《计算流体力学基础》第一次作业

生成绕 NACA0012 翼型的 C 网格。 NACA0012 翼型是一个对称翼型，上表面可以用如下方程近似：

$$y1(x)=0.6(0.2969\sqrt{x}-0.126x-0.3516x^2+0.2843x^3-0.1015x^4), x\in[0,1]$$

下表面 y2(x) 和上表面是关于 x 轴对称的，因此

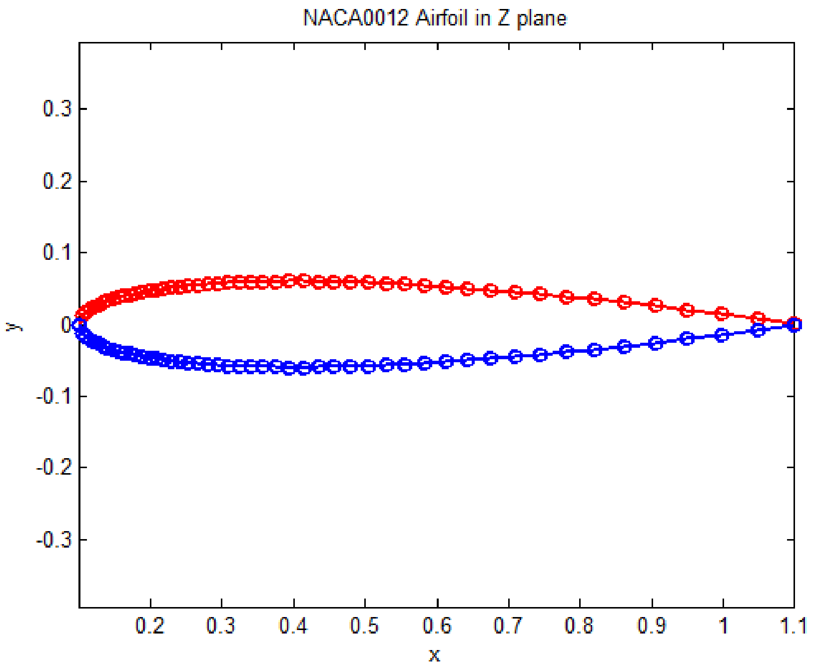
$y2(x)=-y1(x)$。

学号： 150*******

专业： 流体力学

