2.12 Implementation of a mesh generator - DistMesh

1. DistMesh is originally a MATLAB code developed by Per-Olof Persson and Gilbert Strang for generation of *triangular* and *tetrahedral* meshes in 2D and 3D problems respectively.
2. The code is relatively convenient, and the user is able to define a variety of geometric shapes, and desired mesh densities.
3. Basic usage of the DistMesh code:

```
[p,t] = distmesh_2d(fd,fh h,box,iteration_max,fixed);
```

where
(1) `fd` is the name of a signed distance function defining the region to be meshed,
(2) `fh` is the name of a mesh density function,
(3) `h` is the nominal mesh spacing,
(4) `box` is the matrix defining a box that contains the region,
(5) `iteration_max` is the maximum number of iterations,
(6) `fixed` is a list of points which must be included in the mesh, or `[ ]` if no fixed points are given.

4. Reference:
Per-Olof Persson & Gilbert Strang, "A Simple Mesh Generator in MATLAB," *SIAM Review*, Volume 46, Number 2, June 2004, pages 329-345.

5. An examples of generation of *triangular* meshes in 2D problems
(1) A circle region with nominal mesh spacings `h` = 0.4, 0.3, 0.2:

```
iteration_max = 200;
h = 0.4;
filename = 'p01_h04';
[p,t] = p01_demo(iteration_max,h);
plotmesh(p,t,filename);
```

where the essential functions are:
① p01_demo

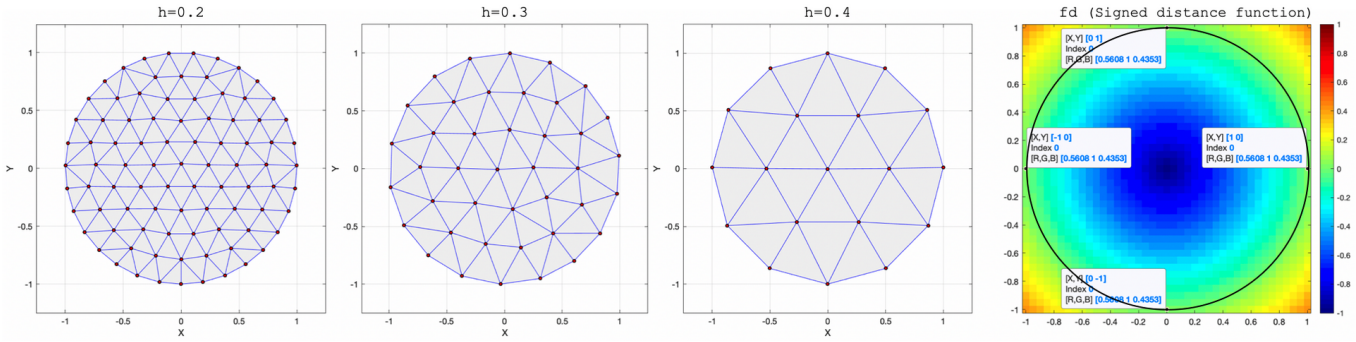
```
function [p,t] = p01_demo(iteration_max,h)
    rng('default');
    fd = @p01_fd;
    fh = @p01_fh;
    box = [-1.0,-1.0; 1.0,1.0];
    fixed = [ ];
    [p,t] = distmesh_2d(fd,fh,h,box,iteration_max,fixed);
    p = p.';
    t = t.';
end
```

② p01_fd

```
function d = p01_fd(p)
    d = sqrt(sum(p.^2, 2))-1;
end
```

③ p01_fh

```
function h = p01_fh(p)
    np = size(p,1);
    h = ones(np,1);
end
```



(2) A square with a hole with finer mesh density near the hole

```
iteration_max = 300;
h = 0.06;
fh = @(p) min(4*sqrt(sum(p.^2,2))-1,2);
filename = 'p03_2';
[p,t] = p03_demo(iteration_max,h,fh);
plotmesh(p,t,filename);
```

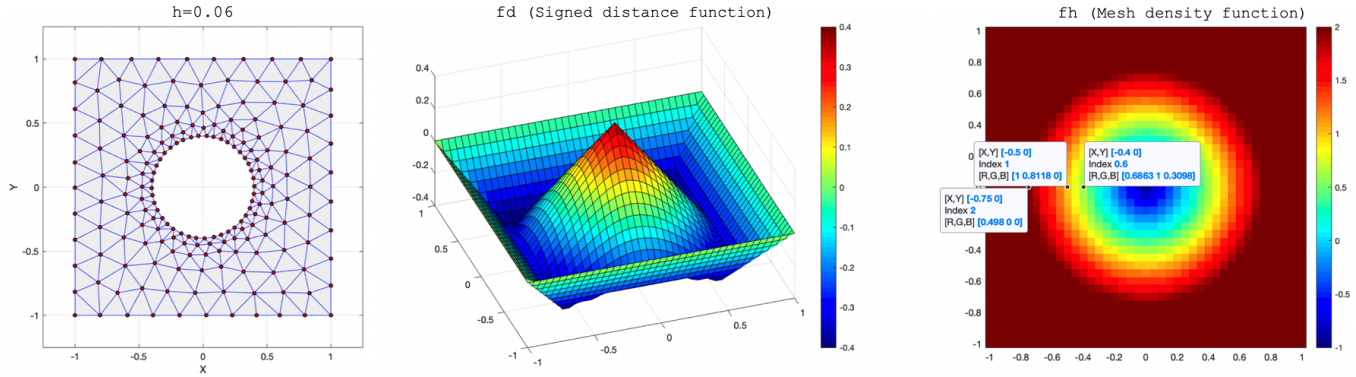
where the essential functions are:

① p03_demo

```
function [p,t] = p03_demo(iteration_max,h,fh)
    rng('default');
    fd = @p03_fd;
    box = [-1.0,-1.0; 1.0,1.0];
    fixed = [-1.0,-1.0; -1.0,1.0; 1.0,-1.0; 1.0,1.0];
    [p,t] = distmesh_2d(fd,fh,h,box,iteration_max,fixed);
    p = p.';
    t = t.';
end
```

② p03_fd

```
function d = p03_fd(p)
    d1 = drectangle(p,-1.0,1.0,-1.0,1.0);
    d2 = dcircle(p,0.0,0.0,0.4);
    d = ddiff(d1,d2);
end
```



6. An examples of generation of *tetrahedral* meshes in 3D problems

A cube with nominal mesh spacings h = 0.2 with its eight specified corner nodes:

```
iteration_max = 200;
  filename = 'q03';
  [p,t] = q03_demo(iteration_max);
  plotmesh(p,t,filename);
```

where the essential functions are:

① q03_demo

```
function [p,t] = q03_demo(iteration_max)
    rng('default');
```

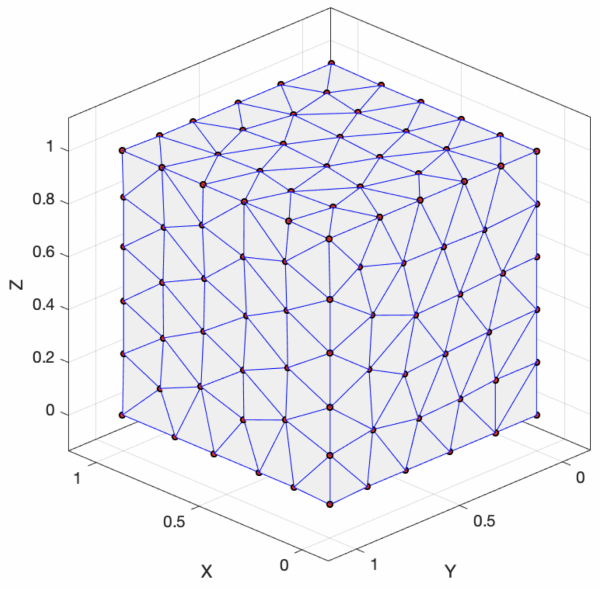
```
    fd = @q03_fd;
    fh = @q03_fh;
    h = 0.2;
    box = [0, 0, 0; 1, 1, 1];
    fixed = [0, 0, 0; 0, 0, 1; 0, 1, 0; 0, 1, 1; 1, 0, 0; 1, 0, 1; 1, 1, 0; 1, 1, 1];
    [p,t] = distmesh_3d(fd,fh,h,box,iteration_max,fixed);
    p = p.';
    t = t.';
end
```

② q03_fd

```
function d = q03_fd(p)
    d = -min(...
      min(...
      min(...
      min(...
      min(p(:,3)-0.0,1.0-p(:,3)),...
      p(:,2)-0.0),...
      1.0-p(:,2)),...
      p(:,1)-0.0),...
      1.0-p(:,1));
end
```

③ q03_fh

```
function h = q03_fh(p)
    np = size(p,1);
    h = ones(np,1);
end
```

