

# **SPACE 3D SUBSTRATE EXTRACTION BY USING FEMLAB**

*S. de Graaf*

Circuits and Systems Group  
Faculty of Electrical Engineering,  
Mathematics and Computer Science  
Delft University of Technology  
The Netherlands

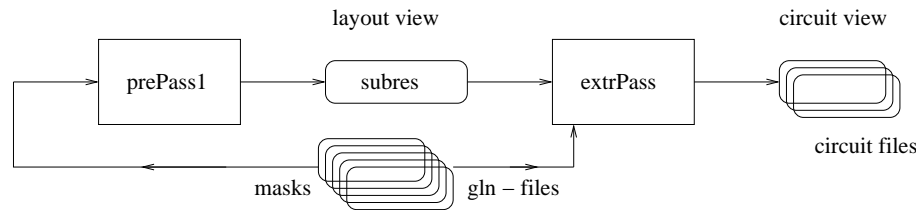
Report EWI-ENS 04-01  
March 26, 2004

Copyright © 2004 by the author.

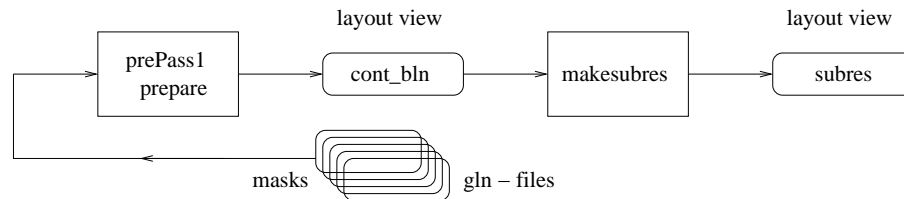
Last revision: December 13, 2004.

## 1. INTRODUCTION

Normally, the *space3d* layout to circuit extractor uses for accurate (3D) substrate resistance extraction (option **-B**) the *makesubres* program to calculate the substrate resistances. See also the "Space 3D Substrate Extraction Application Note" (report EWI-ENS 03-04). For the calculation, the *makesubres* program uses a substrate spider mesh, Green's functions and Schur matrix inversion. The calculated results are written in the cell layout view "subres" file. In the *space3d* extract pass, the results are read again and the substrate resistances are added to the extracted circuit network. See the figure below.



Currently, *makesubres* is a symbolic link to the *space3d* program. In the future, this part may be replaced by a user program. This program must use the same extraction environment, *space* parameter file and technology file and must know the extracted cell name and project path.



Now, as an alternative, FEMLAB (Finite Element Method) can be used by *space3d* to calculate the substrate resistances. See the explanation of the *makefem* program in the following section.

### NOTE:

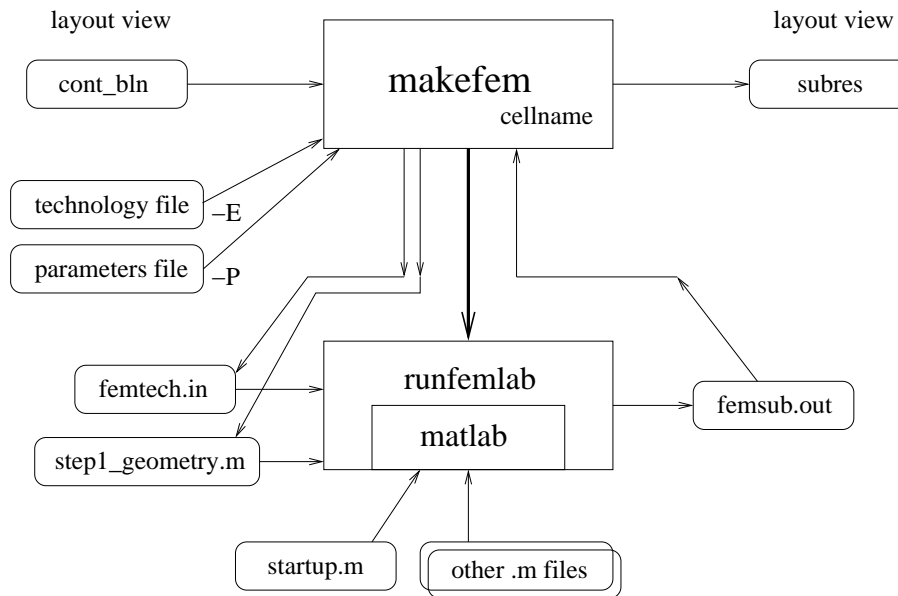
You can use FEMLAB instead of the *makesubres* method, by setting the *makesubres* symbolic link to the *makefem* program.

### NOTE:

The *makefem* program can only be used with *space3d* that supports the new "subres" file format.

## 2. THE MAKEFEM PROGRAM

The *makefem* program reads the substrate contacts and prepares input for using FEMLAB. After running FEMLAB, the output of FEMLAB is converted to stream "subres". For an overview see the following flow chart:



### NOTE:

The *runfemlab* program is a shell script, that defines the *matlab* run command. The "startup.m" file is default read by the *matlab* program and defines for *matlab* what must be done. Note that the "runfemlab" and "startup.m" file must exist in the current working directory.

In appendix A, a listing is given of a number of files. As example, the cell "sub3term" is used (see the "Space Substrate Res. Extraction User's Manual"). Note that the "image part" is also written to stream "subres". These substrate capacitance values can be outputted by *space3d* when option -C is used.