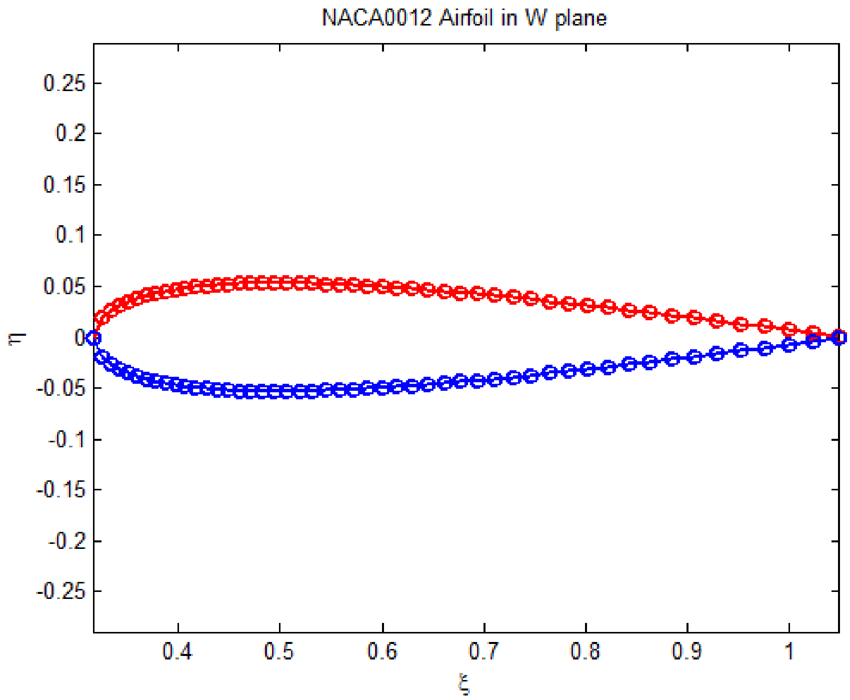
姓名： 杨阳

首先根据翼型多项式在 Z 平面中的翼型表面取控制点，其中我们把翼型向右平移 0.1。

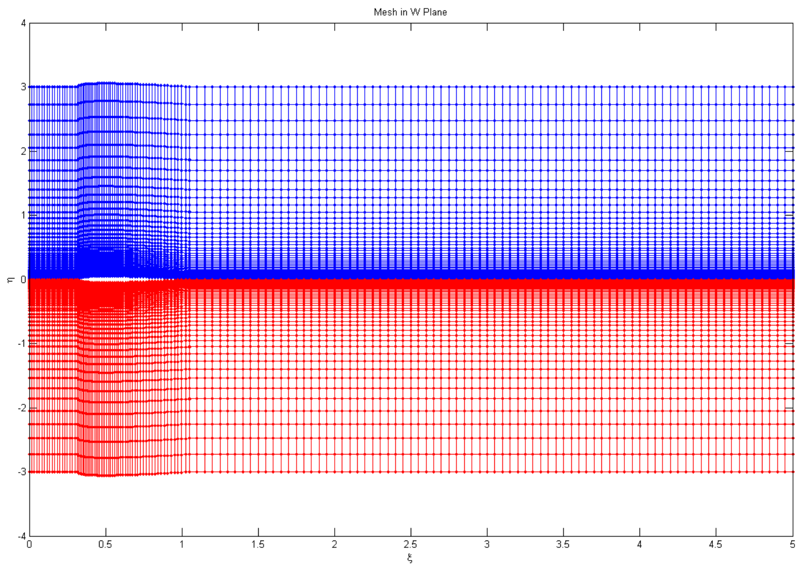


在取点过程中使用网格间距函数使得翼型表面的网格点的间距成等比数列的方式排列，通过调整公比来控制网格点的分布。

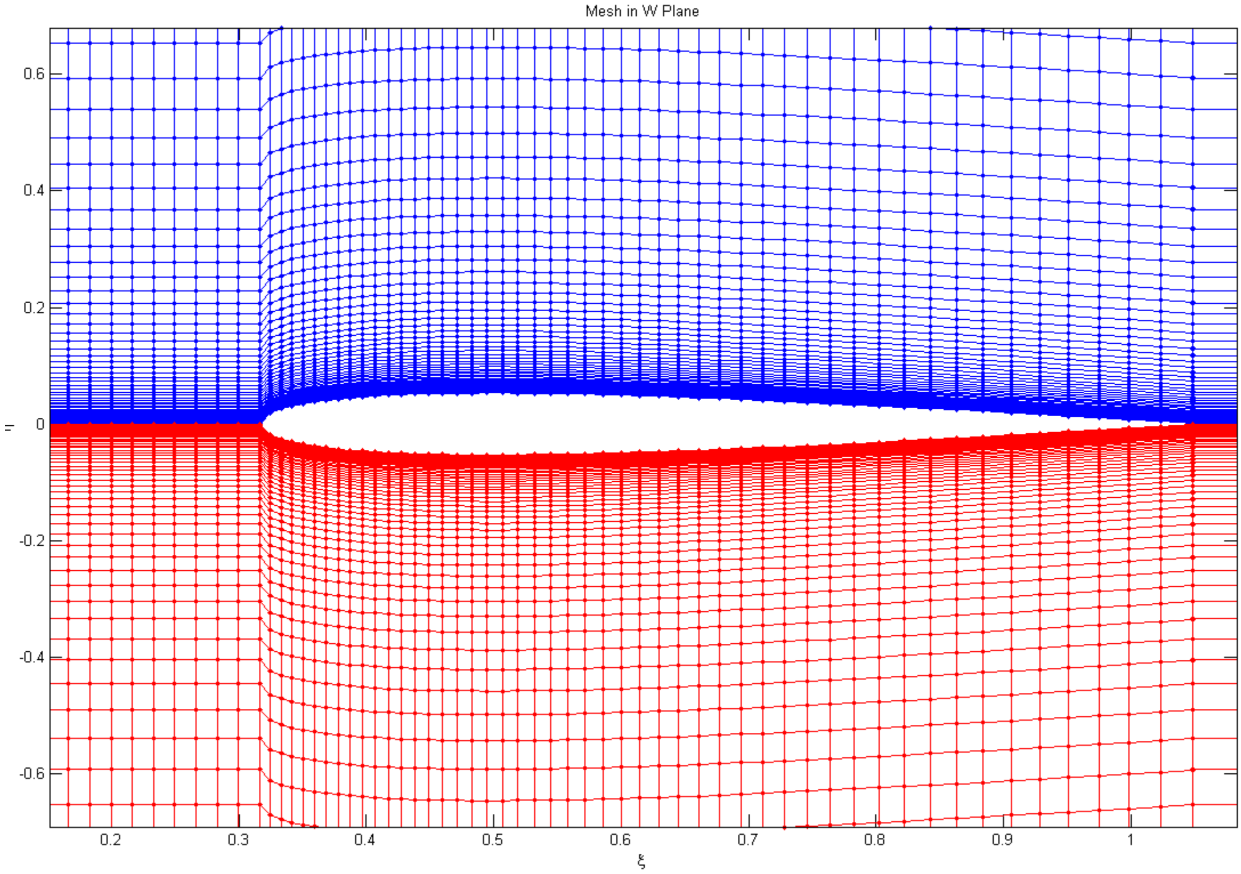
下一步应用复变换$w=\sqrt{z}$将翼型表面的控制点映射到 W 平面，W 平面上面的翼型如下：



