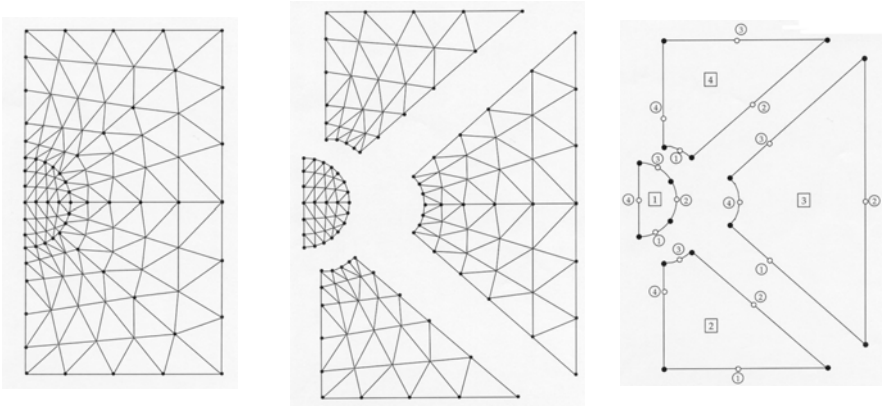
***MESHGEN – FEM mesh generator***

Program **MESHGEN** is an automatic mesh generator which uses quadrilaterals with parabolic sides.

These quadrilaterals may be joined together in any specified manner.

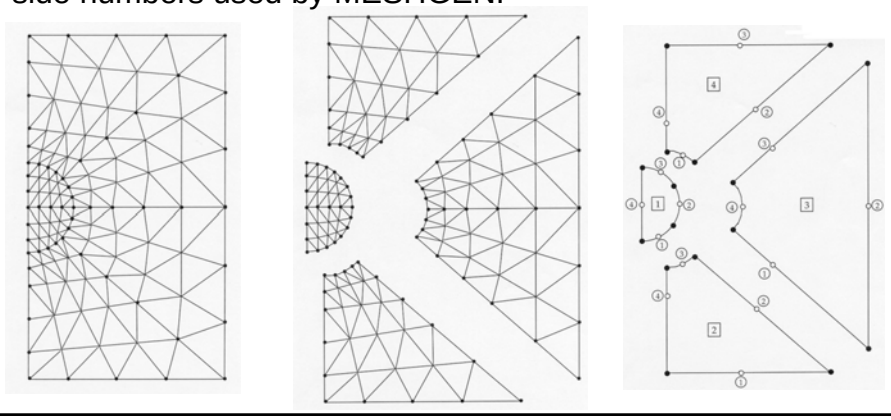


***MESHGEN – FEM mesh generator***

The first figure on the left shows a complete finite element mesh composed of 3-nodded triangular elements.

The middle figure shows an exploded view illustrating the four loops used in this discretization.

The right on the right defines the loop numbers and their side numbers used by MESHGEN.

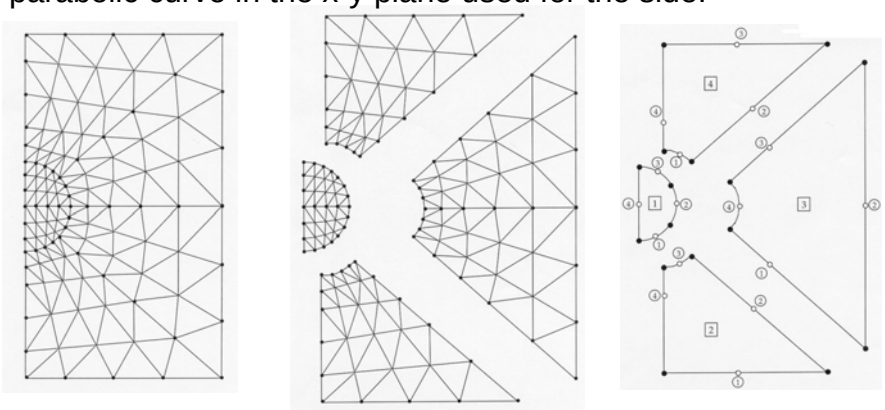


***MESHGEN – FEM mesh generator***

The loop numbering is arbitrary as is the numbering of each side of a loop provided it is done counterclockwise.