The *makefem* program creates for each substrate tile a 'rect2' or 'poly2' object. A number of these tiles can be one substrate contact. To get a better mesh (see appendix B), the tile objects must be joined together with the union operator '+'. Note that the internal borders must be removed with the 'geomdel' function. However, also this mesh is not yet optimal, because points of internal borders are not removed from the contact boundary.

Appendix B shows the mesh with internal boundary, with and without extra border point.

Appendix C gives a flow chart about how the *makefem* program can be tested.

Appendix D shows the output for one big substrate contact of cell "coilgen".

### 3. MAKEFEM OPTIONS

Command line:

```
makefem [-e def | -E file] [-p def | -P file] [-S param=value] [-hiv] cell
```

Options explanation:

<b>-e,-E</b>	for specifying a technology file (see <i>space</i> )
<b>-p,-P</b>	for specifying a parameter file (see <i>space</i> )
<b>-S</b>	for specifying additional parameters (see <i>space</i> )
<b>-h</b>	for getting help info
<b>-i</b>	for statistical information
<b>-v</b>	for setting verbose mode

### 4. MAKEFEM PARAMETERS

#### 4.1 fem.param\_verbose (default: off)

Prints something verbose about read parameters.

#### 4.2 fem.verbose (default: off)

Verbose mode. Equal to option **-v**.

#### 4.3 fem.print\_time (default: off)

Prints the used time of modules of the *makefem* program.

#### 4.4 elim\_sub\_node (default: off)

Can be put "on" to eliminate the substrate node "SUBSTR".

#### 4.5 fem.xoverlap (default: 20)

Cell boundary x-overlap in microns (must be  $\geq 0$ ).

#### 4.6 fem.yoverlap (default: 20)

Cell boundary y-overlap in microns (must be  $\geq 0$ ).

#### 4.7 fem.zbottom (default: 20)

The depth in microns of the substrate. The position of the substrate backcontact (must be  $> 0$ ). The depth must be greater than the specified layers.

#### 4.8 fem.backcontact (default: on)

Can be put "off", when you don't want to add a backside contact.

**4.9 fem.cs\_mask (no default)**

To generate a channelstop area, the channelstop mask must be specified. The program reads the channelstop area from the cell *mask\_gln* stream.

**4.10 fem.cs\_bbox (no default)**

This bounding box can be used, when the channelstop mask has infinity gln edges. This parameter has four arguments (xl xr yb yt) and must be specified in database units.

**4.11 fem.cs\_extension (default: 0)**

When the channelstop mask has infinity gln edges, the cell bounding box is used to give the channelstop mask a finity dimension. Parameter "fem.cs\_extension" can be used to change the dimension and must be specified in database units. The parameter is not used, when parameter "fem.cs\_bbox" is specified.

**4.12 fem.cs\_sigma (default: 0.0)**

The channelstop sigma is default 100 times the sigma of the first sublayer.

**4.13 fem.cs\_thickness (default: 0.5 micron)**

The thickness of the channelstop area (must be > 0).

**4.14 sub3d.makefem (default: off)**

For the *space3d* program. To use *makefem* in place of the *makesubres* program.

## 5. MAKEFEM ARRAY

The *makefem* array is one bigger than the number of substrate contacts, because one big backcontact is added to the number of contacts. This backcontact is eliminated with Gaussian elimination. The backcontact is the last entry in the array. The values on the diagonal from left-top to bottom-right are positive. The other values are all negative (see the "femsub.out" file). The mean value of each pair of values is taken and the sign is changed. The new value is put in the upper-right array part.

$$g_{0,1} = -(g_{0,1} + g_{1,0}) / 2$$

See also the array pictures below.

		j →			
		0	1	2	3
i ↓	0	pos	neg	neg	neg
	1	neg	pos	neg	neg
	2	neg	neg	pos	neg
	3	neg	neg	neg	pos

eliminated backcontact values			
g <sub>0,0</sub>	g <sub>0,1</sub>	g <sub>0,2</sub>	g <sub>0,3</sub>
	g <sub>1,1</sub>	g <sub>1,2</sub>	g <sub>1,3</sub>
		g <sub>2,2</sub>	g <sub>2,3</sub>
			g <sub>3,3</sub>

### NOTE:

The calculated substrate capacitance values are divided by 2\*pi.

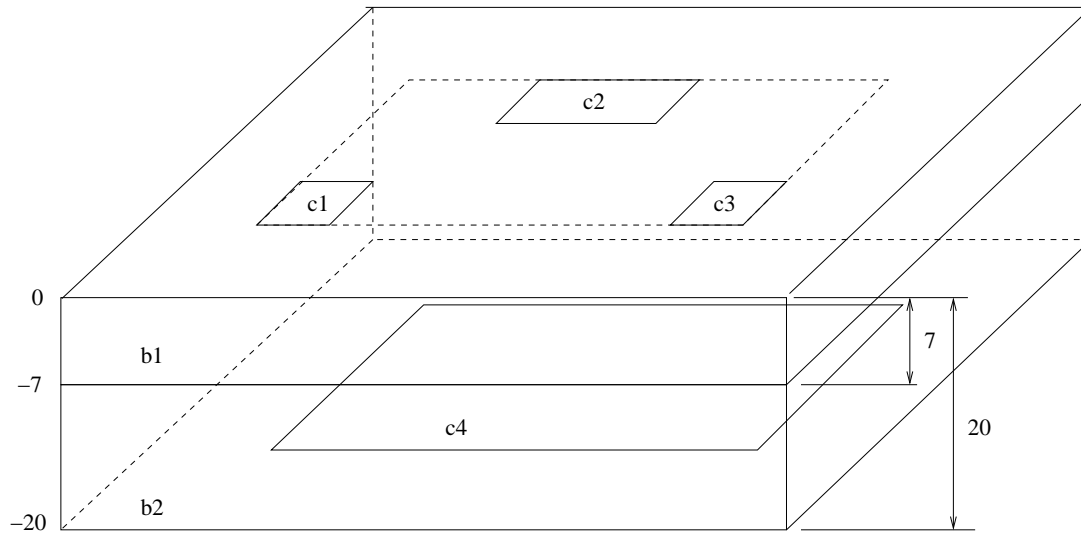
The diagonal values must also be corrected, before writing to the "subres" file.