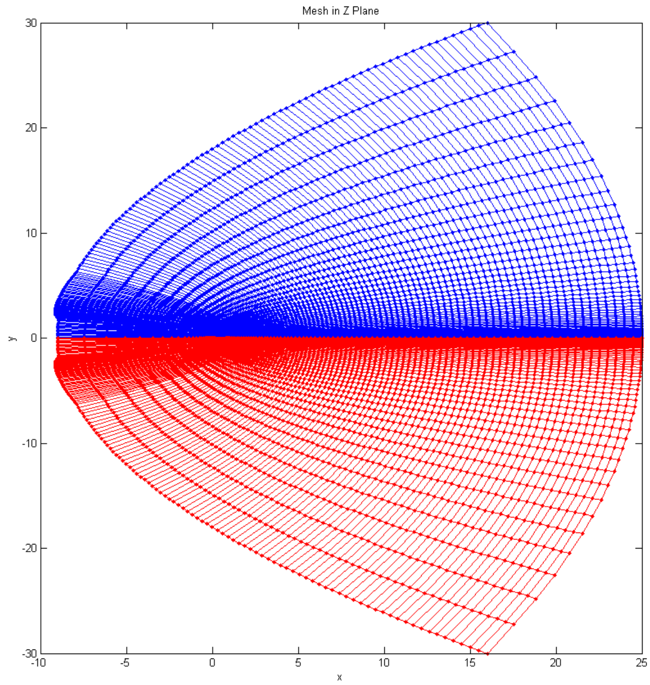
下面在 W 平面上面划分网格，将记忆所在的边向上平移，平移的间距也利用网格间距函数进行变换。下面是 W 平面上面的网格：