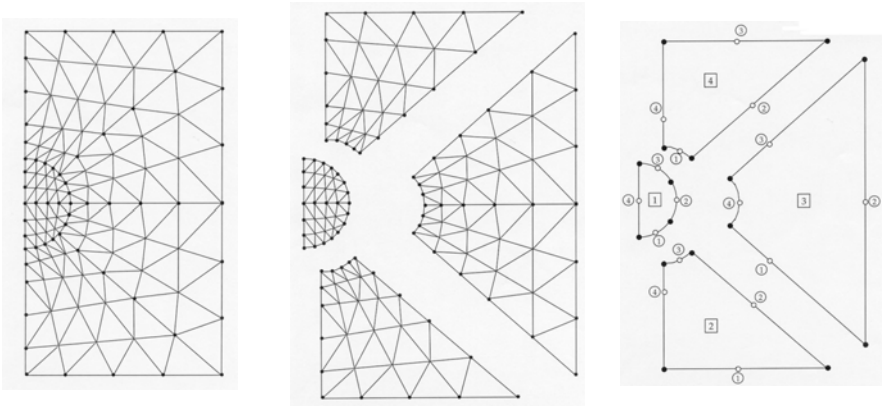
Also shown on the right are the eight points describing the sides of each loop.

The three points along any side are used to define the parabolic curve in the x-y plane used for the side.



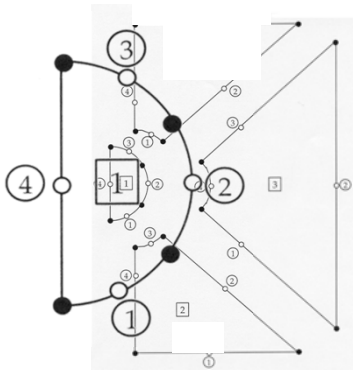
***MESHGEN – FEM mesh generator***

Hence, even though the seven points along sides 1, 2, and 3 of loop 1 all lie in a circle, the actual boundary is made up of three parabolas joined at the corner points (dark circles).



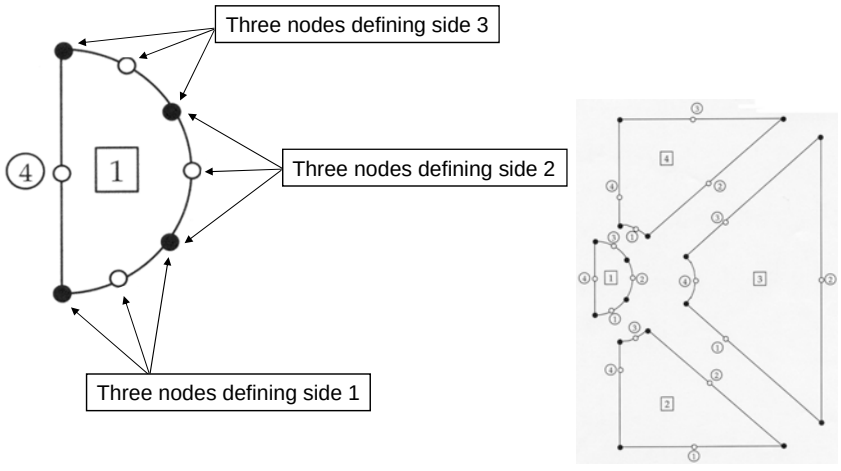
***MESHGEN – FEM mesh generator***

Hence, even though the seven points along sides 1, 2, and 3 of loop 1 all lie in a circle, the actual boundary is made up of three parabolas joined at the corner points (dark circles).




***MESHGEN – FEM mesh generator***

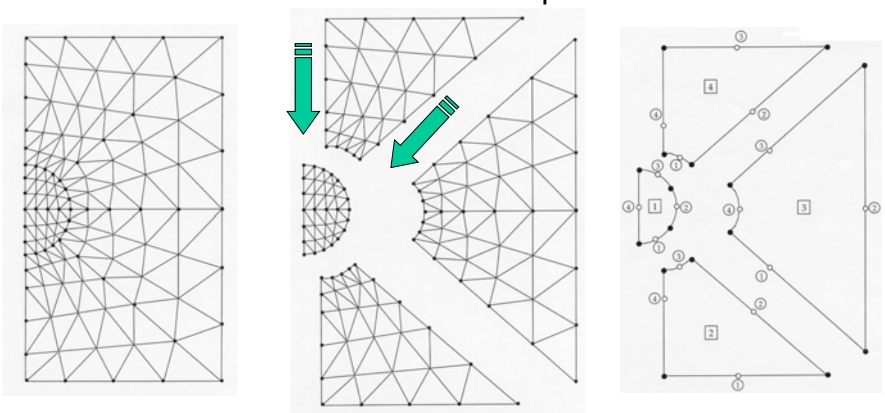
Hence, even though the seven points along sides 1, 2, and 3 of loop 1 all lie in a circle, the actual boundary is made up of three parabolas joined at the corner points (dark circles).