Thus:

$$\begin{aligned} g_{0,0} &= (g_{0,1} + g_{0,2} + g_{0,3}) \\ g_{1,1} &= (g_{0,1} + g_{1,2} + g_{1,3}) \\ g_{2,2} &= (g_{0,2} + g_{1,2} + g_{2,3}) \\ g_{3,3} &= (g_{0,3} + g_{1,3} + g_{2,3}) \end{aligned}$$

## 6. SUBSTRATE GEOMETRY PICTURE

The substrate geometry of cell "sub3term" is specified in a MATLAB format (see the "step1\_geometry.m" file). A picture of this substrate geometry is given below.



### NOTE:

The substrate contact "c1" starts at position  $x=0$  and  $y=0$ .

The substrate contact bounding box has an overlap of 20 microns.

Contact "c4" is the additional backcontact.

## 7. THE COILGEN EXAMPLE

Depending on substrate parameters "sep\_sub\_term" and "sub\_term\_distr\_mask" the files "cont\_bln" and "subres" are different. The *makefem* program must change the number of tiles found in the "cont\_bln" file into the correct number of substrate contacts for the "subres" file.

The following table gives some statistics:

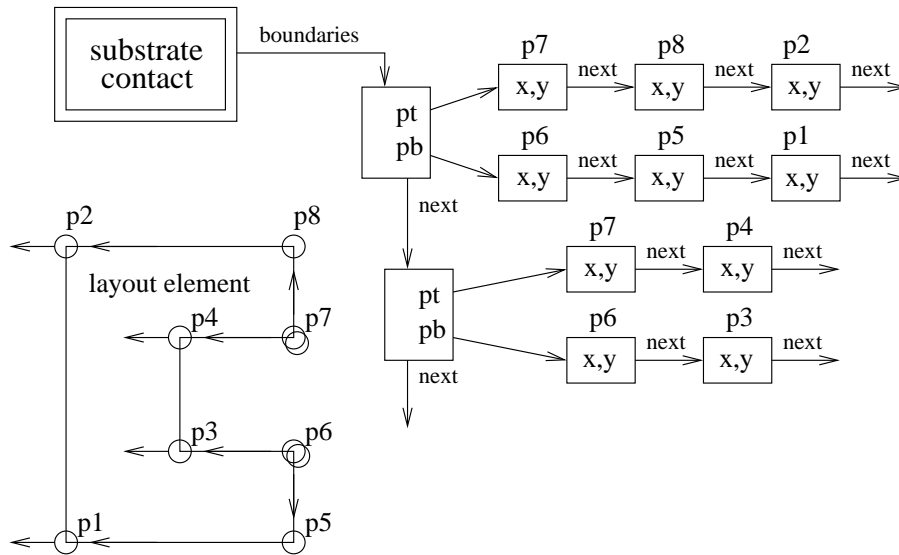
mode	number of tiles	number of contacts
sep_sub_term=off, not distr.	34	1
sep_sub_term=on, not distr.	36	4
sep_sub_term=on, distr_m5/m6	126	126

See appendix D for one result of the "coilgen" example (sep\_sub\_term=off, not distr.).

## 8. MAKEFEM CONTACT BOUNDARY IMPROVEMENT

As explained before (see appendix D), there can be left extra points on the substrate contact boundaries after internal border deletion. Thus, I implemented another method for the substrate border construction.

Out of the substrate tiles a list of substrate border points is extracted. This list is stored by each substrate contact. After the substrate contacts are ready, the border points are outputted as big "poly2" polygons. The "geomdel" function is not more needed, because internal borders don't need to be deleted. If there are holes in a substrate contact, then there are extra internal polygons, which must be subtracted from the external border polygon. The following figure explains the data structure added to a substrate contact.



As seen in the example, parts of the external border polygon may be stored in separate boundary lists. And must be put together, when finished.

Appendix E gives a dump of the data structure for the boundary of cell "coilgen".

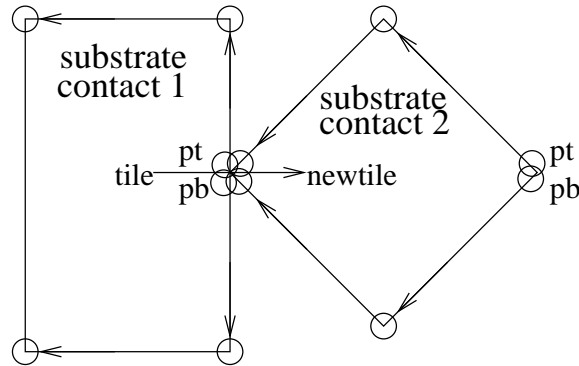
Appendix F gives the results for the improved method for cell "coilgen".

