y=  -3.000000     Z= -0.7435725
```

Figure 23: Although average quality is a little bit worse, lower bound of quality factor is 2%, this number is better than 0% of original volume solid1v8.