***MESHGEN – FEM mesh generator***

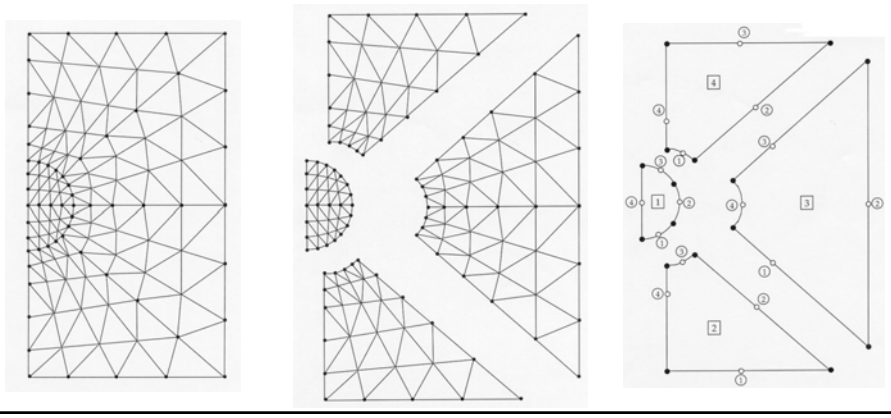
Although three of the 4 sides of loops 2, 3, and 4 are straight lines, the side coordinate points (open circles) do not necessarily fall at the center of these sides.

Notice that they have been moved toward loop 1 and as a consequence the elements shown in the figure on the left have likewise been moved toward loop 1.



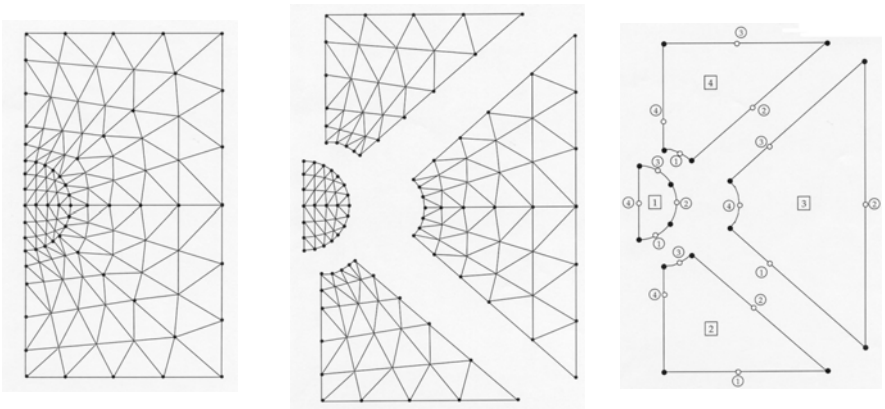
***MESHGEN – FEM mesh generator***

As you gain experience in using MESHGEN you will find that it has the versatility needed to generate a wide variety of meshes.



***MESHGEN – FEM mesh generator***

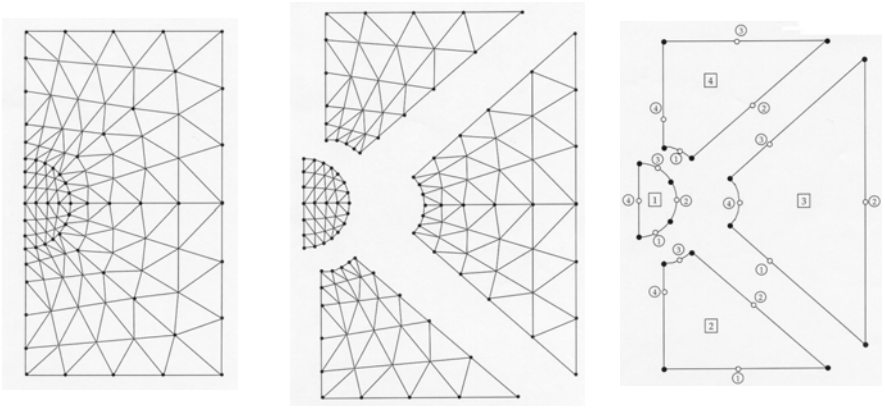
When it cannot be made to produce exactly a desired configuration (for example, nodal points on the boundary of loop 1 not lying in a prefect circle), the user may use MESHGEN to obtain a mesh very close to that which is desired, and then manually adjust whatever data are necessary to achieve the geometry required.



***MESHGEN – FEM mesh generator***

It should be noted that when two loops are joined, the common sides must have the same number of divisions (hence nodal points).

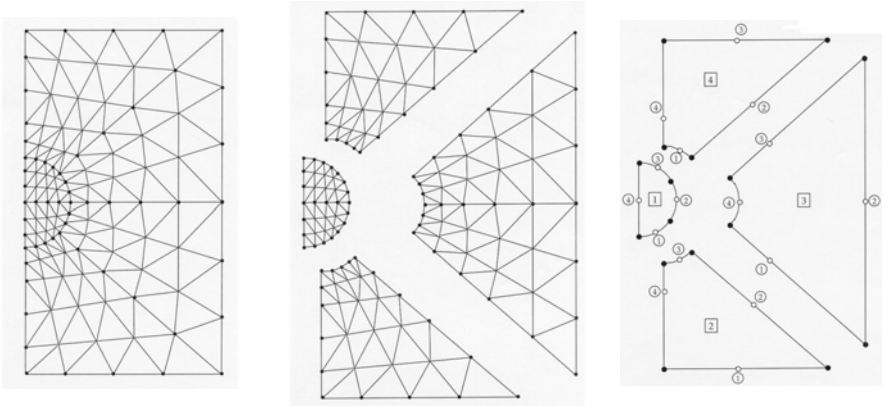
As shown in figure on the right, side 1 of loop 3 is joined to side 2 of loop 2 in the previous example.