I added new code to *makefem* source files "enumtile.c" and "input.c". I added three new functions, (1) *init\_boundary*, (2) *update\_boundary* and (3) *merge\_boundaries*. The functions are called by *enumPair*. Function *init\_boundary* is called for each new boundary. This init can only happen for a vertical edge. Note that two points are also created for a vertical edge with a edge length of zero. Function *update\_boundary* can also call *init\_boundary*, when update fails to find the edge connection. In that case, it must be the start of an internal boundary. Function *merge\_boundaries* is called, when two substrate contacts are joined together.



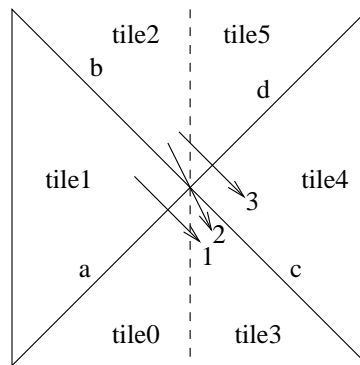
### 8.1 Special Contact Boundary Cases

When two substrate contacts touch each other, but are not joined together, then they can add an extra point on the boundary of each other. Also two substrate contacts are never put together, when they touch each other with a single point. See as an example the figure below:



In this special case ends the boundary of substrate contact 1 on a vertical edge. Thus, the boundary of substrate contact 1 is constructed with 5 points. This is not repaired on output. It is not important, because the point of substrate contact 2 is also touching the contour of substrate contact 1.

The boundary of substrate contact 2 starts with a vertical edge, which edge length is zero. In this special case the `init_boundary` must be done (see `enumPair`), when on both sides of the zero vertical edge is a substrate contact tile. However, this does not happen (as expected) in the case below:



The `enumPair`'s 1 to 3 are all done for a zero length vertical edge. The edges 'a' and 'b' are done first by a call to `tileDeleteEdge`. The last one gives `enumPair(tile1,tile3)`. New edge 'c' is not bundled with edge 'b', because edge 'b' `xr` is equal to thisX. The `tileInsertEdge` of edge 'c' gives `enumPair(tile2,tile3)` and the `tileInsertEdge` of edge 'd' gives `enumPair(tile2,tile4)`.

## 9. APPENDICES

### APPENDIX A -- List of files

#### femtech.in

```
=====
#contacts
contacts 3
backcontact 1
#subdomain
layers 2
epsilon 11.7 11.7
sigma 6.7 2000
#frequence
frequence 1
omega 1e6
```

#### step1\_geometry.m

```
=====
% New geometry 1
fem.sdim={'x','y','z'};

% Geometry of substrate contacts (in meters)
wp=[0 1 0;0 0 1;0 0 0];
s1=rect2(0,1e-6,0,1e-6);
s2=rect2(15e-7,35e-7,3e-6,4e-6);
s3=rect2(4e-6,5e-6,0,1e-6);
c1=embed(s1,wp);
c2=embed(s2,wp);
c3=embed(s3,wp);

% Geometry of substrate blocks (in meters)
% cell bbox: 0 5e-6 0 4e-6
% subc bbox: 0 5e-6 0 4e-6
xl=0;
xr=5e-6;
yb=0;
yt=4e-6;
x=2e-5; % delta x overlap
y=2e-5; % delta y overlap
z=2e-5; % delta z to z-bottom

% bottom plane also has to be defined as separate contact
c4=embed(rect2(xl-x,xr+x,yb-y,yt+y),[0 1 0;0 0 1;-z -z -z]);

b1=block3(xr-xl+2*x,yt-yb+2*y,7e-6,'corner',[xl-x yb-y -7e-6]);
b2=block3(xr-xl+2*x,yt-yb+2*y,z-7e-6,'corner',[xl-x yb-y -z]);

clear s f c p
s.objs={b1,b2};
s.name={'B1','B2'};
f.objs={c1,c2,c3,c4};
f.name={'C1','C2','C3','C4'};
c.objs={};
c.name={};
p.objs={};
p.name={};
drawstruct=struct('s',s,'f',f,'c',c,'p',p);
fem.draw=drawstruct;

fem.geom=geomcsg(fem);
```

## femsub.out

```
=====
frequence 1.000000e+08
real part
 2.934393e-05 -3.443189e-06 -2.061320e-06 -2.383258e-05
-3.443059e-06 3.969221e-05 -3.421877e-06 -3.280838e-05
-2.061767e-06 -3.422627e-06 2.954846e-05 -2.415300e-05
-2.383914e-05 -3.282668e-05 -2.406534e-05 8.072944e-05
image part
 3.026643e-08 -3.711542e-09 -2.237690e-09 -2.253931e-08
-3.677760e-09 4.089951e-08 -3.661541e-09 -2.809899e-08
-2.241087e-09 -3.693202e-09 3.047047e-08 -2.126881e-08
-2.427372e-08 -3.341893e-08 -2.450418e-08 8.219131e-08
```