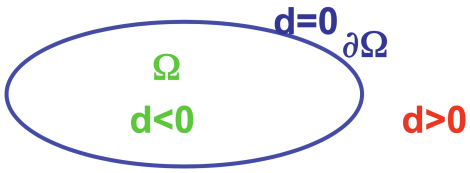


7. Algorithm of DistMesh for MATLAB

(1) Signed distance function $d(\boldsymbol{x})$

Signed distance function is the metric using an appropriate norm function, e.g., the usual Euclidian metric.

$$(2.77)\ d(\boldsymbol{x}) = \begin{cases} < 0, & \boldsymbol{x} \in \Omega \\ 0, & \boldsymbol{x} \in \partial\Omega \\ > 0, & \boldsymbol{x} \notin \Omega \end{cases}$$



(2) Basic set operators for signed distance functions

① Union: $A \cup B$

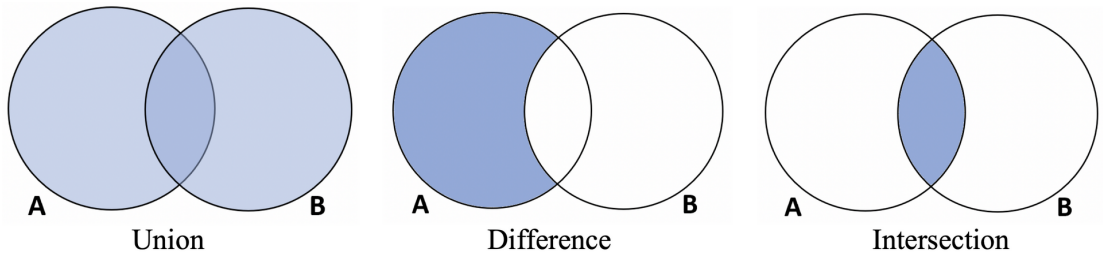
(2.78) $d_{A\cup B} = \min(d_A, d_B)$

② Difference: $A \setminus B$

(2.79) $d_{A\setminus B} = \max(d_A, -d_B)$

③ Intersection: $A \cap B$

(2.80) $d_{A\cap B} = \max(d_A, d_B)$



8. MATLAB code demonstration in 2D problems: Generate unstructured triangular meshes

(1) A circle, with spacings h = 0.4, 0.3, 0.2

```
filename = 'p01';
fd = @p01_fd;
fh = @p01_fh;
h = 0.2;
meshbox = [-1.0,-1.0; 1.0,1.0];
iteration_max = 200;
fixed = [];
```

[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');p = p';

p = p';

t = t';

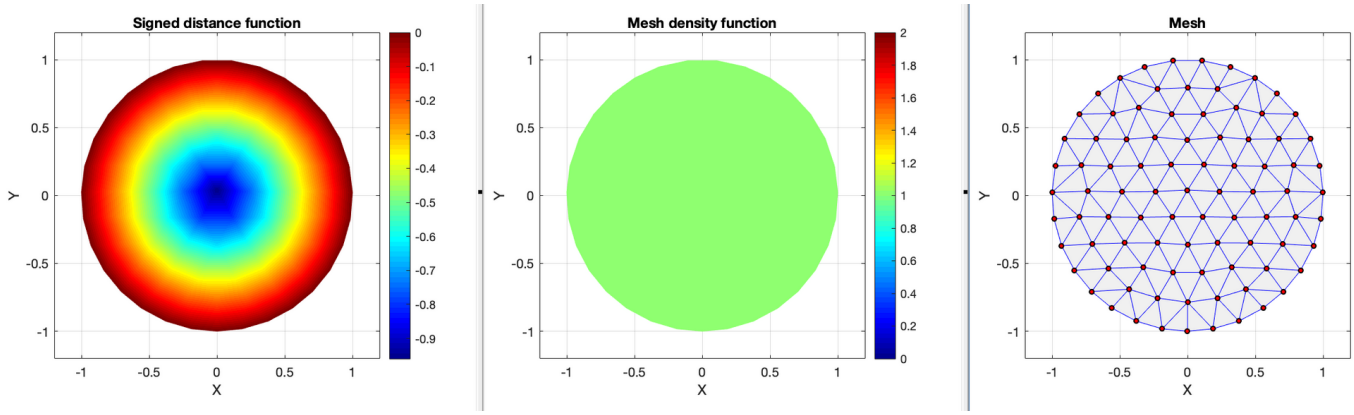
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = p01_fd(p)
    d = sqrt(sum(p.^2,2))-1.0;
end
```

② Mesh density function

```
function h = p01_fh(p)
    h = ones(size(p,1),1);
end
```



(2) A unit circle with a hole

```
filename = 'p02';
fd = @p02_fd;
fh = @p02_fh;
h = 0.15;
meshbox = [-1.0,-1.0; 1.0,1.0];
iteration_max = 200;
fixed = [ ];
```

[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

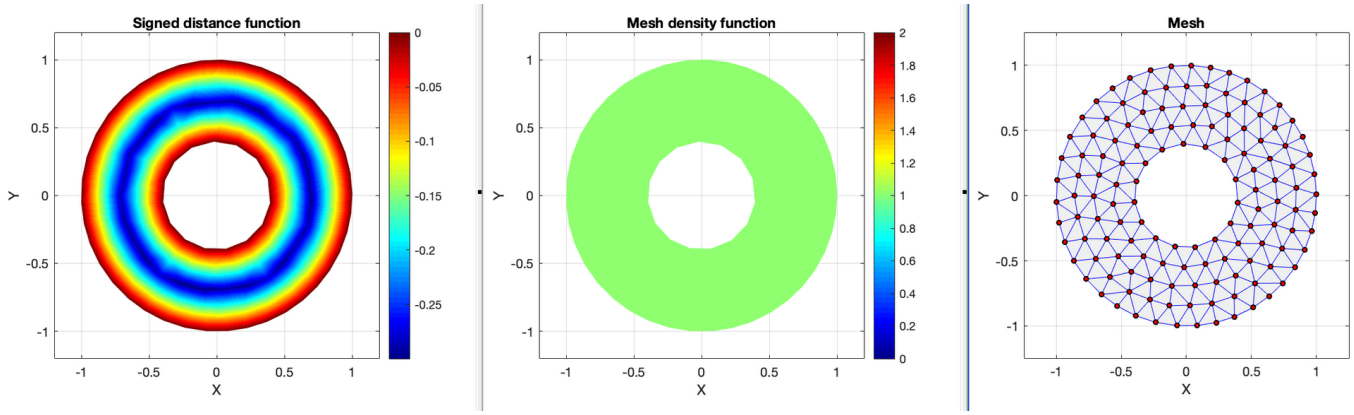
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = p02_fd(p)
    d1 = dcircle(p,0.0,0.0,1.0);
    d2 = dcircle(p,0.0,0.0,0.4);
    d = ddiff(d1,d2);
end
```

② Mesh density function

```
function h = p02_fh(p)
    h = ones(size(p,1),1);
end
```



(3) A square with a hole with uniform density

```
filename = 'p03_1';
fd = @p03_fd;
fh = @p03_fh;
h = 0.15;
meshbox = [-1.0,-1.0; 1.0,1.0];
```

```
iteration_max = 200;
fixed = [-1.0,-1.0; -1.0,1.0; 1.0,-1.0; 1.0,1.0];
```

[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

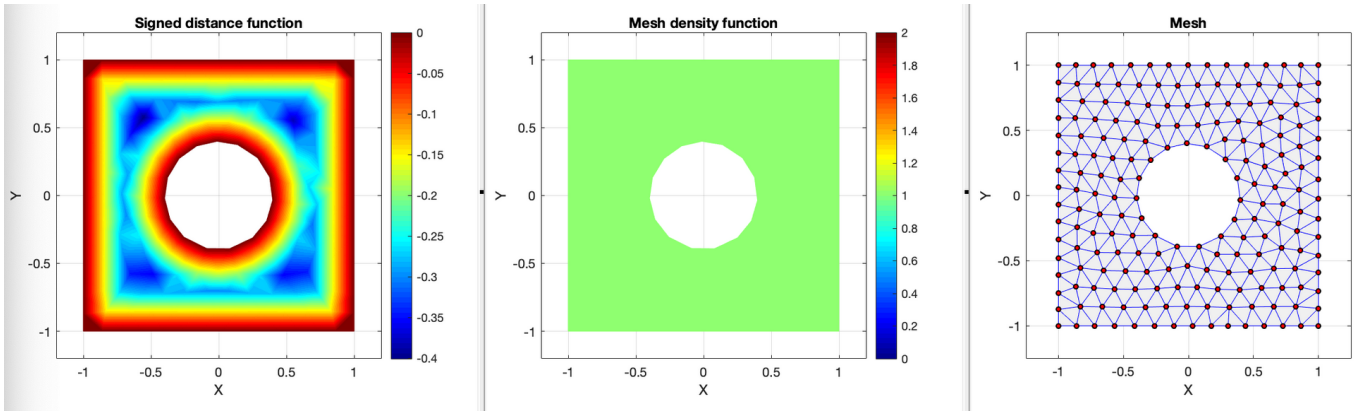
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = p03_fd(p)
    d1 = drectangle(p,-1.0,1.0,-1.0,1.0);
    d2 = dcircle(p,0.0,0.0,0.4);
    d = ddiff(d1,d2);
end
```

② Mesh density function