***MESHGEN – FEM mesh generator***

The numbering of the elements and nodal points begins in the first loop and continues through the last loop.

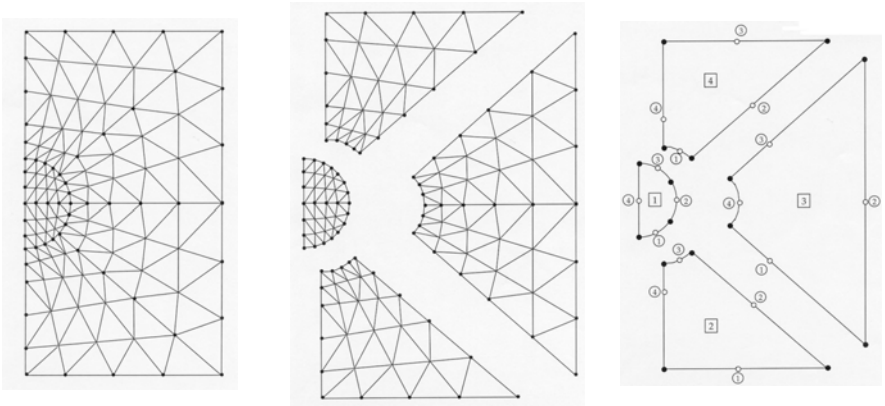
For any given loop the node numbering represents the addition to the last nodal point number and element number assigned by the previous loop.



***MESHGEN – FEM mesh generator***

The diagonal defines the element in each square is chosen to be the shortest of the two distances after mapping to the *x-y* plane.

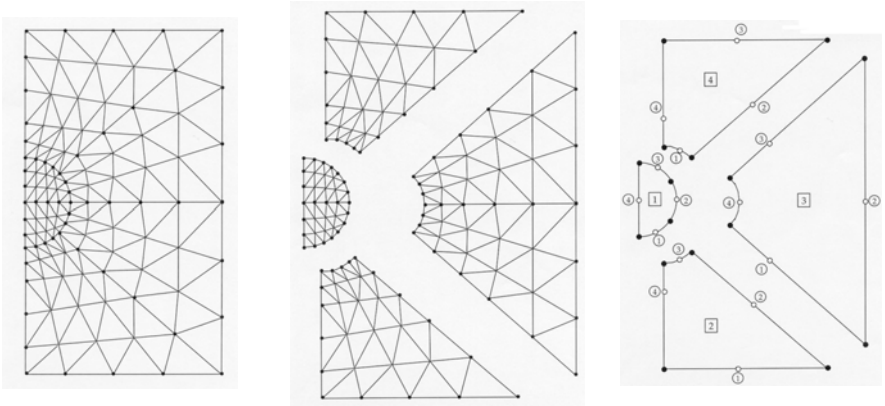
In the case where a series of rectangular elements are created with equal diagonals, MESHGEN will alternate the direction of the diagonal from one element to the next.



***MESHGEN – FEM mesh generator***

One final note before the input file to MESHGEN is described.

The order in which the nodal points are numbered is critical since it determines the band width of the "stiffness" matrix.

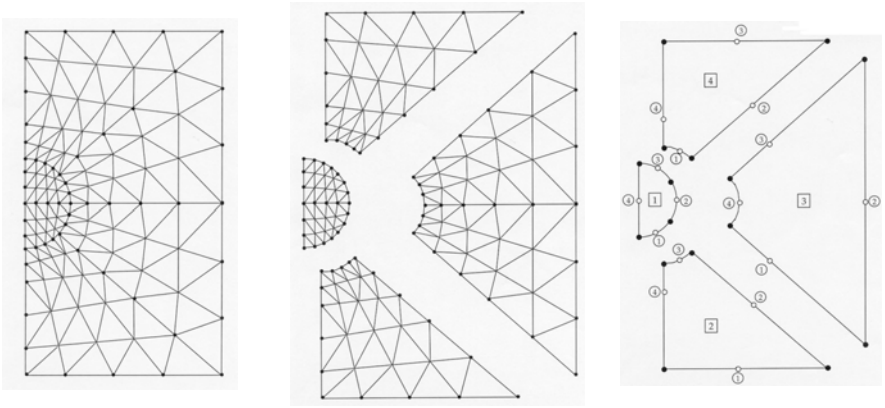


***MESHGEN – FEM mesh generator***

Program MESHGEN can produce a very undesirable node numbering scheme.