## subres

```
=====
c 1 1 xl 0 yb 0 g 4.125942e-05 c 6.739797e-09
nr_neigh 2
c 2 3 xl 60 yb 120 g 5.609761e-05 c 9.036076e-09
nc 1 g 8.286918e-06 c 1.307583e-09
nr_neigh 2
c 3 7 xl 160 yb 0 g 4.160057e-05 c 6.729553e-09
nc 1 g 5.620003e-06 c 8.918057e-10
nc 2 g 8.321587e-06 c 1.288846e-09
nr_neigh 2
```

## cont\_bln

```
=====
% dbcat -vs cont_bln sub3term
=> layout/sub3term/cont_bln
  xl:   xr:   yb:   yt:   ct:
    0    40    0    0    0 (BN 0)
    0    40    40    40 1024 (EN 0)
   60   140   120   120    0 (BN 0)
   60   140   160   160 1024 (EN 0)
  160   200    0    0    0 (BN 0)
  160   200   40    40 1024 (EN 0)
```

## runfemlab

```
=====
#
matlab -nodisplay -nojvm > /dev/null
```

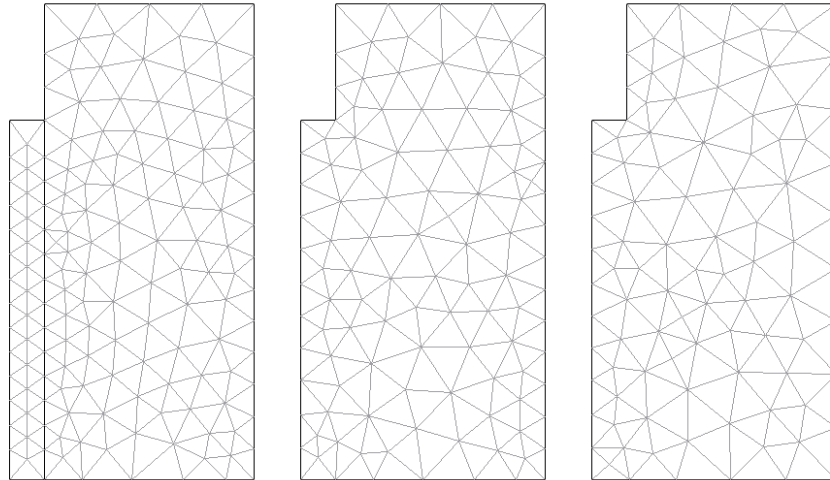
## startup.m

```
=====
main;
quit;
```

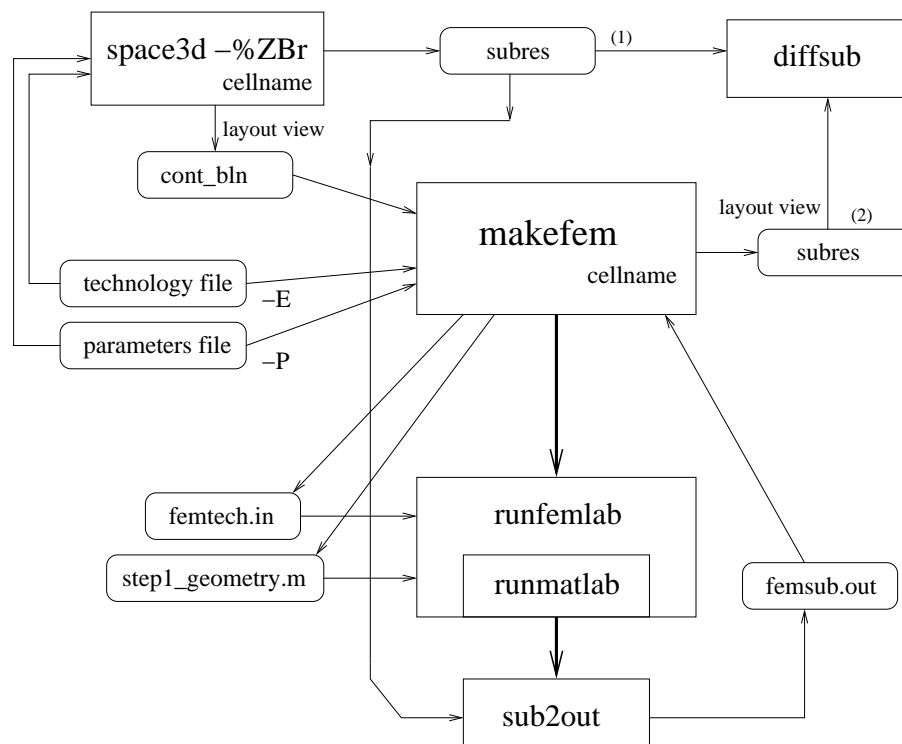
## main.m

```
=====
# FEMLAB algorithm implemented by W.Qiang
```

## APPENDIX B -- Three different mesh results



## APPENDIX C -- Testing makefem



In place of using *matlab*, we can generate *matlab* output with the *sub2out* program to test the working of the *makefem* program. Result subres (1) must be equal to subres (2).

