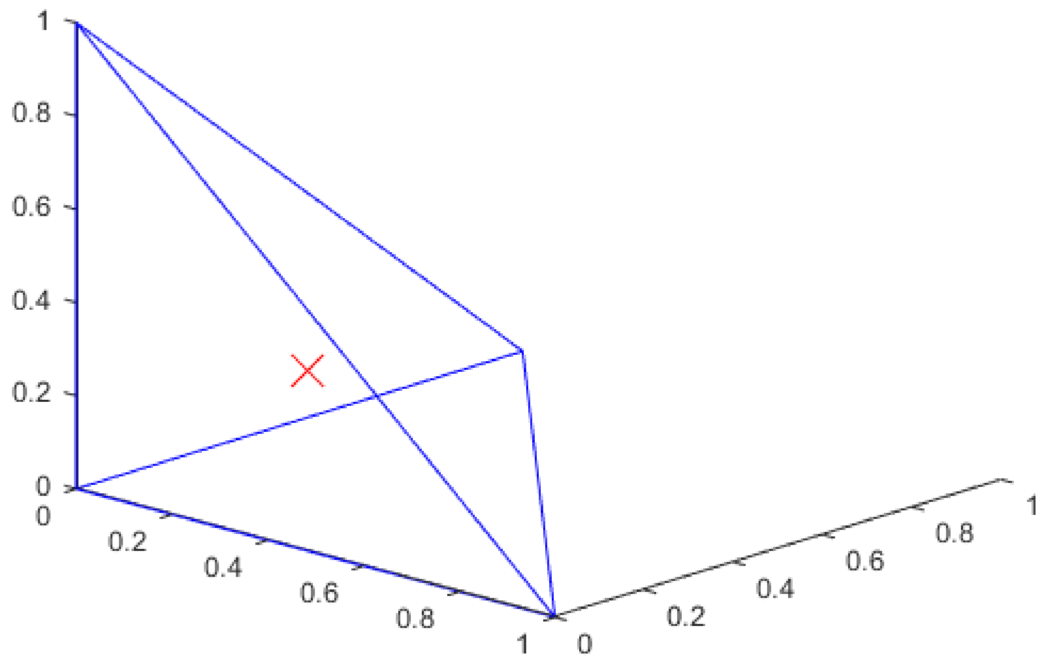
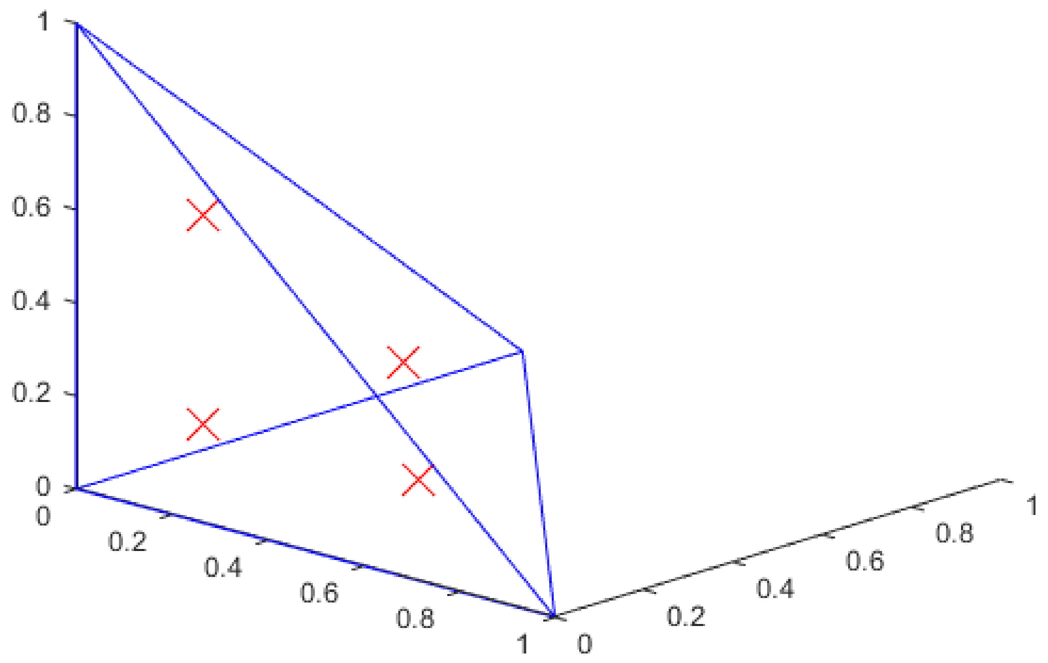
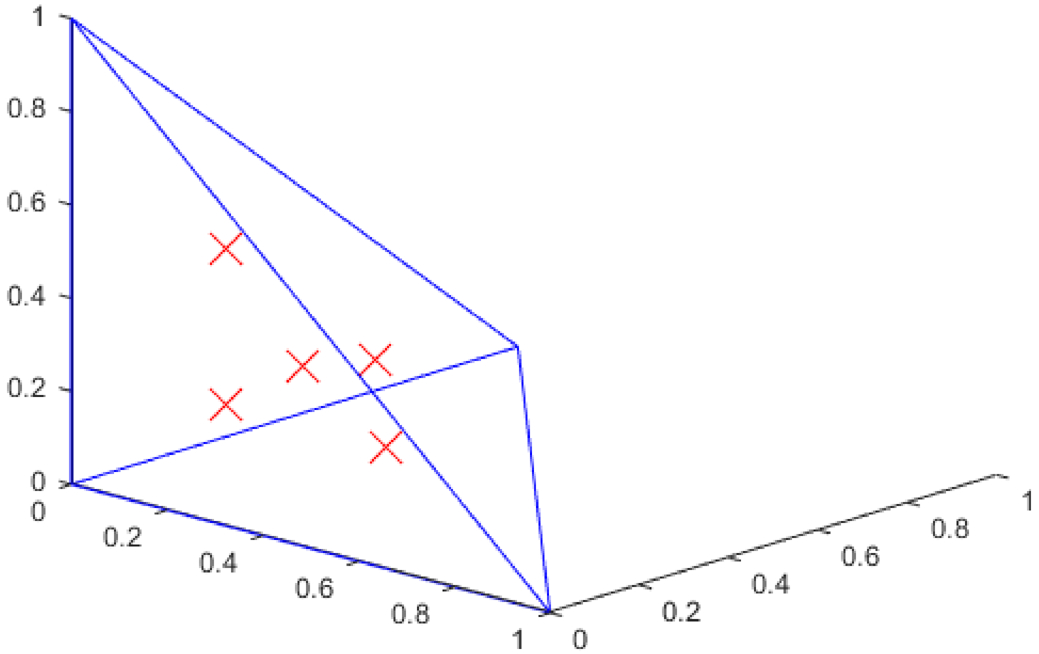
```matlab
close all;
clear all;
clc;
% Draw tetrahedron and its gauss points

x = [0 0 0;
     1 0 0;
     0 1 0;
     0 0 1];
% Tetrahedron vertex coordinates in isoparametric space

ix=[1 2 3 4 1 3 4 2];
% Tetrahedral connection order

nint=1:3;
% Order
for k=1:length(nint)
    [g, w] = TET4_GP(k);
    % Gauss points and weighs
    figure;
    patch('vertices', x, 'faces', ix, 'facecolor', 'none', 'edgecolor', 'b');
    % Draw tetrahedron in isoparametric space
    hold on;
    plot3(g(:, 1), g(:, 2), g(:, 3), 'marker', 'x', 'color', 'r', 'linestyle', 'none', ...
        'markersize', 16);
    % Draw gauss point of corresponding order (nint)
    view(43, 22);
    % Fixed viewing angle
end
% Contributed by Xiong
```