```
function h = p03_fh(p)
    h = ones(size(p,1),1);
end
```



(4) A square with a hole with finer density near the hole

```
filename = 'p03_2';
fd = @p03_fd;
fh = @(p)min(4*sqrt(sum(p.^2,2))-1,2);
h = 0.06;
meshbox = [-1.0,-1.0; 1.0,1.0];
iteration_max = 300;
fixed = [-1.0,-1.0; -1.0,1.0; 1.0,-1.0; 1.0,1.0];
```

[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

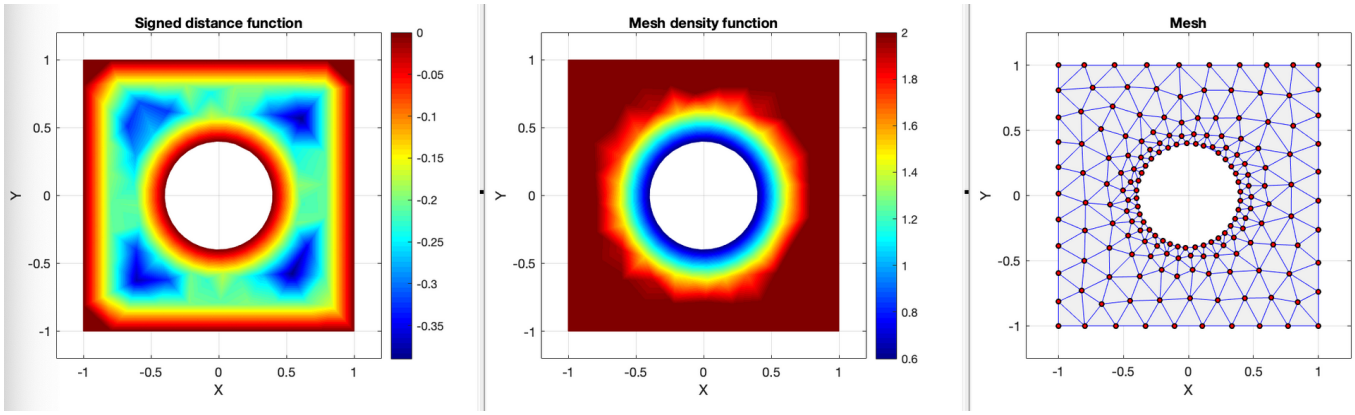
figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

figure('color','w'); plotmesh(p,t,filename); title('Mesh');



(5) A hexagon with a hexagonal hole

```
filename = 'p04';
fd = @p04_fd;
fh = @p04_fh;
h = 0.1;
meshbox = [-1.0,-1.0; 1.0,1.0];
iteration_max = 200;
n = 6;
phi = 2*pi*(0:2:2*(n-1))'/(2*n);
outer = [cos(phi), sin(phi)];
phi = 2*pi*(1:2:2*(n-1)+1)'/(2*n);
inner = 0.5*[cos(phi),sin(phi)];
fixed = [inner; outer];
```

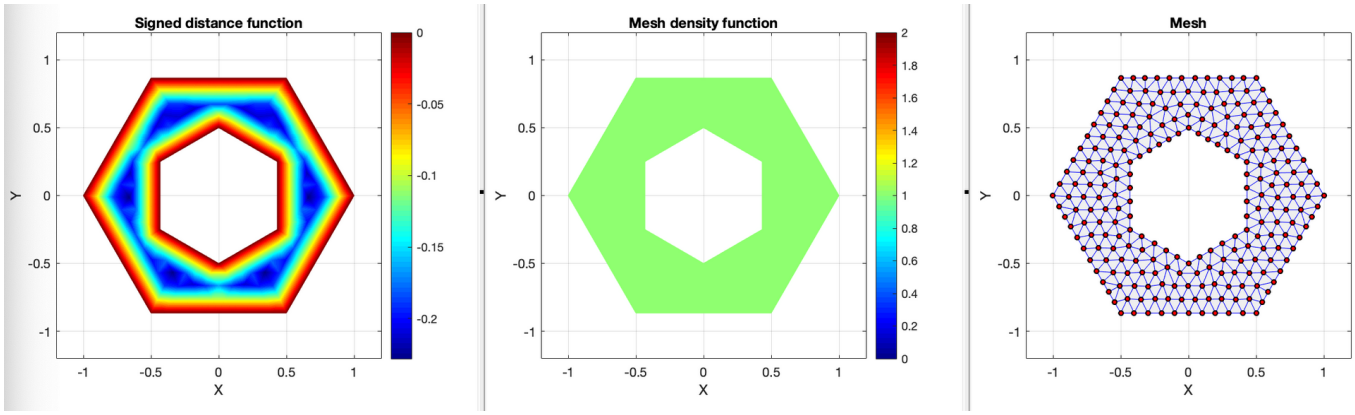
[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);
figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Signed distance function');
figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Mesh density function');
p = p';
t = t';
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = p04_fd(p)
    n = 6;
    phi = 2*pi*(0:2:2*n)'/(2*n);
    outer = [cos(phi), sin(phi)];
    phi = 2*pi*(1:2:2*n+1)'/(2*n);
    inner = 0.5*[cos(phi), sin(phi)];
    d1 = dpoly(p,outer);
    d2 = dpoly(p,inner);
    d = ddiff(d1,d2);
end
```

② Mesh density function

```
function h = p04_fh(p)
    h = ones(size(p,1),1);
end
```



(6) A horn

```
filename = 'p05';
fd = @p05_fd;
fh = @p05_fh;
h = 0.02;
meshbox = [-1.0,0.0; 1.0,1.0];
iteration_max = 200;
fixed = [ -1.0,0.0; -0.95,0.0; 0.15,0.0; 1.0,0.0];
```

[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

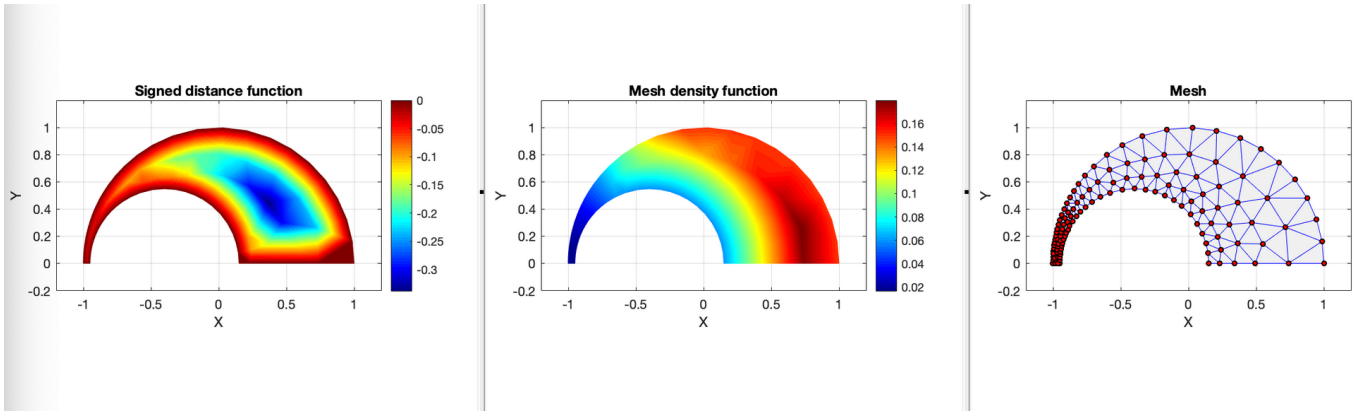
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = p05_fd(p)
    d1 = dcircle(p,0,0,1);
    d2 = dcircle(p,-0.4,0,0.55);
    d = dintersect(-p(:,2),ddiff(d1,d2));
end
```

② Mesh density function