**APPENDIX D -- Example of coilgen**

```

% New geometry 1
fem.sdim={'x','y','z'};

% Geometry of substrate contacts (in meters)
wp=[0 1 0;0 0 1;0 0 0];
s1=rect2(-69e-6,-54e-6,-17e-6,-1e-5);
s2=rect2(-69e-6,-54e-6,1e-5,17e-6);
s3=poly2([-54e-6 -54e-6 -48e-6 -48e-6],[-2237e-8 -1e-5 -1e-5 -2837e-8]);
s4=poly2([-54e-6 -54e-6 -48e-6 -48e-6],[1e-5 2237e-8 2837e-8 1e-5]);
s7=poly2([-45e-6 -45e-6 -4e-5 -4e-5],[-1864e-8 1864e-8 2364e-8 -2364e-8]);
s5=poly2([-48e-6 -48e-6 -2237e-8 -2237e-8],[-2837e-8 -1988e-8 -4551e-8 -54e-6]);
s6=poly2([-48e-6 -48e-6 -2237e-8 -2237e-8],[1988e-8 2837e-8 54e-6 4551e-8]);
s10=poly2([-2237e-8 -2237e-8 -1988e-8 -1988e-8],[-54e-6 -4551e-8 -48e-6 -54e-6]);
s11=poly2([-2237e-8 -2237e-8 -1988e-8 -1988e-8],[4551e-8 54e-6 54e-6 48e-6]);
s8=poly2([-4e-5 -4e-5 -1864e-8 -1864e-8],[-2364e-8 -1657e-8 -3793e-8 -45e-6]);
s9=poly2([-4e-5 -4e-5 -1864e-8 -1864e-8],[1657e-8 2364e-8 45e-6 3793e-8]);
s14=poly2([-1864e-8 -1864e-8 -1657e-8 -1657e-8],[-45e-6 -3793e-8 -4e-5 -45e-6]);
s15=poly2([-1864e-8 -1864e-8 -1657e-8 -1657e-8],[3793e-8 45e-6 45e-6 4e-5]);
s16=rect2(-1657e-8,1657e-8,-45e-6,-4e-5);
s17=rect2(-1657e-8,1657e-8,4e-5,45e-6);
s18=poly2([1657e-8 1657e-8 1864e-8 1864e-8],[-45e-6 -4e-5 -3793e-8 -45e-6]);
s19=poly2([1657e-8 1657e-8 1864e-8 1864e-8],[4e-5 45e-6 45e-6 3793e-8]);
s12=rect2(-1988e-8,1988e-8,-54e-6,-48e-6);
s13=rect2(-1988e-8,1988e-8,48e-6,54e-6);
s22=poly2([1988e-8 1988e-8 2237e-8 2237e-8],[-54e-6 -48e-6 -4551e-8 -54e-6]);
s23=poly2([1988e-8 1988e-8 2237e-8 2237e-8],[48e-6 54e-6 54e-6 4551e-8]);
s20=poly2([1864e-8 1864e-8 4e-5 4e-5],[-45e-6 -3793e-8 -1657e-8 -2364e-8]);
s21=poly2([1864e-8 1864e-8 4e-5 4e-5],[3793e-8 45e-6 2364e-8 1657e-8]);