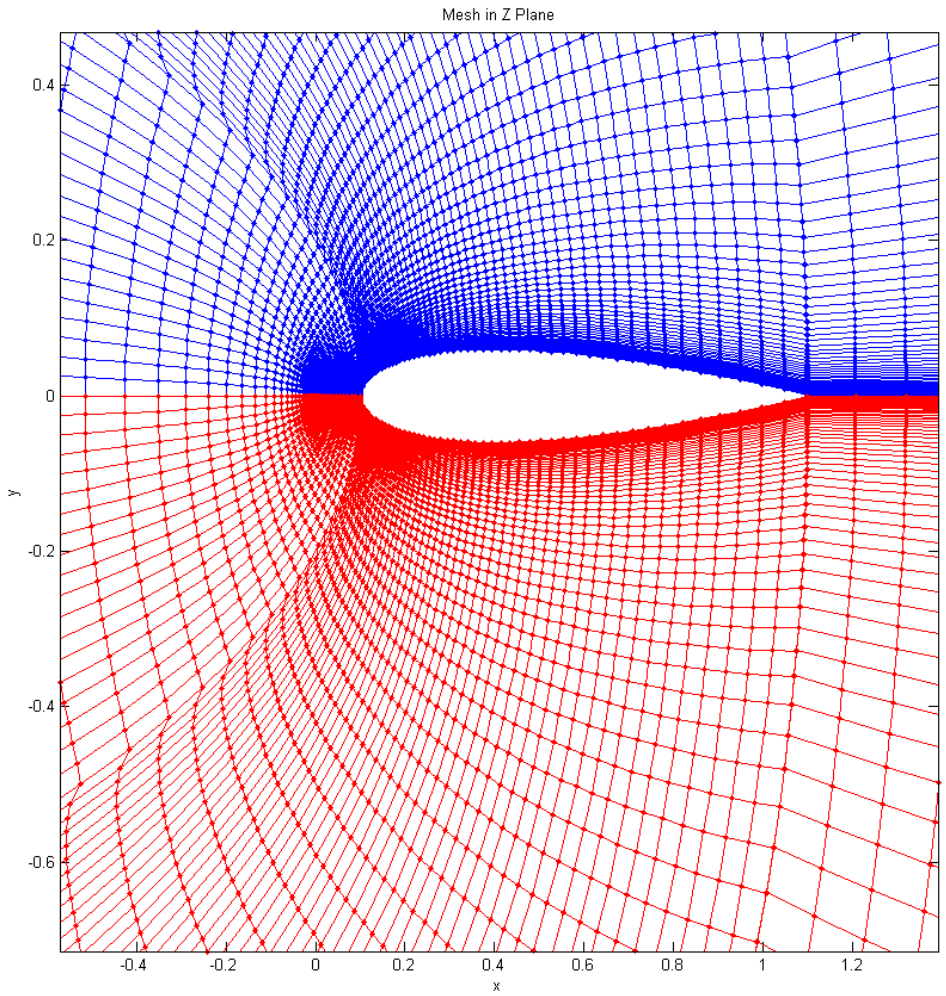


翼型表面的网格细节：

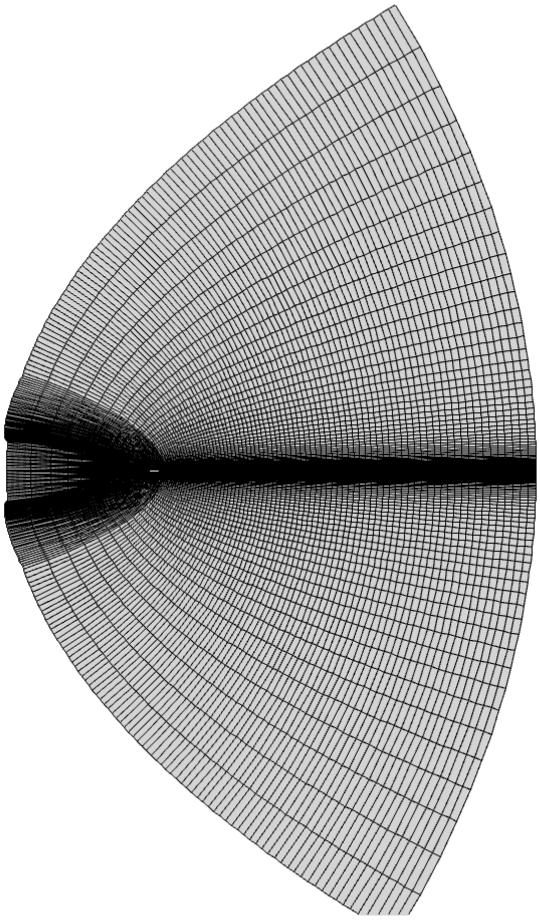


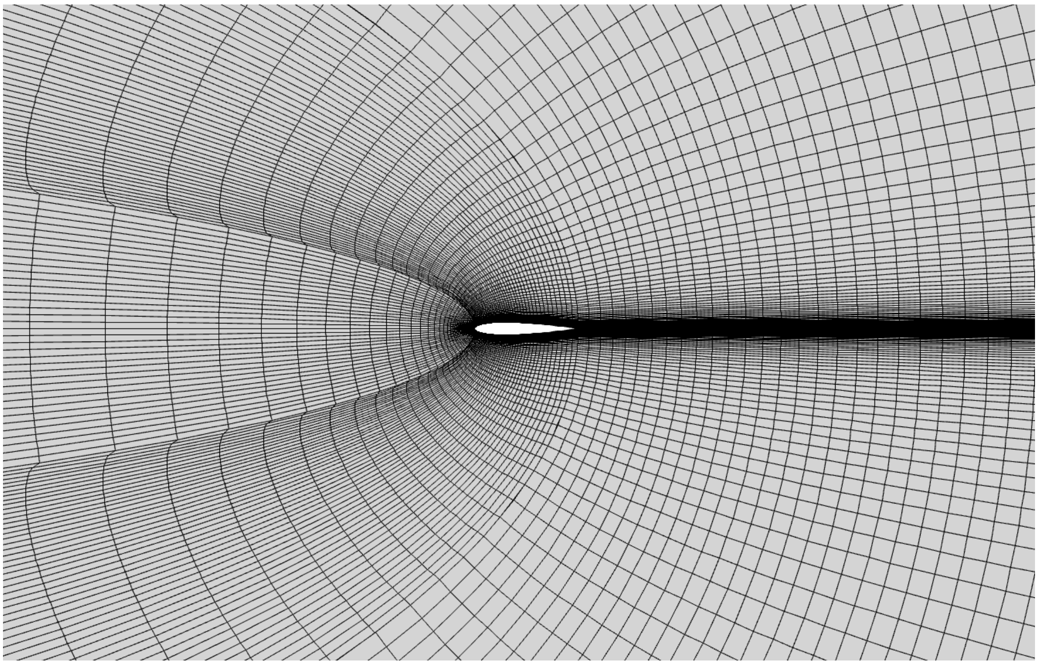
下面利用逆变换 $z = w^2$ 将网格变换会 Z 平面，得到的 Z 平面上的网格为：