```
function h = p05_fh(p)
    d1 = dcircle(p,0.0,0.0,1.0);
    d2 = dcircle(p,-0.4,0.0,0.55);
    h1 = (0.15-0.2*d1);
    h2 = (0.06+0.2*d2);
    h3 = (d2-d1)/3;
    h = min(min(h1,h2),h3);
end
```



(7) Holey pie slice with uniform density

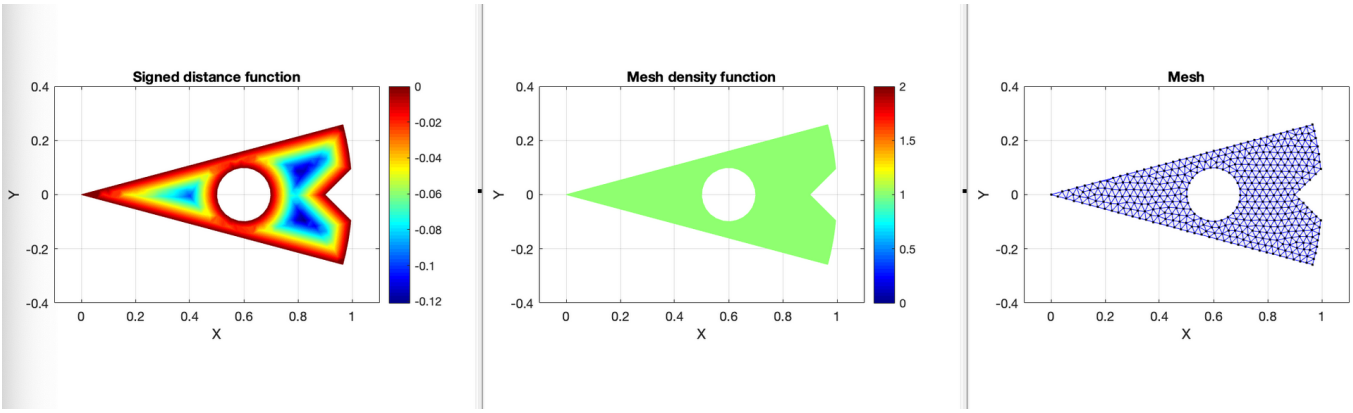
```
filename = 'p08_1';
fd = @p08_fd;
fh = @huniform;
h = 0.025;
meshbox = [0.0,-1.0; 1.0,1.0];
iteration_max = 200;
c = (sqrt(119)-9)/20;
fixed = [0.9+c, c; cos(pi/12), sin(pi/12)];
fixed = [fixed; fixed];
fixed(3:4,2) = -fixed(3:4,2);
fixed = [fixed; 0.0,0.0; 0.9,0.0];
```

```
[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);
figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Signed distance function');
figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Mesh density function');
```

```
p = p';
t = t';
figure('color','w'); plotmesh(p,t,filename); title('Mesh');
```

① Signed distance function

```
function d = p08_fd(p)
    g1 = dcircle(p,0,0,1);
    g2 = drectangle(protate(p,pi/12),-1,1,0,1);
    g3 = drectangle(protate(p,-pi/12),-1,1,-1,0);
    g4 = drectangle(protate(pshift(p,-0.9,0),-pi/4),0,0.2,0,0.2);
    g5 = dcircle(p,0.6,0,0.1);
    d = ddiff(ddiff(ddiff(ddiff(g1,g2),g3),g4),g5);
end
```



(8) Holey pie slice with variable density

```
filename = 'p08_2';
fd = @p08_fd;
fh = @p08_fh;
h = 0.005;
meshbox = [0.0,-1.0; 1.0,1.0];
iteration_max = 200;
c = (sqrt(119)-9)/20;
fixed = [0.9+c, c; cos(pi/12), sin(pi/12)];
fixed = [fixed; fixed];
fixed(3:4,2) = -fixed(3:4,2);
fixed = [fixed; 0.0,0.0; 0.9,0.0];
```

```
[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);
figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');
```

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

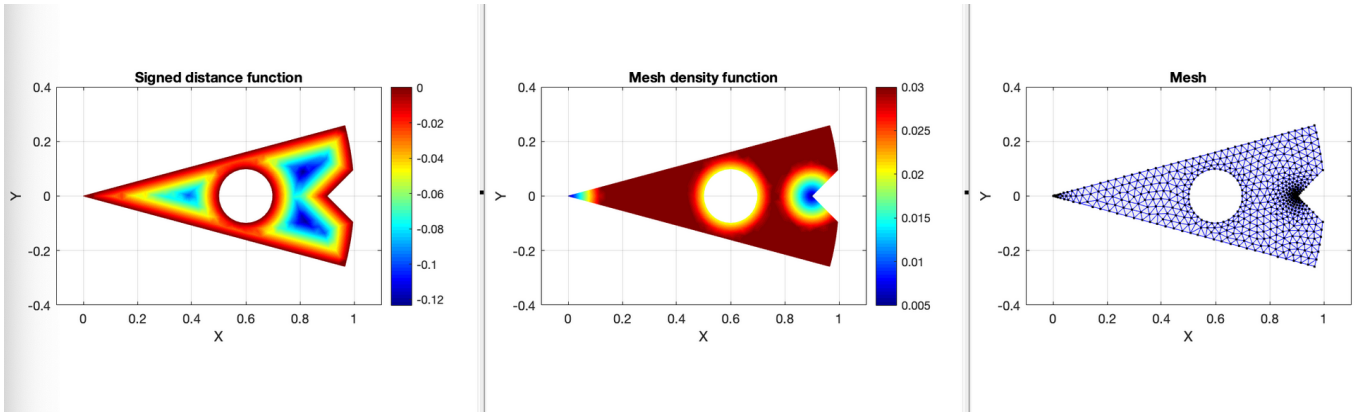
p = p';

t = t';

figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Mesh density function

```
function h = p08_fh(p)
    h1 = 0.005+0.2*sqrt(sum(p.^2,2));
    h2 = 0.02+0.2*(sqrt((p(:,1)-0.6).^2+p(:,2).^2)-0.1);
    h3 = 0.005+0.2*sqrt((p(:,1)-0.9).^2+p(:,2).^2);
    h = min(min(min(h1,h2),h3),0.03);
end
```



(9) The Sandia Fork

```
filename = 'p13';
fd = @p13_fd;
fh = @p13_fh;
h = 0.025;
meshbox = [0.0,0.0; 1.0,1.0];
iteration_max = 200;
fixed = [ ...
0.10000, 0.00000; ...
0.20000, 0.00000; ...
0.80000, 0.00000; ...
0.90000, 0.00000; ...
0.55000, 0.39686; ...
0.55000, 0.90000; ...
0.45000, 0.90000; ...
0.45000, 0.39686 ];
```

[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

figure('color','w'); plotmesh(p,t,filename); title('Mesh');

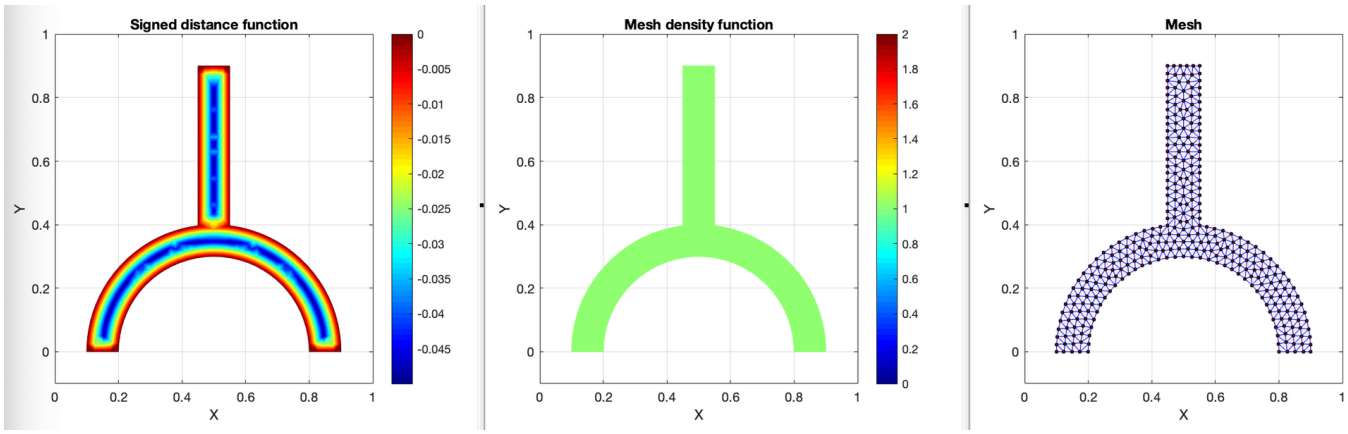
① Signed distance function

```
function d = p13_fd(p)
    g1 = drectangle(p,0.450,0.550,0.390,0.900);
    g2 = dcircle(p,0.500,0.000,0.400);
    g3 = dcircle(p,0.500,0.000,0.300 );
```

```
    d = dunion(g1,dintersect(-p(:,2),ddiff(g2,g3)));
end
```

② Mesh density function