s26=poly2([4e-5 4e-5 4261e-8 4261e-8],[-2364e-8 -226e-8 35e-8 -2103e-8]);
s27=poly2([4e-5 4e-5 4261e-8 4261e-8],[296e-8 2364e-8 2103e-8 35e-8]);
s28=poly2([4261e-8 4261e-8 45e-6 45e-6],[-2103e-8 2103e-8 1864e-8 -1864e-8]);
s29=poly2([45e-6 45e-6 4615e-8 4615e-8],[-574e-8 504e-8 389e-8 -459e-8]);
s30=poly2([4615e-8 4615e-8 4685e-8 4685e-8],[-459e-8 389e-8 459e-8 -389e-8]);
s24=poly2([2237e-8 2237e-8 48e-6 48e-6],[-54e-6 -4551e-8 -1988e-8 -2837e-8]);
s31=poly2([4685e-8 4685e-8 48e-6 48e-6],[-389e-8 459e-8 574e-8 -504e-8]);
s25=poly2([2237e-8 2237e-8 48e-6 48e-6],[4551e-8 54e-6 2837e-8 1988e-8]);
s32=poly2([48e-6 48e-6 5039e-8 5039e-8],[-2837e-8 2837e-8 2598e-8 -2598e-8]);
s33=poly2([5039e-8 5039e-8 54e-6 54e-6],[-2598e-8 -35e-8 -396e-8 -2237e-8]);
s34=poly2([5039e-8 5039e-8 54e-6 54e-6],[-35e-8 2598e-8 2237e-8 326e-8]);
c1=embed(geomdel(s34+s33+s25+s23+s13+s11+s6+s4+s2+s31+s30+s29+s28+s27+s26+s21+ ...
    s20+s19+s18+s17+s16+s15+s14+s9+s8+s7+s32+s24+s22+s12+s10+s5+s3+s1),wp);

% Geometry of substrate blocks (in meters)
xl=-69e-6; xr=54e-6; yb=-54e-6; yt=54e-6;
x=2e-5; % delta x overlap
y=2e-5; % delta y overlap
z=2e-5; % delta z to z-bottom

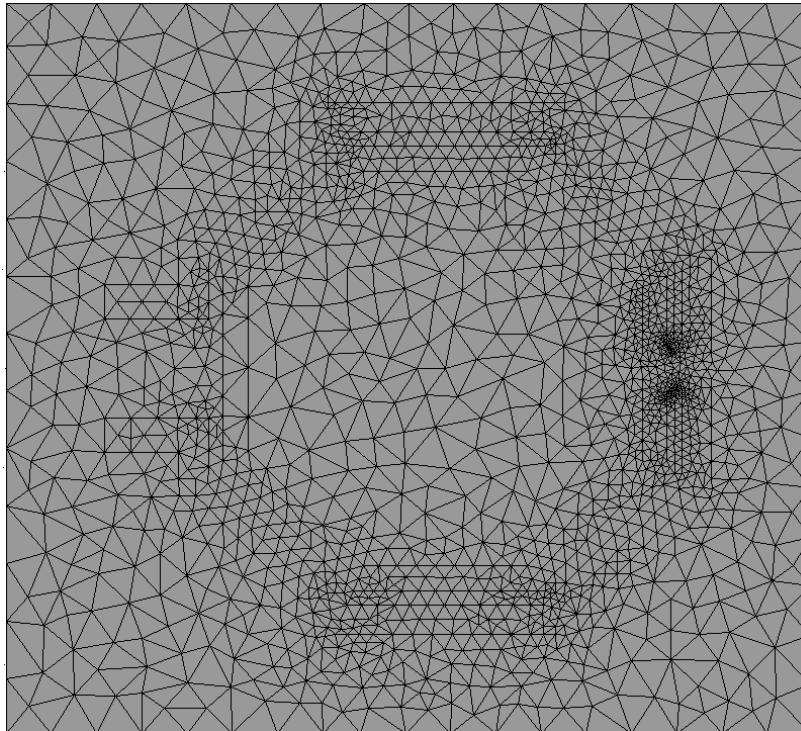
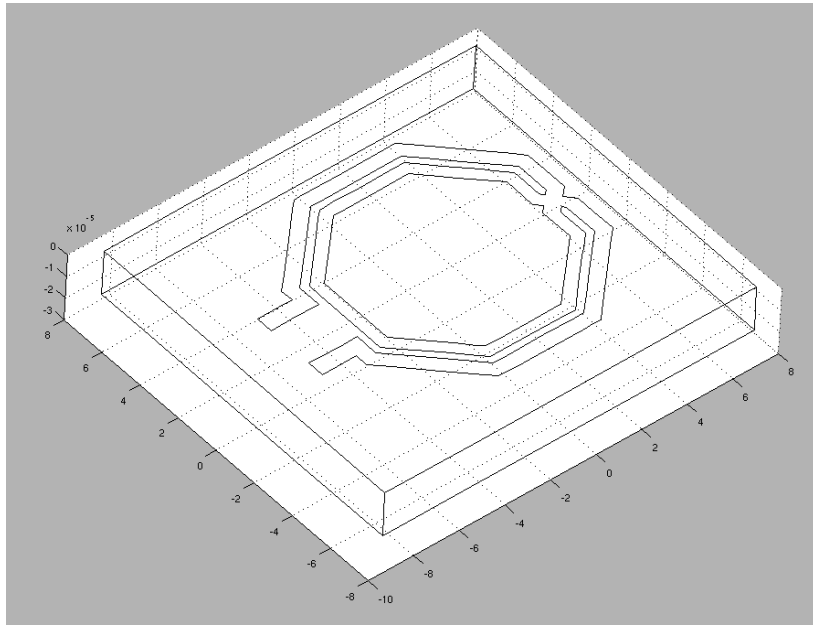
% bottom plane also has to be defined as separate contact
c2=embed(rect2(xl-x,xr+x,yb-y,yt+y),[0 1 0;0 0 1;-z -z -z]);

b1=block3(xr-xl+2*x,yt-yb+2*y,z,'corner',[xl-x yb-y -z]);

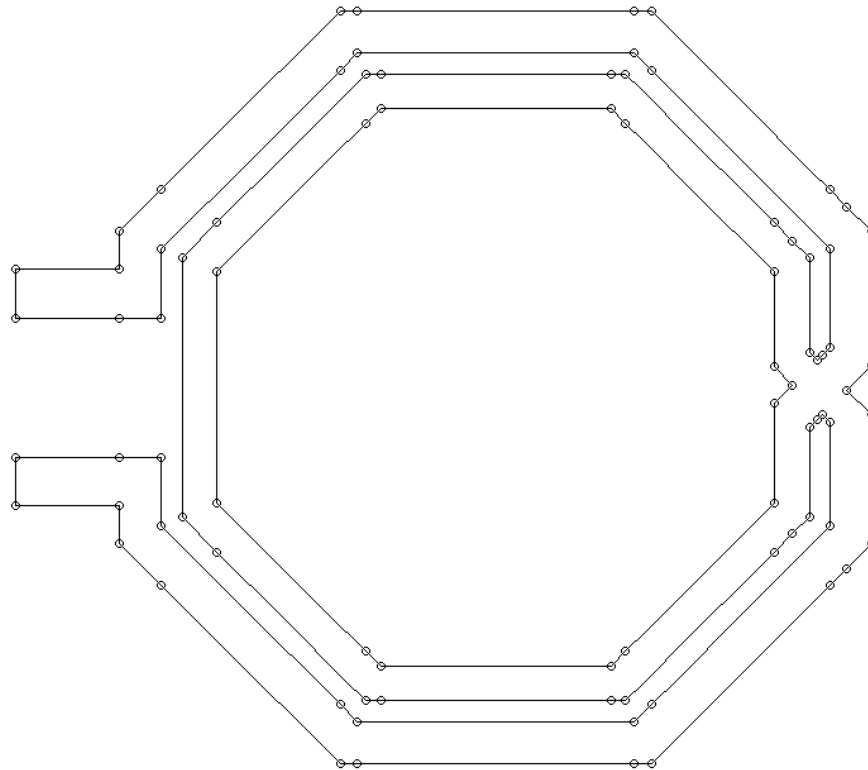
clear s f c p
s.objs={b1}; s.name={'B1'};
f.objs={c1,c2}; f.name={'C1','C2'};
c.objs={}; c.name={};
p.objs={}; p.name={};
drawstruct=struct('s','f','c','p');
fem.draw=drawstruct;