1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1



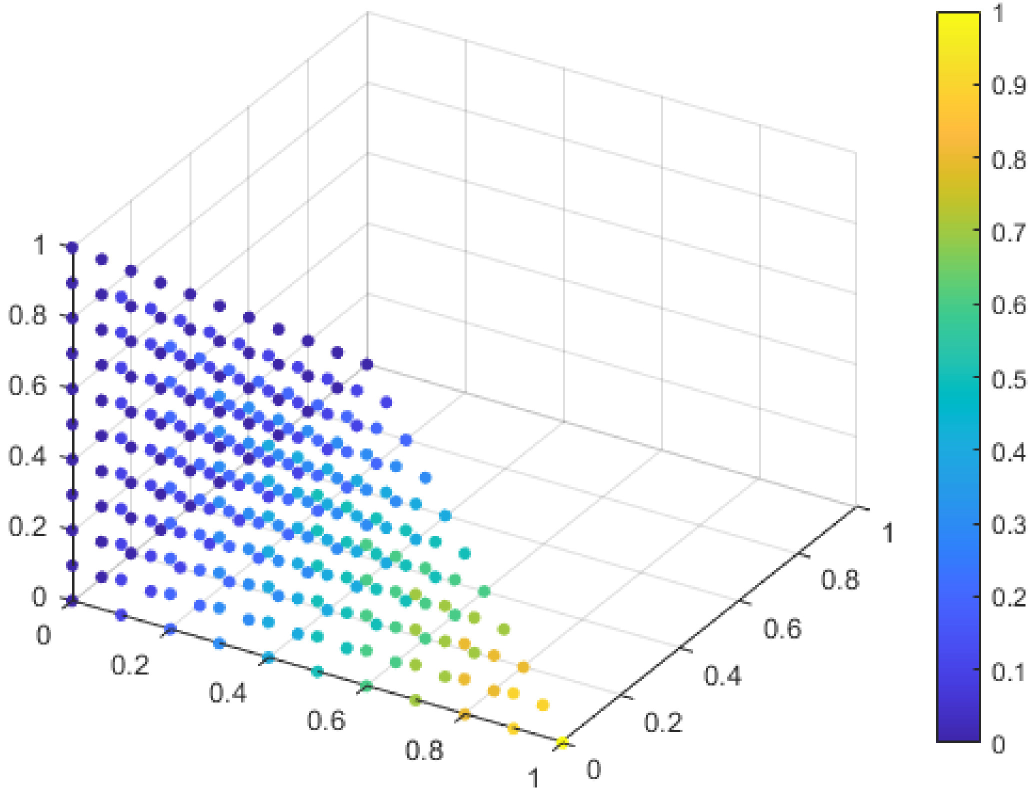
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1



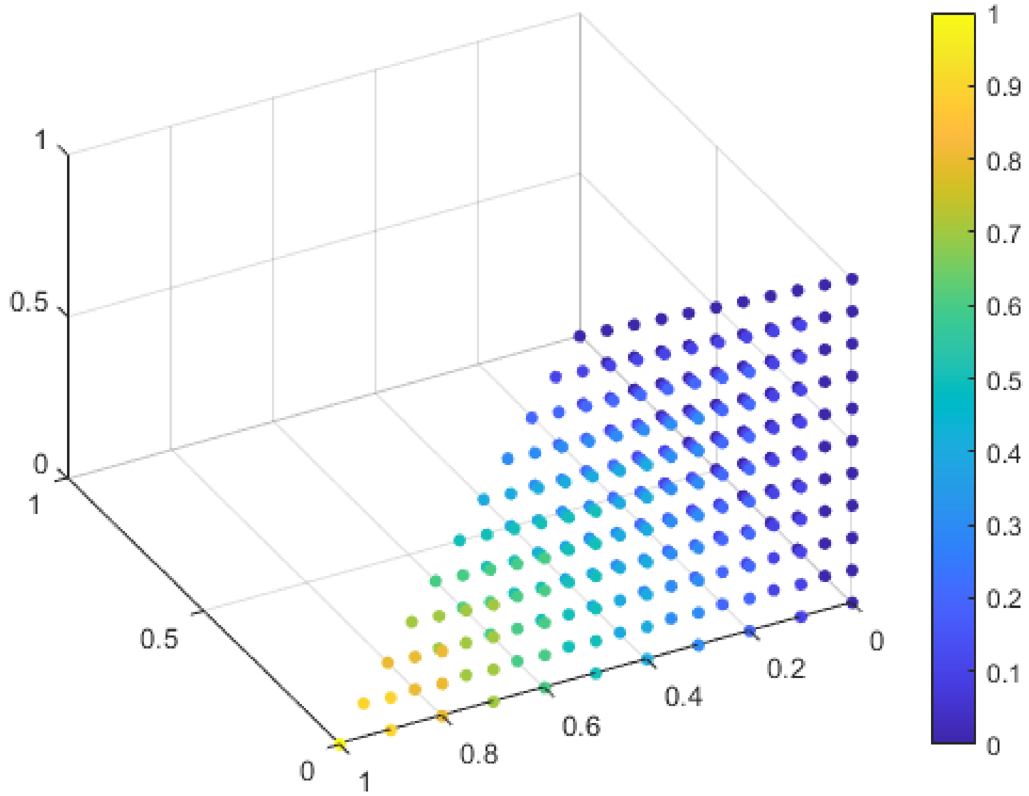
Published with MATLAB® R2019b

```
function [g, w] = TET4_GP(nint)
% Gauss points and wights
% [g,w] = TET4_GP(nint)

% nint: order of gauss points

% g : gauss point coordinates
% w : corresponding weigh

switch nint
    case 1
        g = [1/4, 1/4, 1/4];
        w = 1;
    case 2
        a = 0.58541020;
        b = 0.13819660;
        g = [a, b, b;
             b, a, b;
             b, b, a;
             b, b, b];
         w = 1/4 * ones(1, 4);
    case 3
        g = [1/4, 1/4, 1/4;
             1/2, 1/6, 1/6;
             1/6, 1/2, 1/6;
             1/6, 1/6, 1/2;
             1/6, 1/6, 1/6];
         w = [-4/5, 9/20, 9/20, 9/20, 9/20];
end
% Store gauss point data of order one to three
end
% Contributed by Xiong
```