```
function h = p13_fh(p)
    h = ones(size(p,1),1);
end
```



(10) The Sandia Fork

```
filename = 'p14_2';
fd = @p14_fd;
fh = @p14_fh;
h = 10.0;
meshbox = [100.0,145.0; 634.0,799.0];
iteration_max = 50;
outer = [ ...
316.43027, 404.47559; 291.04946, 400.70917; 265.16504, 409.77890; 241.46794, 402.40310; ...
216.55145, 396.52064; 163.28492, 411.37102; 142.81752, 391.16355; 111.95404, 346.70264; ...
100.03538, 325.72710; 103.98723, 302.51587; 128.72978, 285.72802; 147.49111, 266.23345; ...
196.65261, 242.24055; 213.56835, 221.67192; 226.49969, 198.09326; 248.37126, 183.50473; ...
262.21952, 165.39102; 278.42330, 149.91715; 300.71846, 145.82601; 311.12698, 166.71094; ...
326.66315, 184.58335; 359.78574, 225.48049; 357.08892, 252.88958; 358.76685, 285.34403; ...
361.50834, 303.71287; 371.68926, 314.92452; 380.49890, 324.58632; 396.37634, 328.88990; ...
412.59116, 327.25238; 425.48394, 315.28623; 435.84305, 302.44664; 458.34025, 297.55121; ...
479.66439, 288.99238; 493.09812, 270.20636; 518.87309, 264.56427; 547.18014, 268.18846; ...
600.49708, 240.62570; 625.96183, 238.40347; 633.90530, 260.70629; 621.50451, 285.88914; ...
576.87224, 322.14121; 570.51915, 348.85423; 567.16400, 378.24075; 558.00668, 406.86552; ...
565.19008, 435.75599; 567.56437, 465.33407; 550.87626, 490.96358; 532.98174, 515.84491; ...
500.66817, 551.89078; 478.75120, 562.17222; 430.03371, 583.94286; 401.20454, 587.69910; ...
368.32214, 581.10110; 354.26303, 585.86085; 346.75200, 601.10367; 332.85137, 628.74602; ...
308.02188, 645.84180; 295.52344, 647.18525; 286.51519, 651.60328; 285.98846, 662.07339; ...
298.93455, 665.66316; 301.70226, 682.79570; 278.65857, 689.63850; 266.25737, 712.11005; ...
287.28701, 732.77147; 318.19548, 736.85151; 343.83067, 753.60957; 375.53164, 758.35231; ...
405.73444, 768.98687; 406.33873, 785.59001; 378.35436, 789.44240; 350.02151, 795.02238; ...
338.68030, 788.87325; 325.67930, 786.10177; 319.05995, 798.04657; 301.78158, 795.34254; ...
280.69272, 773.86634; 254.55844, 758.02898; 234.07759, 737.42090; 218.38337, 711.41500; ...
220.99086, 682.17833; 224.50640, 651.96297; 240.25971, 631.36117; 259.86174, 612.60253; ...
291.85381, 556.70385; 315.52139, 537.56387; 341.63663, 520.12519; 351.37130, 458.75372; ...
349.33183, 431.31454; 328.80465, 412.43055];
inner = [ ...
238.64853, 266.58978; 235.14026, 287.95183; 238.20736, 303.46785; 250.13902, 303.71290; ...
258.51675, 297.46973; 274.55300, 291.27357; 284.66230, 280.72063; 279.73288, 267.83455; ...
270.68478, 255.55440; 255.73801, 249.16872; 241.72690, 256.73448];
fixed = [outer; inner];
```

[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

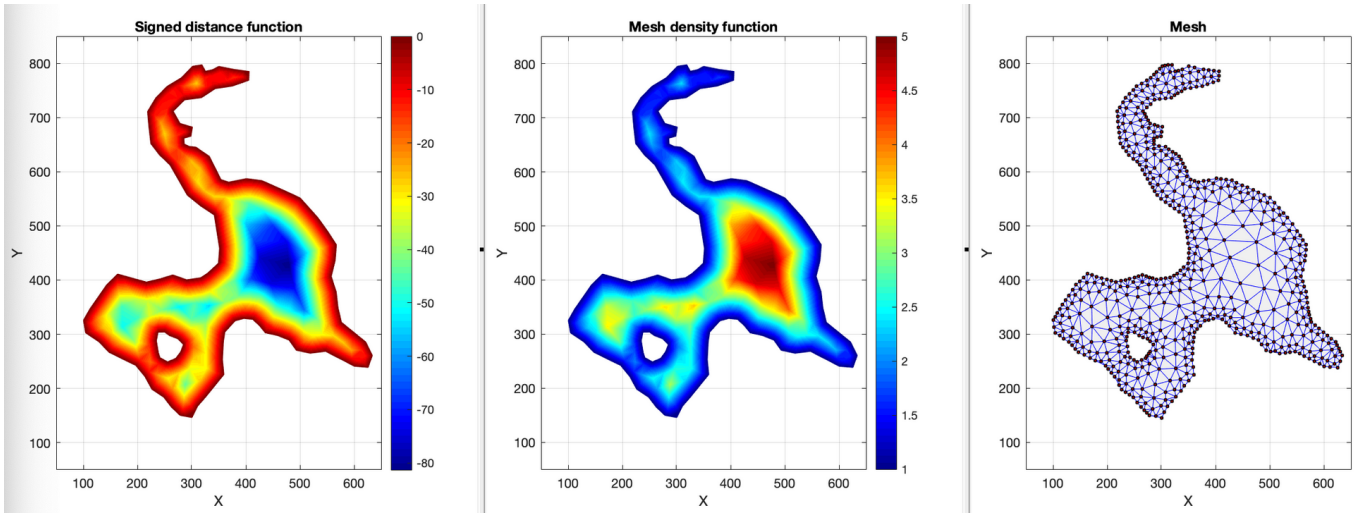
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = p14_fd(p)
    v1 = [ same contents as outer matrix; coordinate of the first point in outer matrix];
    v2 = [ same contents as inner matrix; coordinate of the first point in inner matrix];
    d1 = dpoly(p,v1);
    d2 = dpoly(p,v2);
    d = ddiff(d1,d2);
end
```

② Mesh density function

```
function h = p14_fh(p)
    hmax = 5;
    hmin = 1;
    dmax = max(abs(p14_fd(p)));
    dmin = min(abs(p14_fd(p)));
    h = ((dmax-abs(p14_fd(p)))*hmin+(abs(p14_fd(p))-dmin)*hmax)/(dmax-dmin);
end
```



(11) Reuleaux triangle

```
filename = 'p17';
fd = @p17_fd;
fh = @p17_fh;
h = 0.05;
meshbox = [0.0, -1.0; 5.0, 1.0];
iteration_max = 50;
fixed = [ ];
```

[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

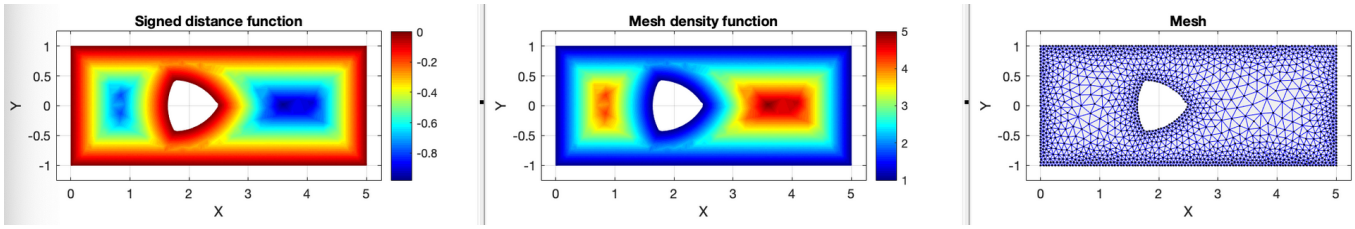
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = p17_fd(p)
    alpha = 0.0;
    d1 = drectangle(p,0.0,5.0,-1.0,1.0);
    r = 0.5;
    cx1 = 2.0+r*cos(alpha);
    cy1 = 0.0+r*sin(alpha);
    dc1 = dcircle(p,cx1,cy1,sqrt(3)*r);
    cx2 = 2.0+r*cos(alpha+2*pi/3);
    cy2 = 0.0+r*sin(alpha+2*pi/3);
    dc2 = dcircle(p,cx2,cy2,sqrt(3)*r);
    cx3 = 2.0+r*cos(alpha+4*pi/3);
    cy3 = 0.0+r*sin(alpha+4*pi/3);
    dc3 = dcircle(p,cx3,cy3,sqrt(3)*r);
    d2 = max(max(dc1,dc2),dc3);
    d = ddiff(d1,d2);
end
```

② Mesh density function