```

```
% This is for visual feedback:  
fem.geom=geomcsg(fem);
```



The following picture shows the border points for cell "coilgen".  
As you can see, there are a number of extra points.



**APPENDIX E -- Dump of boundary points for coilgen**

```

pt: 54000 3260      <----- boundary 1 : top
pt: 54000 22370
pt: 22370 54000
pt: -22370 54000
pt: -54000 22370
pt: -54000 17000
pt: -69000 17000
pt: -69000 10000
pt: -48000 10000
pt: -48000 19880
pt: -19880 48000
pt: 19880 48000
pt: 48000 19880
pt: 48000 5740
pt: 46150 3890
pt: 45000 5040
pt: 45000 18640
pt: 18640 45000
pt: -18640 45000
pt: -45000 18640
pt: -45000 -18640
pt: -18640 -45000
pt: 18640 -45000
pt: 45000 -18640
pt: 45000 -5740
pt: 46850 -3890
pt: 48000 -5040
pt: 48000 -19880
pt: 19880 -48000
pt: -19880 -48000
pt: -48000 -19880
pt: -48000 -10000
pt: -69000 -10000
pb: 54000 -3960     <----- boundary 1 : bottom
pb: 54000 -22370
pb: 22370 -54000
pb: -22370 -54000
pb: -54000 -22370
pb: -54000 -17000
pb: -69000 -17000
-----
pt: 42610 350       <----- boundary 2 : top
pt: 40000 2960
pt: 40000 16570
pt: 16570 40000
pt: -16570 40000
pt: -40000 16570
pb: 42610 350       <----- boundary 2 : bottom
pb: 40000 -2260
pb: 40000 -16570
pb: 16570 -40000
pb: -16570 -40000
pb: -40000 -16570
-----
pt: 54000 3260      <----- boundary 3 : top
pt: 50390 -350
pb: 54000 -3960     <----- boundary 3 : bottom
pb: 50390 -350

```



**APPENDIX F -- Results for the improved coilgen**

```

% New geometry 1
fem.sdim={'x','y','z'};

% Geometry of substrate contacts (in meters)
wp=[0 1 0;0 0 1;0 0 0];
s1=poly2([-69e-6 -48e-6 -48e-6 -1988e-8 1988e-8 48e-6 48e-6 4685e-8 45e-6 45e-6 ...
1864e-8 -1864e-8 -45e-6 -45e-6 -1864e-8 1864e-8 45e-6 45e-6 4615e-8 48e-6 ...
48e-6 1988e-8 -1988e-8 -48e-6 -48e-6 -69e-6 -69e-6 -54e-6 -54e-6 -2237e-8 ...
2237e-8 54e-6 54e-6 5039e-8 54e-6 54e-6 2237e-8 -2237e-8 -54e-6 -54e-6 -69e-6], ...
[-1e-5 -1e-5 -1988e-8 -48e-6 -48e-6 -1988e-8 -504e-8 -389e-8 -574e-8 -1864e-8 ...
-45e-6 -45e-6 -1864e-8 1864e-8 45e-6 45e-6 1864e-8 504e-8 389e-8 574e-8 ...
1988e-8 48e-6 48e-6 1988e-8 1e-5 1e-5 17e-6 17e-6 2237e-8 54e-6 ...
54e-6 2237e-8 326e-8 -35e-8 -396e-8 -2237e-8 -54e-6 -54e-6 -2237e-8 -17e-6 -17e-6]);
s2=poly2([-4e-5 -1657e-8 1657e-8 4e-5 4e-5 4261e-8 4e-5 4e-5 1657e-8 -1657e-8 -4e-5], ...
[1657e-8 4e-5 4e-5 1657e-8 296e-8 35e-8 -226e-8 -1657e-8 -4e-5 -4e-5 -1657e-8]);
c1=embed(s1-s2,wp);

% Geometry of substrate blocks (in meters)
% cell bbox: -69e-6 54e-6 -54e-6 54e-6
% subc bbox: -69e-6 54e-6 -54e-6 54e-6
xl=-69e-6;
xr=54e-6;
yb=-54e-6;
yt=54e-6;
x=2e-5; % delta x overlap
y=2e-5; % delta y overlap
z=2e-5; % delta z to z-bottom

% bottom plane also has to be defined as separate contact
c2=embed(rect2(xl-x,xr+x,yb-y,yt+y),[0 1 0;0 0 1;-z -z -z]);

b1=block3(xr-xl+2*x,yt-yb+2*y,z,'corner',[xl-x yb-y -z]);

clear s f c p
s.objs={b1};
s.name={'B1'};
f.objs={c1,c2};
f.name={'C1','C2'};
c.objs={};
c.name={};
p.objs={};
p.name={};
drawstruct=struct('s',s,'f',f,'c',c,'p',p);
fem.draw=drawstruct;

% This is for visual feedback:
fem.geom=geomcsg(fem);

```

**Mesh statistics of FEMLAB:**

method	nodes	elements
old	3921	16598
new	2220	9560