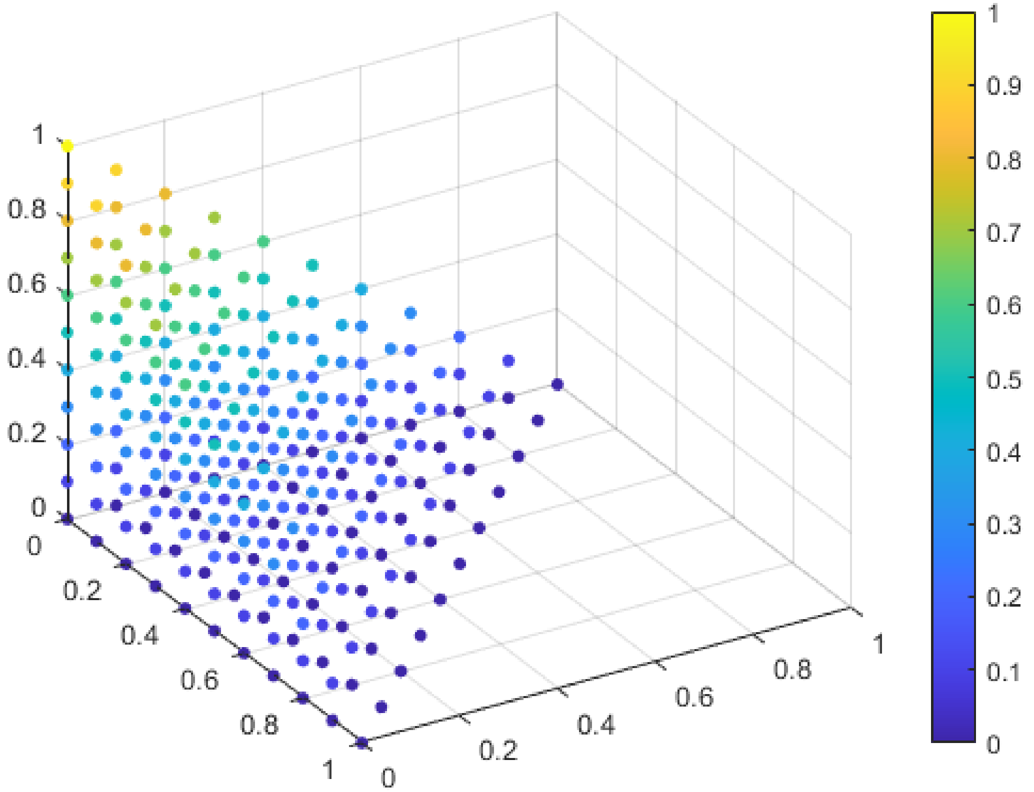
```matlab
close all;
clear all;
clc;
% show shape function in isoparametric space

[Xi_x,Xi_y,Xi_z]...
        =   meshgrid(0:0.1:1,0:0.1:1,0:0.1:1);
Xi_x    =   Xi_x(:);
Xi_y    =   Xi_y(:);
Xi_z    =   Xi_z(:);
Xi      =   [Xi_x(:),Xi_y(:),Xi_z(:)];
% creat grid points

Xi      =   Xi((Xi_x + Xi_y + Xi_z) <= 1,:);
% select the points inside and on the tetrahedron

NumXi   =   size(Xi,1);
N       =   zeros(NumXi,4);
for i   =   1:NumXi
    N(i,:)...
        =   ShapeFun(Xi(i,:));
end
% find the shape function of each node

figure(1);
scatter3(Xi(:,1),Xi(:,2),Xi(:,3),20,N(:, 1),'filled')
view(31,38);
colorbar;
figure(2);
scatter3(Xi(:,1),Xi(:,2),Xi(:,3),20,N(:, 2),'filled')
view(-118,43);
colorbar;
figure(3);
scatter3(Xi(:,1),Xi(:,2),Xi(:,3),20,N(:, 3),'filled')
view(59,35);
colorbar;
figure(4);
scatter3(Xi(:,1),Xi(:,2),Xi(:,3),20,N(:, 4),'filled')
view(-130,30);
colorbar;
% Draw the image of the shape function in the parameter function space

% Contributed by OuYang
```