```
function h = p17_fh(p)
    hmax = 5;
    hmin = 1;
    dmax = max(abs(p17_fd(p)));
    dmin = min(abs(p17_fd(p)));
    h = ((dmax-abs(p17_fd(p)))*hmin+(abs(p17_fd(p))-dmin)*hmax)/(dmax-dmin);
end
```



(12) A dumbbell with nonuniform mesh density

```
filename = 'p18_2';
fd = @p18_fd;
fh = @p18_nonuniform_fh;
h = 0.1;
meshbox = [-3.0,-1.0; 3.0,1.0];
iteration_max = 50;
r = 1.0;
height = 1.0/3.0;
y_corner = height;
x_corner = 1.0+r-sqrt((r+height)*(r-height));
fixed = [-x_corner,-y_corner; x_corner,-y_corner; x_corner,y_corner; -x_corner,y_corner];
[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);
figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Signed distance function');
figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Mesh density function');
p = p';
t = t';
figure('color','w'); plotmesh(p,t,filename); title('Mesh');
```

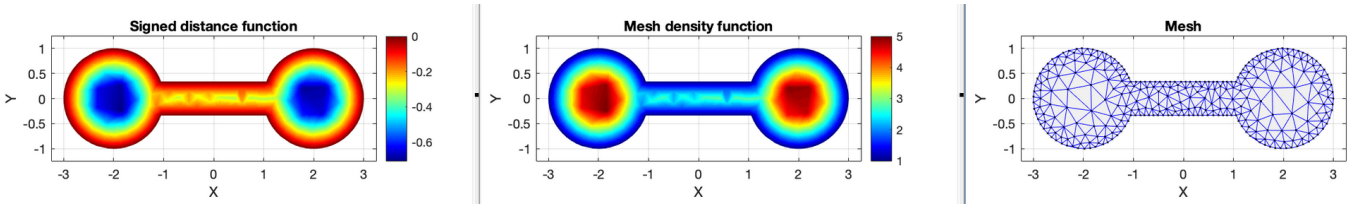
① Signed distance function

```
function d = p18_fd(p)
    r1 = 1.0;
```

```
    r2 = 1.0;
    c1 = [-2.0, 0.0];
    c2 = [2.0, 0.0];
    width = 2.0;
    height = 1.0/3.0;
    d1 = sqrt((p(:,1)-c1(1)).^2+(p(:,2)-c1(2)).^2)-r1;
    d2 = sqrt((p(:,1)-c2(1)).^2+(p(:,2)-c2(2)).^2)-r2;
    d3 = drectangle(p,-width,width,-height,height);
    d = min(d1,min(d2,d3));
end
```

② Mesh density function

```
function h = p18_nonuniform_fh(p)
    hmax = 5;
    hmin = 1;
    dmax = max(abs(p18_fd(p)));
    dmin = min(abs(p18_fd(p)));
    h = ((dmax-abs(p18_fd(p)))*hmin+(abs(p18_fd(p))-dmin)*hmax)/(dmax-dmin);
end
```



(13) Hans-Werner van Wyk's Big C

```
filename = 'p22';
fd = @p22_fd;
fh = @p22_fh;
h = 0.08;
meshbox = [-1.0, -1.0; 1.0, 1.0];
iteration_max = 200;
fixed = [0.4, 0.1; 1, 0.1; 0.4, -0.1; 1, -0.1];
```

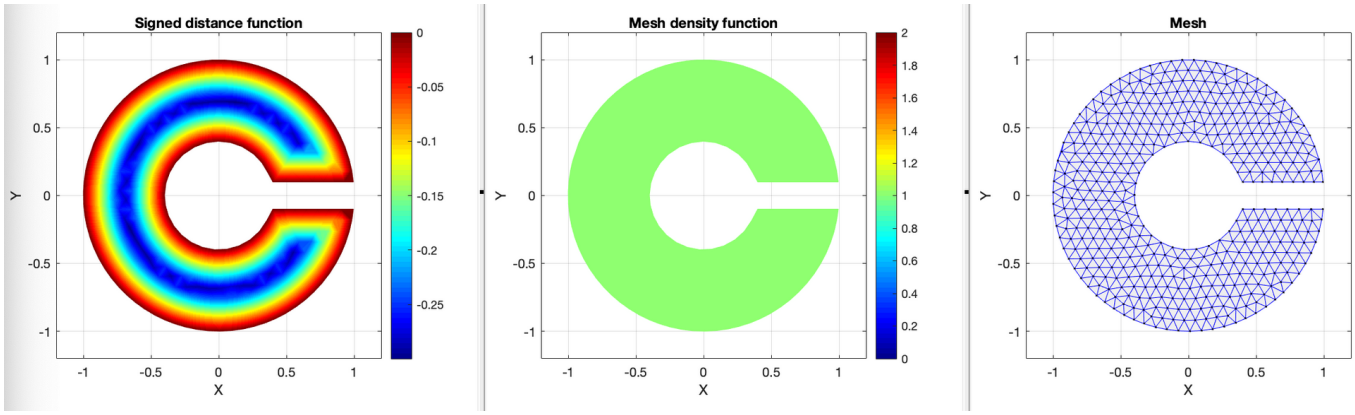
```
[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);
figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Signed distance function');
figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Mesh density function');
p = p';
t = t';
figure('color','w'); plotmesh(p,t,filename); title('Mesh');
```

① Signed distance function

```
function d = p22_fd(p)
    d1 = dcircle(p,0.0,0.0,1.0);
    d2 = dcircle(p,0.0,0.0,0.4);
    d12 = ddiff(d1,d2);
    d3 = drectangle(p,0.0,1.0,-0.1,0.1);
    d = ddiff(d12,d3);
end
```

② Mesh density function

```
function h = p22_fh(p)
    h = ones(size(p,1),1);
end
```



(14) Mike Schneier's nonuniform square

```
filename = 'p23';
fd = @p23_fd;
fh = @p23_fh;
h = 0.025;
meshbox = [0.0, 0.0; 1.0, 1.0];
iteration_max = 200;
fixed = [0.0,0.0; 0.0,1.0; 1.0,0.0; 1.0,1.0];
```

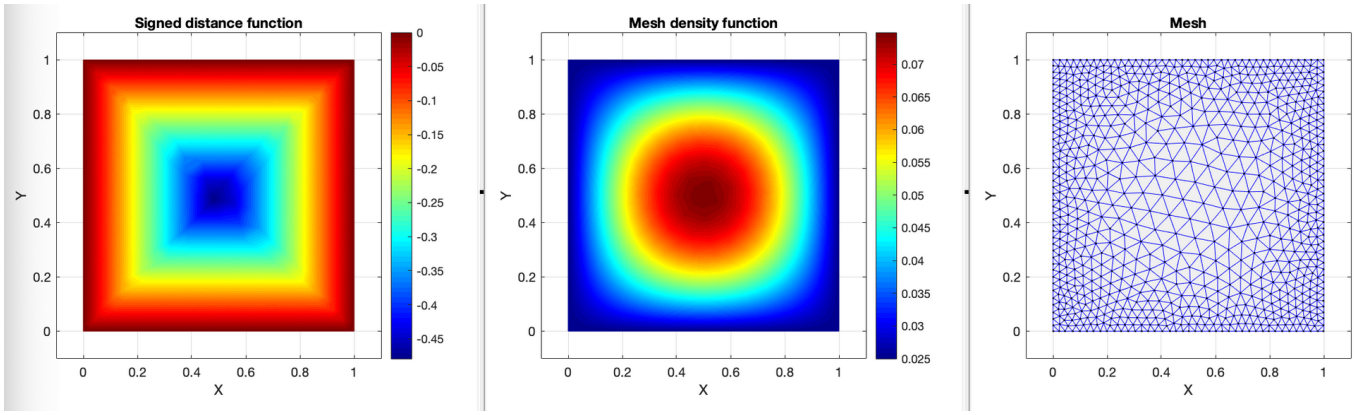
```
[p,t] = distmesh_2d(fd,fh,h,meshbox,iteration_max,fixed);
figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Signed distance function');
figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Mesh density function');
p = p';
t = t';
figure('color','w'); plotmesh(p,t,filename); title('Mesh');
```

① Signed distance function

```
function d = p23_fd(p)
    d = drectangle ( p, 0.0, 1.0, 0.0, 1.0 );
end
```

② Mesh density function

```
function h = p23_fh(p)
    np = size(p,1);
    h = 0.025+0.05*sin(pi*p(1:np,1)).*sin(pi*p(1:np,2));
end
```



9. MATLAB code demonstration in 3D problems: Generate unstructured tetrahedral meshes

(1) Problem 1

```
filename = 'q01';
fd = @q01_fd;
fh = @q01_fh;
h = 0.2;
meshbox = [0.0, 0.0, 0.0; 3.0, 1.0, 1.0];
```