When this happens, a new numbering can be developed through the subroutine ORDER.

Use of this subroutine will be described in the next section



## *MESHGEN – FEM mesh generator*

```
%*********************************************************************
%                                                                    *
%                                                                    *
%   FINITE ELEMENT    MESH GENERATOR                                 *
%                                                                    *
%   THIS PROGRAM GENERATES FINTIE ELEMENT MESHES for EITHER           *
%   3 NODE OR 6 NODE TRIANGULAR ELEMENTS                             *
%                                                                    *
%                                                                    *
%   NNPE          = NUMBER OF NODES PER ELEMENT (MUST BE 3 OR 6)     *
%   NUMLPS        = NUMBER OF LOOPS:BE USED for CURRENT MESH         *
%   NDIV(I,J)     = NUMBER OF DIVISIONS ON SIDE J OF LOOP I          *
%                       J=1 REFERS:SIDE DEFINED BY FIRST THREE       *
%                           (XCOR,YCOR). WILL AUTOMATICALLY          *
%                           DESIGNATE SIDE 3 AS HAVING THE SAME      *
%                           NUMBER OF DIVISIONS                      *
%                                                                    *
%                       J=2 REFER:SIDE DEFINED BY (XCOR,YCOR)        *
%                           2, 3 AND 4. WILL DESIGNATE SIDE 4 AS     *
%                           HAVING THE SAME NUMBER OF DIVISIONS      *
%                                                                    *
%   JOIN(I,J,K)   = CURRENT LOOP (LOPP I) SIDE K IS JOINED:          *
%                   :SIDE JOIN(I,J,2) OF LOOP (I,J,1)                *
%   XCOR(I,J)     = THE 8 X-COORDINATES OF LOOP I                    *
%   YCOR(I,J)     = THE 8 Y-COORDINATES OF LOOP I                    *
%                                                                    *
% DIMENSIONS OF THE ARRAYS USED IN THIS PROGRAM                      *
%                                                                    *
% MXEE = NUMBER OF ELEMENTS                                          *
% MXLL = NUMBER OF LOOPS                                             *
% MXNN = NUMBER OF NODAL POINTS                                      *
% MXSS = MAXIMUM NUMBER OF NODES ALONG ANY LOOP SIDE                 *
%               MXSS MUST BE EQUAL OR GREATER THAN:                  *
%                  NDIV + 1    for NNPE = 3                          *
%                2*NDIV +1     for NNPE = 6                          *
%                                                                    *
%*********************************************************************
```

## *MESHGEN – FEM mesh generator*

```
mxee = 2000;
mxnn = 2200;
mxss = 99;
mxll = 10;
```

Maximum number of elements

Maximum number of node points

Maximum number of nodes along any loop side

Maximum number of loops

```
xcor(1:mxll,1:8) = 0;
ycor(1:mxll,1:8) = 0;
xord(1:mxnn) = 0;
yord(1:mxnn) = 0;
rn(1:8) = 0;

npn(1:mxnn)             = 0;
np(1:mxee,1:6)          = 0;
ndiv(1:mxll,1:4)        = 0;
joinn(1:mxll,1:4,1:2) = 0;
lnr(1:4,1:mxss)         = 0;
lnp(1:6,1:4,1:mxss)     = 0;
```

Dimension arrays

***MESHGEN – FEM mesh generator***

The remaining portion of the code does not require you to make any corrections or adjustments.

The following is an example of nodal and element numbers designated by MESHGEN for a typical loop.

If this is not the first loop then these numbers will start where the previous nodal point and element numbers left off.

Subroutine ORDER will change the nodal point numbers if IRDER = 1, but will keep the same element numbers.

The printed output will designate the numbering scheme above as the old node point number and those assigned by ORDER as the new node point numbers.

***MESHGEN – FEM mesh generator***

**MESHGEN Input**

The following is the format of the input file read by **MESHGEN**

Variables:

```
NNPE, NUMLPS

    { NDIV(I,1) ,NDIV(I,2)
    { ((JOIN(I,J,K),K+1,2),J=1,8)
    { (XCOR(I,J),YCOR(I,J),J=1,8)
```

**Note:** One set of data is required for each loop

***MESHGEN – FEM mesh generator***

**MESHGEN Input**

`NNPE` = number of nodes per element (must be either 3 or 6)
`NUMLPS` = number of loops

```
NNPE, NUMLPS

    { NDIV(I,1) ,NDIV(I,2)
    { ((JOIN(I,J,K),K+1,2),J=1,8)
    { (XCOR(I,J),YCOR(I,J),J=1,8)
```