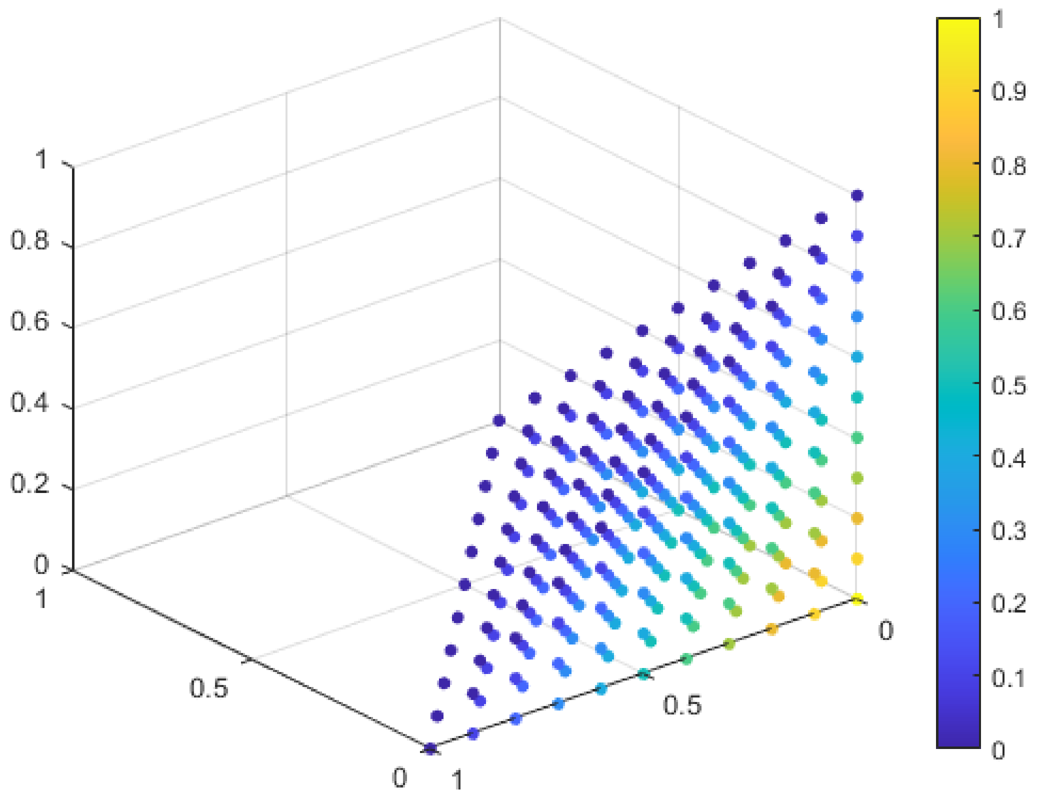
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0