在程序的最后将MATLAB中生成的网格写成Tecplot格式，并倒入tecplot软件绘图：





附录：MATLAB 代码

1. NACA0012CMesh.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%
%%%%%% Program name: NACA0012CMesh.m
%%%%%% Program Aurthor: Yang Yang
%%%%%% Date: 2015.09.23
%%%%%% Version 1.0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%

%% Geometry parameters
x0 = 0.1;          % First points for airfoil

%% Mesh parameters
N_xi =  150;       % Mesh number in xi direction
N_eta =  100;      % Mesh number in eta direction

N_eps = 20;        % Mesh number before Airfoil
N = 50;            % Mesh number in Airfoil

xi_max = 5;        % maximun xi
eta_max = 3;       % maximun eta
```

```
%% Plot Airfoil in Z plane and W plane
x_span = [x0, x0+1];          % Airfoil span
x = meshfun1( x_span, N );    % generate nodes' x
y = NACA0012( x , x0);        % generate nodes' y
z = x + sqrt(-1)*y;

w = sqrt(z);                  % conformal mapping
xi = real(w);                 % separate xi
eta = imag(w);                % separate eta

% plot airfoil in Z plane
figure('Color',[1 1 1]);
plot(x,y,'r-o','linewidth',2);
hold on;
plot(x,-y,'b-o','linewidth',2);
title('NACA0012 Airfoil in Z plane');
xlabel('x');
ylabel('y');
axis equal;

% plot airfoil in W plane
figure('Color',[1 1 1]);
plot(xi,eta,'r-o','linewidth',2);
hold on;
```

```
plot(xi,-eta,'b-o','linewidth',2);
title('NACA0012 Airfoil in W plane');
xlabel('\xi');
ylabel('\eta');
axis equal;

%% Generate Mesh
mesh_xi = zeros(N_xi,N_eta);          % Array to store mesh
nodes' xi
mesh_eta = zeros(N_xi,N_eta);         % Array to store mesh
nodes' eta

% generate first xi edge
xi_before = linspace(0,min(xi),N_eps);
xi_behind = linspace(max(xi),xi_max, (N_xi - N_eps - N));
xi_edge_mesh_xi = [xi_before, xi, xi_behind];
xi_edge_mesh_eta = [zeros(1,length(xi_before)), eta,
zeros(1,length(xi_behind))];

% generate delta in eta edge
eta_span = [0, eta_max];
[eta_edge_mesh_eta,delta_eta] = meshfun2( eta_span,
N_eta );
```

```
% generate mesh

mesh_xi(:,1) =  xi_edge_mesh_xi;
mesh_eta(:,1) = xi_edge_mesh_eta;
for j = 2:N_eta
    for i = 1:N_xi
        mesh_xi(i,j) = mesh_xi(i,j-1);
        mesh_eta(i,j) = mesh_eta(i,j-1) + delta_eta(j-1);
    end
end

% Confromal Mapping
W = mesh_xi + sqrt(-1)*mesh_eta;
Z = W.^2;
X = real(Z);
Y = imag(Z);