**Note:** One set of data is required for each loop

***MESHGEN – FEM mesh generator***

**MESHGEN Input**

`NDIV(I,1), NDIV(I,2)` = the number of divisions on sides 1 and 2 of loop **`I`**

```
NNPE, NUMLPS

    { NDIV(I,1) ,NDIV(I,2)
    { ((JOIN(I,J,K),K+1,2),J=1,8)
    { (XCOR(I,J),YCOR(I,J),J=1,8)
```

**Note:** One set of data is required for each loop

***MESHGEN – FEM mesh generator***

**MESHGEN Input**

`JOIN(I,J,K)` = loop `I`, side `J` is joined to loop `JOIN(I,J,1)` side `JOIN(I,J,2)`

```
NNPE, NUMLPS

{ NDIV(I,1) ,NDIV(I,2)
{ ((JOIN(I,J,K),K+1,2),J=1,8)
{ (XCOR(I,J),YCOR(I,J),J=1,8)
```

**Note:** One set of data is required for each loop

***MESHGEN – FEM mesh generator***

**MESHGEN Input**

`XCOR(I,J),YCOR(I,J)` = the *x* and *y* coordinates of loop `I`'s control point

```
NNPE, NUMLPS

{ NDIV(I,1) ,NDIV(I,2)
{ ((JOIN(I,J,K),K+1,2),J=1,8)
{ (XCOR(I,J),YCOR(I,J),J=1,8)
```

**Note:** One set of data is required for each loop

## *MESHGEN – FEM mesh generator*

### MESHGEN Input

The following is the input file read by MESHGEN for the example problem shown in figure below.

```
   3        4
   4        6
 0  0     0  0     0  0     0  0
0.000   -0.500
0.200   -0.458
0.400   -0.300
0.500    0.000
0.400    0.300
0.200    0.458
0.000    0.500
0.000    0.000
```

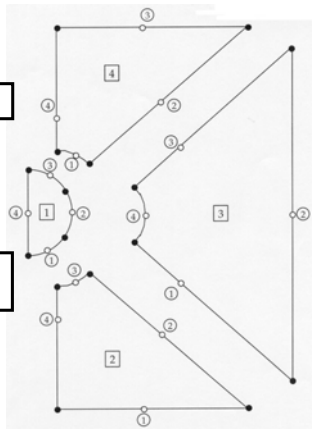
NNPE

NUMPLS

NDIV(1,1), NDIV(1,2)

JOIN(I,J,K)

XCOR(I,J), YCOR(I,J)
For the 8 points defining the loop 1



## *MESHGEN – FEM mesh generator*

### MESHGEN Input

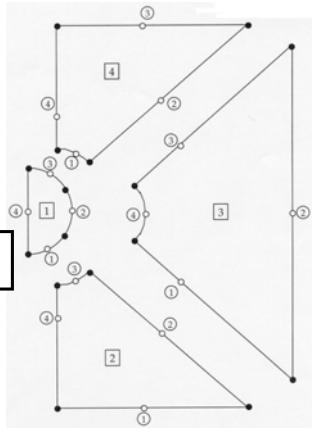
The following is the input file read by MESHGEN for the example problem shown in figure below.

```
   4        5
 0  0     0  0     1  1     0  0
0.000   -2.500
1.200   -2.500
3.000   -2.500
1.180   -0.960
0.400   -0.300
0.200   -0.458
0.000   -0.500
0.000   -1.200
```

NDIV(2,1), NDIV(2,2)

JOIN(I,J,K)

XCOR(I,J), YCOR(I,J)
For the 8 points defining the loop 2



## *MESHGEN – FEM mesh generator*

### MESHGEN Input

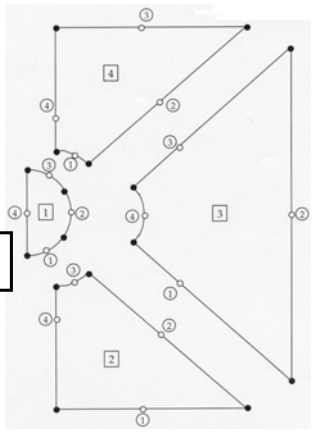
The following is the input file read by MESHGEN for the example problem shown in figure below.

```
    5        6
 2  2     0  0     0  0     1  2
0.400    -0.300
1.180    -0.960
3.000    -2.500
3.000     0.000
3.000     2.500
1.180     0.960
0.400     0.300
0.500     0.000
```

NDIV(3,1), NDIV(3,2)

JOIN(I,J,K)