```
iteration_max = 200;
fixed = [0.0,0.0,0.0; 0.0,0.0,1.0; 0.0,1.0,0.0; 0.0,1.0,1.0; 3.0,0.0,0.0; 3.0,0.0,1.0;
3.0,1.0,0.0; 3.0,1.0,1.0];
```

[p,t] = distmesh_3d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

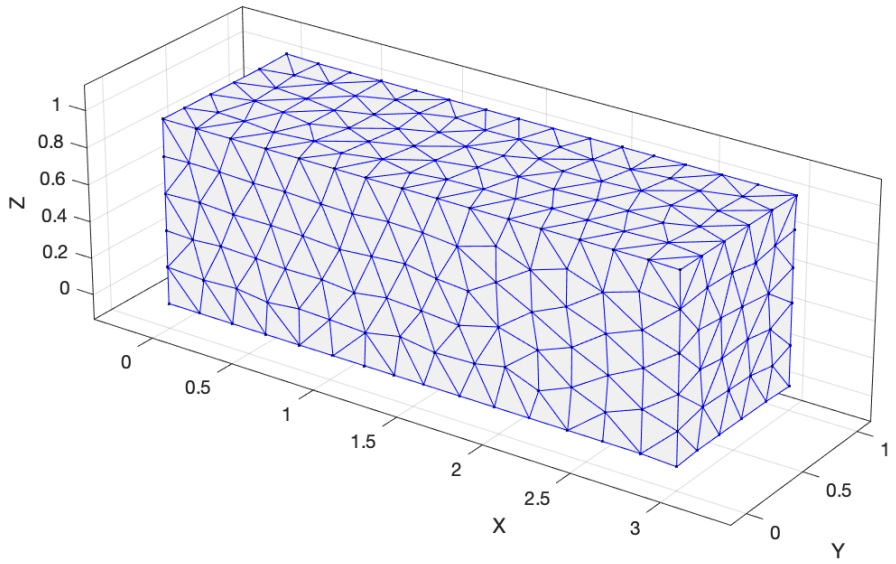
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = q01_fd(p)
    d = -min(min(min(min(min( ...
      -0.0+p(:,3),1.0-p(:,3)),-0.0+p(:,2)),1.0-p(:,2)),-0.0+p(:,1)),3.0-p(:,1));
end
```

② Mesh density function

```
function h = q01_fh(p)
    h = ones(size(p,1),1);
end
```



(2) Problem 2

```
filename = 'q02';
fd = @q02_fd;
fh = @q02_fh;
h = 0.1;
meshbox = [0.0, 0.0, 0.0; 1.0, 1.0, 4.0];
iteration_max = 200;
fixed = [ ];
```

[p,t] = distmesh_3d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

figure('color','w'); plotmesh(p,t,filename); title('Mesh');

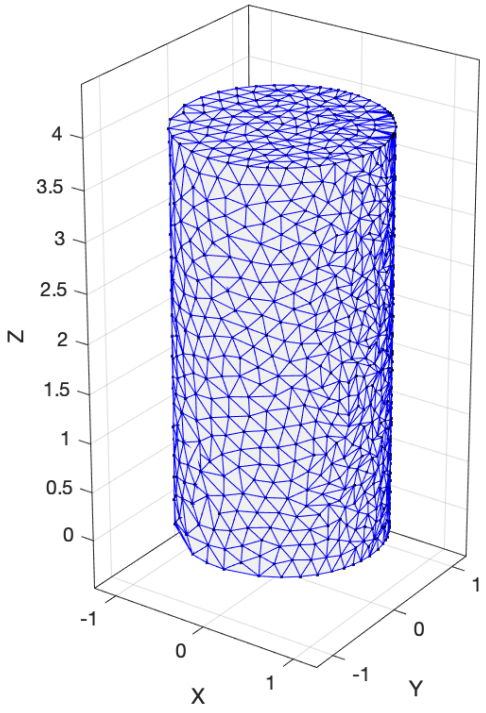
① Signed distance function

```
function d = q02_fd(p)
```

```
    d = -min(min(-0.0+p(:,3),4.0-p(:,3)),1.0-sqrt(sum(p(:,1:2).^2,2)));
end
```

② Mesh density function

```
function h = q02_fh(p)
    h = ones(size(p,1),1);
end
```



(3) Problem 3

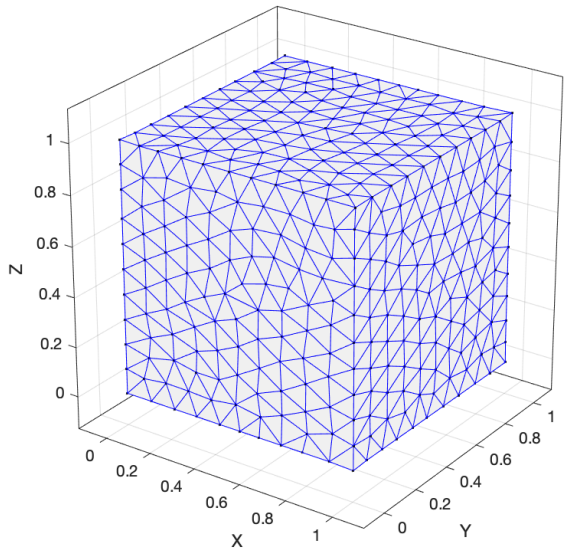
```
filename = 'q03';
fd = @q03_fd;
fh = @q03_fh;
h = 0.1;
meshbox = [0.0, 0.0, 0.0; 1.0, 1.0, 4.0];
iteration_max = 200;
fixed = [...
0.0, 0.0, 0.0; 0.0, 0.0, 1.0; 0.0, 1.0, 0.0; 0.0, 1.0, 1.0; ...
1.0, 0.0, 0.0; 1.0, 0.0, 1.0; 1.0, 1.0, 0.0; 1.0, 1.0, 1.0];
[p,t] = distmesh_3d(fd,fh,h,meshbox,iteration_max,fixed);
figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Signed distance function');
figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');
colorbar; colormap jet; title('Mesh density function');
p = p';
t = t';
figure('color','w'); plotmesh(p,t,filename); title('Mesh');
```

① Signed distance function

```
function d = q03_fd(p)
    d = -min(min(min(min(min( ...
      -0.0+p(:,3),1.0-p(:,3)),-0.0+p(:,2)),1.0-p(:,2)),-0.0+p(:,1)),1.0-p(:,1));
end
```

② Mesh density function

```
function h = q03_fh(p)
    h = ones(size(p,1),1);
end
```



(4) Problem 4

```
filename = 'q04';
fd = @q04_fd;
fh = @q04_fh;
h = 0.15;
meshbox = [-1.0, -1.0, -1.0; 1.0, 1.0, 1.0];
iteration_max = 200;
fixed = [ ];
```

[p,t] = distmesh_3d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

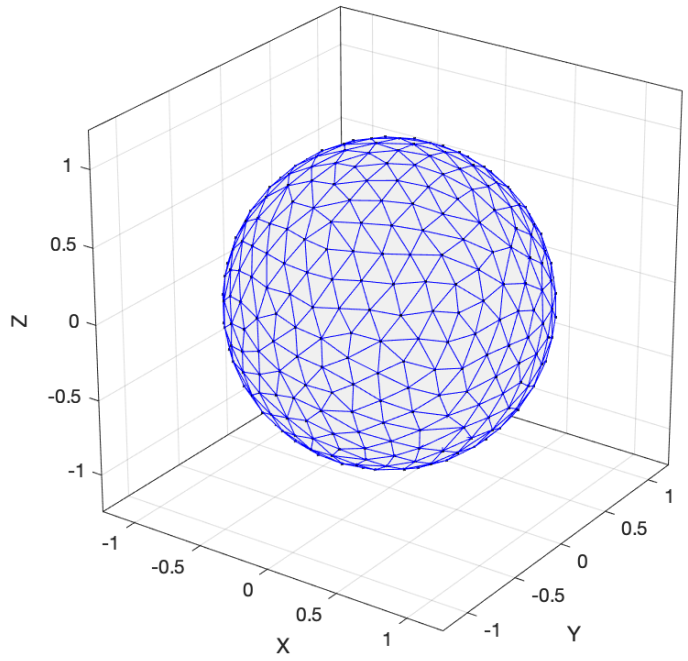
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = q04_fd(p)
    d = sqrt(sum(p.^2,2))-1.0;
end
```

② Mesh density function