%% Plot Mesh in MATLAB
% plot mesh in W plane
figure('Color',[1 1 1]);
for j = 1:N_eta
    for i = 1:N_xi - 1

plot([mesh_xi(i,j),mesh_xi(i+1,j)],[mesh_eta(i,j),mesh
```

```
_eta(i+1,j)],'b.-');
        hold on;
    end
end
for j = 1:N_eta
    for i = 1:N_xi - 1

plot([mesh_xi(i,j),mesh_xi(i+1,j)],[-mesh_eta(i,j),-me
sh_eta(i+1,j)],'r.-');
        hold on;
    end
end

for i = 1:N_xi
    for j = 1:N_eta - 1

plot([mesh_xi(i,j),mesh_xi(i,j+1)],[mesh_eta(i,j),mesh
_eta(i,j+1)],'b.-');
        hold on;
    end
end
for i = 1:N_xi
    for j = 1:N_eta - 1
```

```
plot([mesh_xi(i,j),mesh_xi(i,j+1)],[-mesh_eta(i,j),-me
sh_eta(i,j+1)],'r.-');
      hold on;
   end
end
hold off;
title('Mesh in W plane');
xlabel('\xi');
ylabel('\eta');

% plot mesh in Z plane
figure('Color',[1 1 1]);
for i = 1:N_xi
   for j = 1:N_eta - 1
      plot([X(i,j),X(i,j+1)],[Y(i,j),Y(i,j+1)],'b.-');
      hold on;
   end
end
for j = 1:N_eta
   for i = 1:N_xi - 1
      plot([X(i,j),X(i+1,j)],[Y(i,j),Y(i+1,j)],'b.-');
```

```
      hold on;
   end
end

for i = 1:N_xi
   for j = 1:N_eta - 1

plot([X(i,j),X(i,j+1)],[-Y(i,j),-Y(i,j+1)],'r.-');
      hold on;
   end
end
for j = 1:N_eta
   for i = 1:N_xi - 1

plot([X(i,j),X(i+1,j)],[-Y(i,j),-Y(i+1,j)],'r.-');
      hold on;
   end
end
hold off;
title('Mesh in Z plane');
xlabel('x');
ylabel('y');
```

```
%% Output Mesh in tecplot file format
fp = fopen('NACA0012CMesh.dat','w');
fprintf(fp,'TITLE = NACA0012\n');
fprintf(fp,'VARIABLES = "X", "Y"\n');
fprintf(fp,'ZONE I =%d, J =%d,F = point\n',N_xi,N_eta);
for j = 1:N_eta
    for i = 1:N_xi
        fprintf(fp,'%e, %e\n',X(i,j),Y(i,j));
    end
end
fclose(fp);
```

2.NACA0012.m

```
function y = NACA0012( x , x0)
% This function generate NACA0012 Airfoil Shape

x = x - x0;
y = 0.6*(0.2969*sqrt(x) - 0.126*x - 0.3516*x.^2 +
0.2843*x.^3 - 0.1015*x.^4);

end
```

3.meshfun1.m

```
function x = meshfun1( x_span, N )
% This function is designed for generate property mesh
factor = 1.05;                                   % space factor

delta = zeros(1,N-1);                            % Temporaray
array to store delta
x = zeros(1,N);                                 % store mesh node
a1 = (x_span(2) - x_span(1)) ...
    * (1 - factor) / (1- factor^(N-1));          % compute fist
element in delta

delta(1) = a1;                                   %Initial first
element in delta array
x(1) = x_span(1);                                %Initial fist
element in x array

for i = 2:N-1                                    % generate
equlibrium distribution nodes
    delta(i) = delta(i-1)*factor;
    x(i) = x(i-1) + delta(i-1);
end
x(i+1) = x(i) + delta(i);                        % The last
element in mesh nodes
```

```
end
```

4.meshfun2

```
function [x,delta] = meshfun2( x_span, N )
% This function is designed for generate property mesh
factor = 1.1;                            % space factor

delta = zeros(1,N-1);                      % Temporaray
array to store delta
x = zeros(1,N);                          % store mesh node
a1 = (x_span(2) - x_span(1)) ...
    * (1 - factor) / (1- factor^(N-1));    % compute fist
element in delta

delta(1) = a1;                           %Initial first
element in delta array
x(1) = x_span(1);                         %Initial fist
element in x array

for i = 2:N-1                            % generate
equlibrium distribution nodes
    delta(i) = delta(i-1)*factor;
```

```
    x(i) = x(i-1) + delta(i-1);
end
x(i+1) = x(i) + delta(i);                    % The last element
in mesh nodes

end
```