XCOR(I,J), YCOR(I,J)
For the 8 points defining the loop 3



## *MESHGEN – FEM mesh generator*

### MESHGEN Input

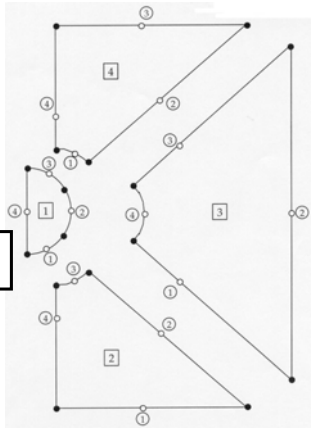
The following is the input file read by MESHGEN for the example problem shown in figure below.

```
    4        5
 1  3     3  3     0  0     0  0
0.000     0.500
0.200     0.458
0.400     0.300
1.180     0.960
3.000     2.500
1.200     2.500
0.000     2.500
0.000     1.200
```

NDIV(4,1), NDIV(4,2)

JOIN(I,J,K)

XCOR(I,J), YCOR(I,J)
For the 8 points defining the loop 4



## *MESHGEN – FEM mesh generator*

### MESHGEN Input

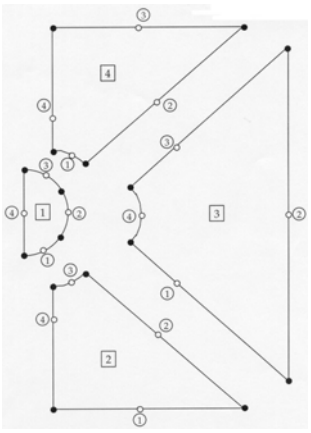
```
    3       4
    4       6
 0  0     0  0     0  0     0  0
0.000   -0.500
0.200   -0.458
0.400   -0.300
0.500    0.000
0.400    0.300
0.200    0.458
0.000    0.500
0.000    0.000
    4       5
 0  0     0  0     1  1     0  0
0.000   -2.500
1.200   -2.500
3.000   -2.500
1.180   -0.960
0.400   -0.300
0.200   -0.458
0.000   -0.500
0.000   -1.200
    5       6
 2  2     0  0     0  0     1  2
0.400   -0.300
1.180   -0.960
3.000   -2.500
3.000    0.000
3.000    2.500
1.180    0.960
0.400    0.300
0.500    0.000
    4       5
 1  3     3  3     0  0     0  0
0.000    0.500
0.200    0.458
0.400    0.300
1.180    0.960
3.000    2.500
1.200    2.500
0.000    2.500
0.000    1.200
```

`XCOR(I,J), YCOR(I,J)`
For the 8 points defining the loop 1

`XCOR(I,J), YCOR(I,J)`
For the 8 points defining the loop 2

`XCOR(I,J), YCOR(I,J)`
For the 8 points defining the loop 3

`XCOR(I,J), YCOR(I,J)`
For the 8 points defining the loop 4



## *MESHGEN – FEM mesh generator*

### MESHGEN Input

```
    3       4
    4       6
 0  0     0  0     0  0     0  0
0.000   -0.500
0.200   -0.458
0.400   -0.300
0.500    0.000
0.400    0.300
0.200    0.458
0.000    0.500
0.000    0.000
    4       5
 0  0     0  0     1  1     0  0
0.000   -2.500
1.200   -2.500
3.000   -2.500
1.180   -0.960
0.400   -0.300
0.200   -0.458
0.000   -0.500
0.000   -1.200
    5       6
 2  2     0  0     0  0     1  2
0.400   -0.300
1.180   -0.960
3.000   -2.500
3.000    0.000
3.000    2.500
1.180    0.960
0.400    0.300
0.500    0.000
    4       5
 1  3     3  3     0  0     0  0
0.000    0.500
0.200    0.458
0.400    0.300
1.180    0.960
3.000    2.500
1.200    2.500
0.000    2.500
0.000    1.200
```

These data are fairly self-explanatory with the possible exception of the JOIN array.

Notice that the input data for this array has four groups of two integers each.

These four groups correspond to the four sides of the loop under consideration.

## *MESHGEN – FEM mesh generator*

### MESHGEN Input

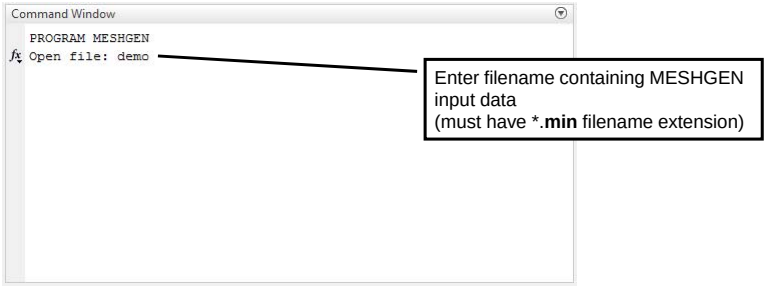
```
    3       4
    4       6
 0  0     0  0     0  0     0  0
0.000    -0.500
0.200    -0.458
0.400    -0.300
0.500     0.000
0.400     0.300
0.200     0.458
0.000     0.500
0.000     0.000
    4       5
 0  0     0  0     1  1     0  0
0.000    -2.500
1.200    -2.500
3.000    -2.500
1.180    -0.960
0.400    -0.300
0.200    -0.458
0.000    -0.500
0.000    -1.200
    5       6
 2  2     0  0     0  0     1  2
0.400    -0.300
1.180    -0.960
3.000    -2.500
3.000     0.000
3.000     2.500
1.180     0.960
0.400     0.300
0.500     0.000
    4       5
 1  3     3  3     0  0     0  0
0.000     0.500
0.200     0.458
0.400     0.300
1.180     0.960
3.000     2.500
1.200     2.500
0.000     2.500
0.000     1.200
```