```
function h = q04_fh(p)
    h = ones(size(p,1),1);
end
```



(5) Problem 5

```
filename = 'q05';
fd = @q05_fd;
fh = @q05_fh;
h = 0.075;
meshbox = [-5, -5, -1; 5, 5, 1];
iteration_max = 300;
fixed = [ ];
```

[p,t] = distmesh_3d(fd,fh,h,meshbox,iteration_max,fixed);

figure('color','w'); dist_plot(p,t,fd); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Signed distance function');

figure('color','w'); dist_plot(p,t,fh); axis equal on; box on; xlabel('X'); ylabel('Y');

colorbar; colormap jet; title('Mesh density function');

p = p';

t = t';

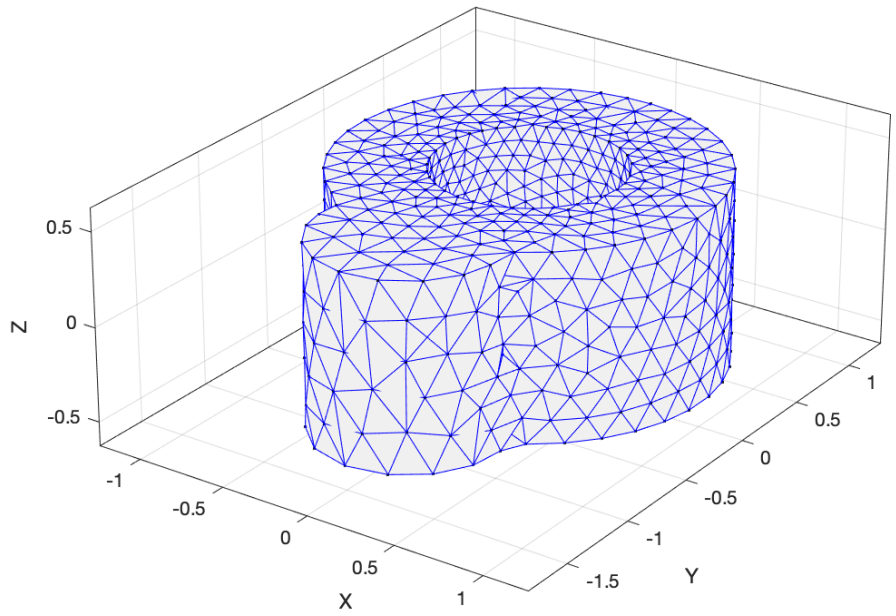
figure('color','w'); plotmesh(p,t,filename); title('Mesh');

① Signed distance function

```
function d = q05_fd(p)
    z1 = -(p(:,3)-(-0.5));
    z2 = -(0.5-p(:,3));
    r1 = -(1-sqrt((p(:,1)-(0)).^2+(p(:,2)-(0)).^2));
    r2 = -(0.5-sqrt((p(:,1)-(0)).^2+(p(:,2)-(0)).^2));
    r3 = -(0.5-sqrt((p(:,1)-(0)).^2+(p(:,2)-(-1)).^2));
    d0 = max(z1,z2);
    d1 = max(d0,r1);
    d2 = max(d1,-r2);
    d3 = max(z1,z2);
    d4 = max(d3,r3);
    d5 = min(d2,d4);
    d = d5;
end
```

② Mesh density function

```
function h = q05_fh(p)
    h = 1*(sqrt((p(:,1)-(0)).^2+(p(:,2)-(0)).^2)-0.5+0.6);
end
```



10. MATLAB code demonstration for converting linear triangular and tetrahedral elements into quadratic meshes

(1) Converte linear triangular elements into quadratic meshes of 2D problems

① The same number of triangles are used, but each triangle is given three extra midside nodes. The coordinates of these nodes are determined by averaging the coordinates of pairs of vertices of the triangles.

② Such refinement operation in *DistMesh* is straightforward. The usage of the *DistMesh* code:

```
triangulation_l2q(prefix);
```

where `prefix` is the filename prefix for the existing files (input files) of:

① `prefix_nodes.txt` containing the node coordinates of the original linear meshes.

② `prefix_elements.txt` containing the element definitions of the original linear meshes.

③ The corresponding output files are;

④ `prefix_l2q_nodes.txt` containing the node coordinates of the newly quadratic meshes.

⑤ `prefix_l2q_elements.txt` will contain the element definitions of the newly quadratic meshes.

⑥ For example, in the first demonstration generating unstructured triangular meshesof 2D problems (A circle, with spacings h = 0.4, 0.3, 0.2), the `prefix` is `'p01'`.

(2) Converte linear tetrahedral elements into quadratic meshes of 3D problems

① The same number of tetrahedrons are used, but each tetrahedron is given six extra midside nodes. The coordinates of these nodes are determined by averaging the coordinates of pairs of vertices of the tetrahedrons.

② Similarly, the usage of the *DistMesh* code:

```
tet_mesh_l2q(prefix);
```

where `prefix` is the filename prefix for the existing files (input files) of node coordinates and element definitions for the original linear meshes. The corresponding output files are also `prefix_l2q_nodes.txt` and `prefix_l2q_elements.txt`.

2.13 Code for post-processing of finite element equation results

1. After solving the finite element equation: $\boldsymbol{Ku} = \boldsymbol{r}$ for the results:

(2.81) $\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{r}$

by the code:

```
uglob = kglob\rglob;
```

the post-processing can be done as follows.

2. Coordinates of deformed meshes for visualization

```
scalefac = 10; % Set a scaling factor for visualization
coordef = zeros(ndime,nnode);
for i = 1:nnode
```

```
    for j = 1:ndime
      coordef(j,i) = coor(j,i)+uglob(ndime*(i-1)+j)*scalefac;
    end
end
```

3. Element strain, stress, and integration point position

```
M = numIntegPt(ndime,nelnd);
cmat = MatStif(ndime,mate);
elemip = zeros(ndime,M,nelem);
if(ndime == 2)
    elemsn = zeros(3,M,nelem);
    elemss = zeros(3,M,nelem);
else
    elemsn = zeros(6,M,nelem);
    elemss = zeros(6,M,nelem);
end
coorie = zeros(ndime,nelnd);
uglobie = zeros(ndime,nelnd);
xii = zeros(ndime,1);
dxdxi = zeros(ndime,ndime);
dNdx = zeros(nelnd,ndime);
sn = zeros(ndime,ndime);
ss = zeros(ndime,ndime);
for ie = 1:nelem
    for a = 1:nelnd
      for i = 1:ndime
        coorie(i,a) = coor(i,conn(a,ie));
      end
      for i = 1:ndime
        uglobie(i,a) = uglob(ndime*(conn(a,ie)-1)+i);
      end
    end
    for im = 1:M
      for i = 1:ndime
        xii(i) = xi(i,im);
      end
      N = ShpFunc(nelnd,ndime,xii);
      dNdxi = ShpFuncDeri(nelnd,ndime,xii);
      for i = 1:ndime
        for a = 1:nelnd
          elemip(i,im,ie) = elemip(i,im,ie) + coori(i,a)*N(a);
        end
      end
      dxdxi = 0;
      for i = 1:ndime
        for j = 1:ndime
          for a = 1:nelnd
            dxdxi(i,j) = dxdxi(i,j)+coorie(i,a)*dNdxi(a,j);
          end
        end
      end
      dxidx = inv(dxdxi);
      dNdx = 0.;
      for a = 1:nelnd
        for i = 1:ndime
```

```
          for j = 1:ndime
            dNdx(a,i) = dNdx(a,i)+dNdxi(a,j)*dxidx(j,i);
          end
        end
      end
      sn = 0;
      for i = 1:ndime
        for j = 1:ndime
          for a = 1:nelnd
            sn(i,j) = sn(i,j)+0.5*(uglobie(i,a)*dNdx(a,j)+uglobie(j,a)*dNdx(a,i));
          end
        end
      end
      if(ndime == 2)
        elemsn(1,im,ie) = sn(1,1);
        elemsn(2,im,ie) = sn(2,2);
        elemsn(3,im,ie) = sn(1,2);
      else
        elemsn(1,im,ie) = sn(1,1);
        elemsn(2,im,ie) = sn(2,2);
        elemsn(3,im,ie) = sn(3,3);
        elemsn(4,im,ie) = sn(2,3);
        elemsn(5,im,ie) = sn(3,1);
        elemsn(6,im,ie) = sn(1,2);
      end
      ss = 0;
      for i = 1:ndime
        for j = 1 : ndime
          for k = 1 : ndime
            for l = 1: ndime
              ss(i,j) = ss(i,j)+cmat(i,j,k,l)*sn(k,l);
            end
          end
        end
      end
      if(ndime == 2)
        elemss(1,im,ie) = ss(1,1);
        elemss(2,im,ie) = ss(2,2);
        elemss(3,im,ie) = ss(1,2);
      else
        elemss(1,im,ie) = ss(1,1);
        elemss(2,im,ie) = ss(2,2);
        elemss(3,im,ie) = ss(3,3);
        elemss(4,im,ie) = ss(2,3);
        elemss(5,im,ie) = ss(3,1);
        elemss(6,im,ie) = ss(1,2);
      end
    end
end
```