1
0.5
0
1
0.5
0
1
0.8
0.6
0.4
0.2
0
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0



1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

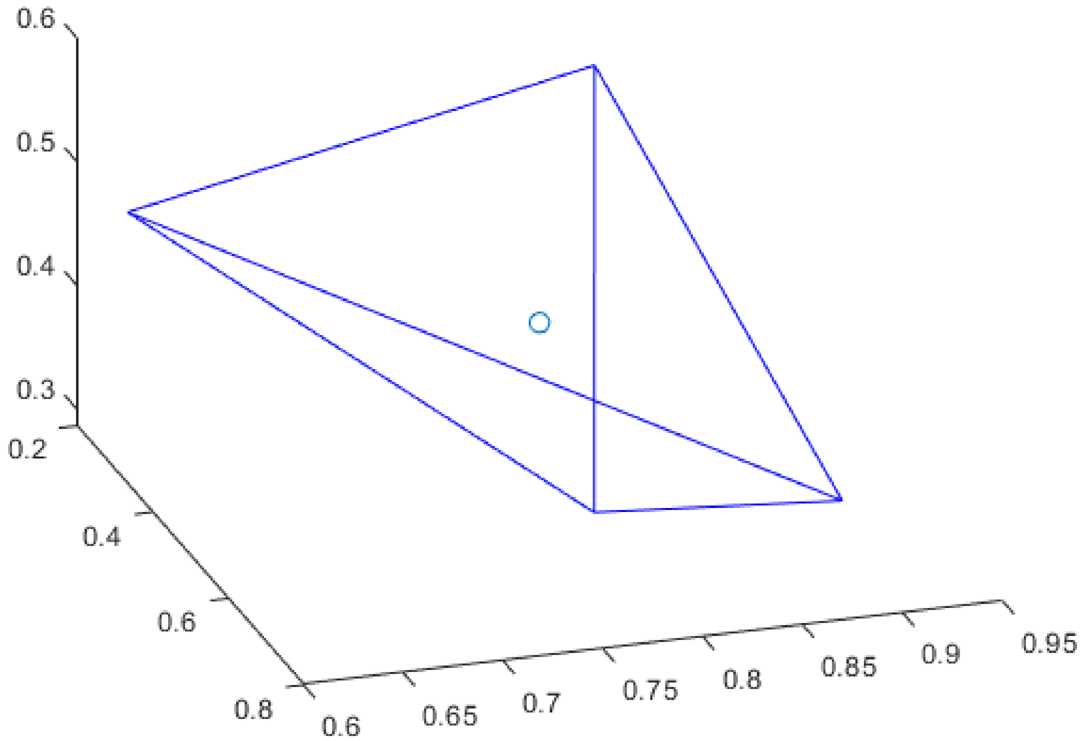


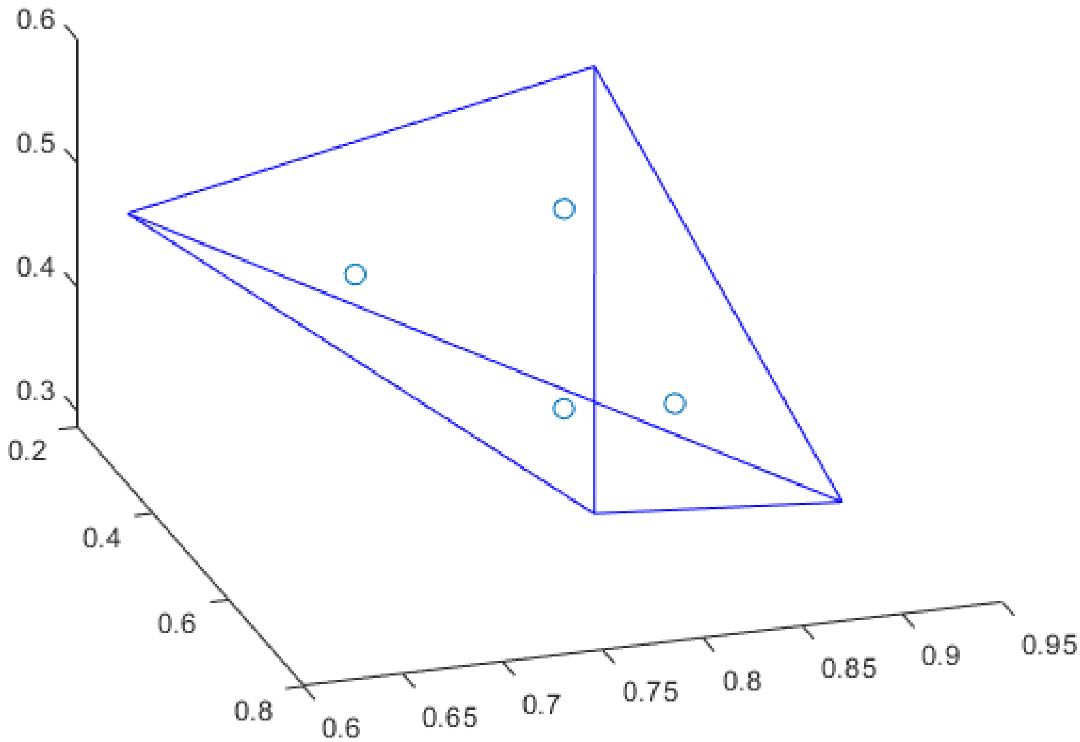
1
0.8
0.6
0.4
0.2
0
1
0.5
0
1
0.5
0
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

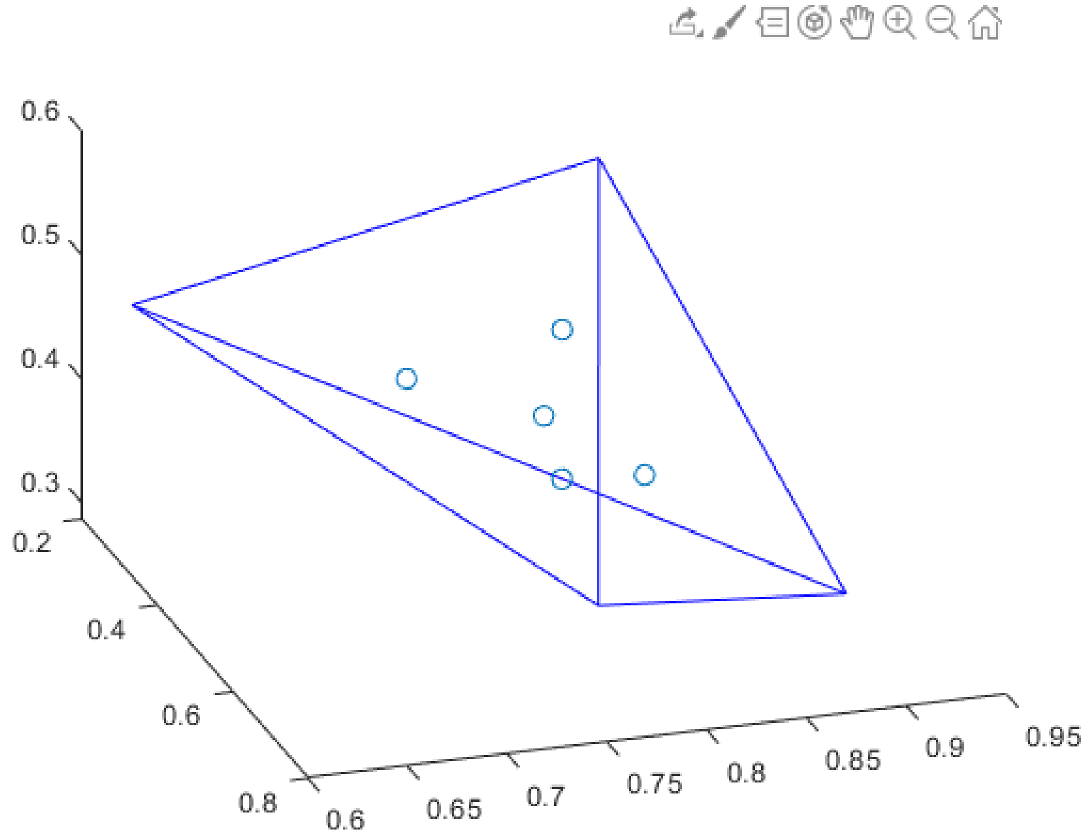
```
function [N,dN] = ShapeFun(Xi)
% Shape function of tetrahedron
% Syntax: [N,dN] = ShapeFun(Xi)

%       N : Shape Function N=[N1,N2,N3,N4]
%   dN: dN(i,j)=dN(i)/dXi(j)

N  = [Xi(1),Xi(2),Xi(3),1-Xi(1)-Xi(2)-Xi(3)];
% Shape function
dN = [1 0 0;
      0 1 0;
      0 0 1;
     -1 -1 -1];
% Divergence of all component of shape function in natural coordinates
end
% Contributed by OuYang
```