Thus, for loop 3 the two members of the first group indicate that side 1 is joined to loop 2 side 2.

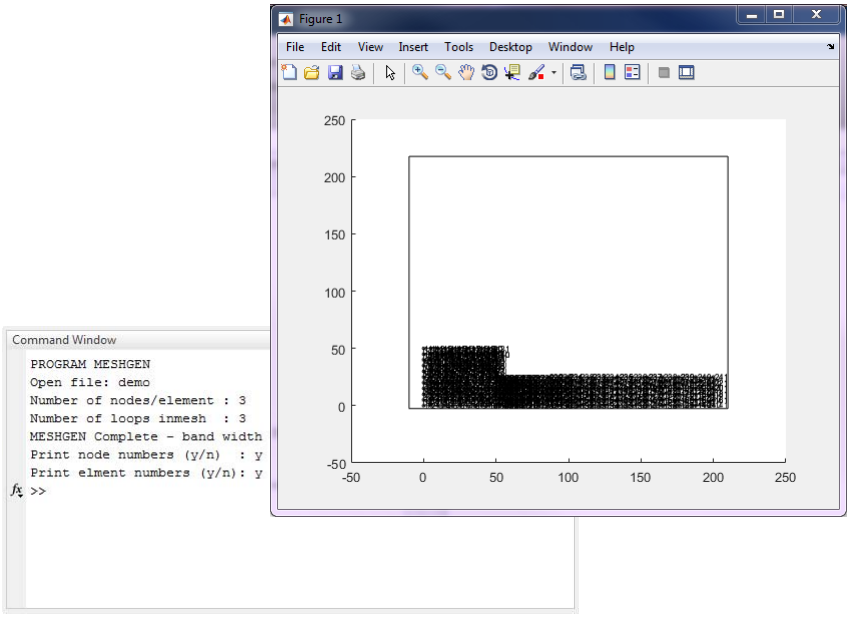
The next two groups being zeros indicate that sides 2 and 3 are not joined to an earlier defined loop.

The last group indicates side 4 of this loop is joined to loop 1, side 2.

## *MESHGEN – FEM mesh generator*

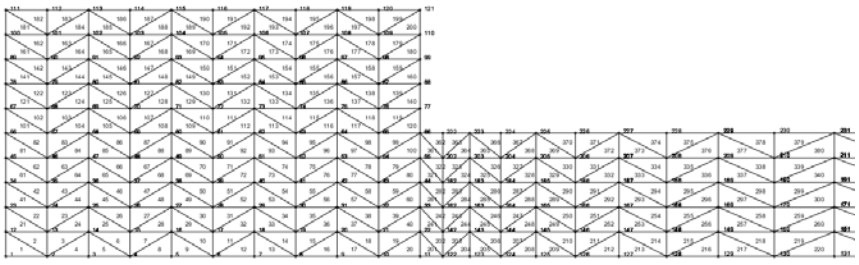



## *MESHGEN – FEM mesh generator*




## *MESHGEN – FEM mesh generator*

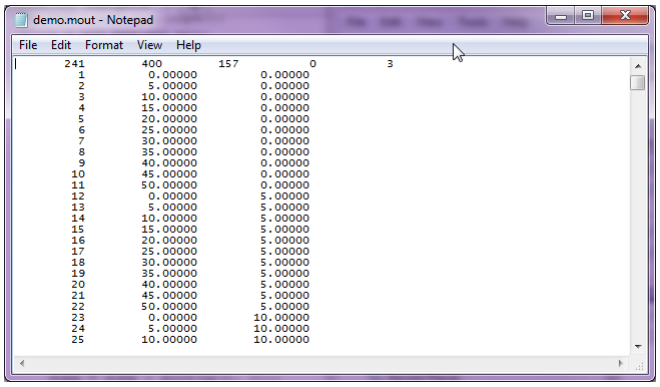
You can zoom in and out of the graphics window to visualize the node and element numbers.



***MESHGEN – FEM mesh generator***

The original MESHGEN output is file ***demo.mout*** defines a mesh with:

241 nodes
400 elements
156 bandwidth




# End of
# MESHGEN Program