Published with MATLAB® R2019b

```matlab
clear;
close all;
clc;
% Draw the tetrahedron and Gauss point in real space

nint = 1:3;
% Order

PraCor = [0.2511,0.3517,0.5497,0.7572;
        0.6160,0.8308,0.9172,0.7537;
        0.4733,0.5853,0.2858,0.3804];
% Draw a specific tetrahedron

[D,nnde] = size(PraCor);
% Find the dimensionality and node's number in input data

ix   =[1 2 3 4 1 3 4 2];
% Tetrahedral connection order

 for k=1:length(nint)
    figure;
    patch('vertices', PraCor', 'faces', ix, 'facecolor', 'none', 'edgecolor', 'b')
    % Draw the tetrahedron in isoparametric space
    hold on

    [g, w] = TET4_GP(k);
    ngp    = size(g,1);
    % Calculate the coordinates (g) and number (ngp) of gauss points
    GusCor = zeros(ngp,D);
    % Define gauss point matrix in real space

    for i = 1:ngp
        [N,~] = ShapeFun(g(i,:));
        % Calculate shape function value in the gauss point
        GusCor(i,:) = (sum(N.*PraCor,2))';
        % Interpolate nodes to find Gauss points in real space
    end

    scatter3(GusCor(:,1),GusCor(:,2),GusCor(:,3),50)
    view(72,35);
    % Draw the position of the Gauss point in the tetrahedron in real space
 end
 % % Contributed by OuYang, Xiong
```


0.6
0.5
0.4
0.3
0.2
0.4
0.6
0.8
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95


0.6
0.5
0.4
0.3
0.2
0.4
0.6
0.8
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95



Published with MATLAB® R2019b

```matlab
function CC = ElastTensor(E,nu)
% Elastic Tensor of 3D
% Syntax: CC = ElastTensor(E,nu)

%   E  : Young's modulus
%   nu : Poisson's ratio

%   CC : ElastTensor (Ce by Ce),Ce = 6 (3D)

mu = E / 2 / (1 + nu);
lm = E * nu / (1 + nu) / (1 - 2 * nu);
% Lame constants

CC = [...
     2 * mu + lm, lm, lm, 0, 0, 0;
     lm, 2 * mu + lm, lm, 0, 0, 0;
     lm, lm, 2 * mu + lm, 0, 0, 0;
     0, 0, 0, mu, 0, 0;
     0, 0, 0, 0, mu, 0;
     0, 0, 0, 0, 0, mu];
% Calculate elastic tensor
end
% Contributed by OuYang
```

Published with MATLAB® R2019b

```matlab
function [J, detJ] = ShapeFunJacob(dN, x)
% Jacobi matrix and det
% [J, detJ] = ShapeFunJacob(dN, x)

% dN  : dN(i, j) = dN(i) / dg(j)
% x   : nodal coords. (number of nodes by D)

% J   : dx/dg
% detJ: determinant of J

[D,nnde] = size(x);
% Find the dimensionality and node's number in input data

J = zeros(D, D);
% Define jacobi matrix

for i = 1:nnde
    J = J + x(:,i)*dN(i, :);
end
% Calculate jacobi matrix
detJ  = det(J);
% Calculate det of jacobi matrix
end
% Contributed by Xiong
```