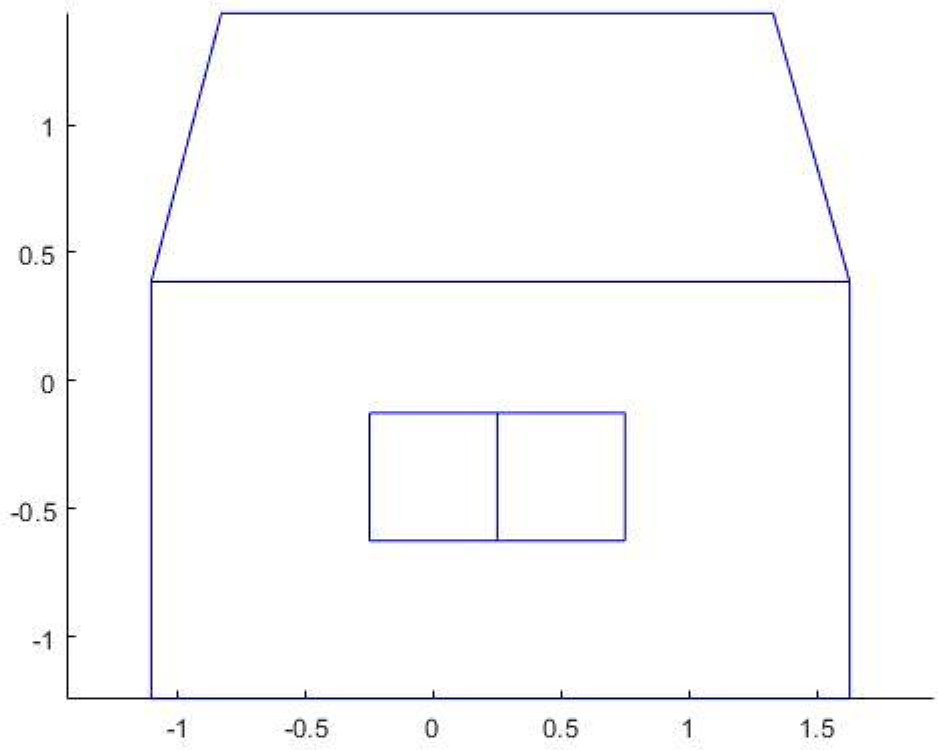
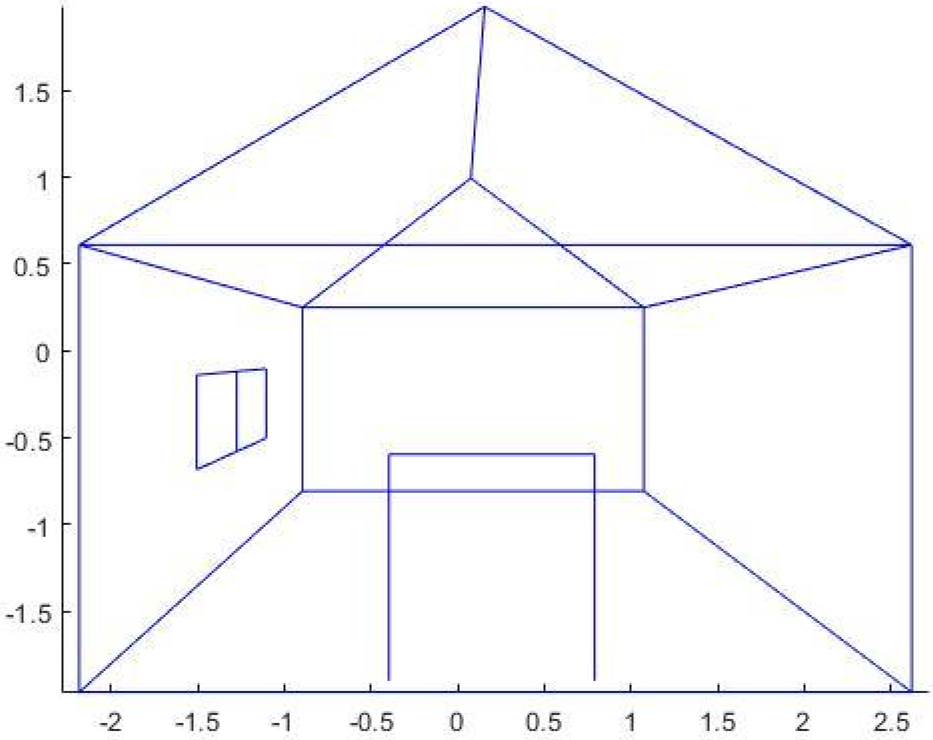
1. A right projection

Center of Projection (xyz):[0 0 10000]
z0: 0
Scaling amounts [sx sy sz]:[1 1 1]
Rotation angles [x y z]:[0 90 180]



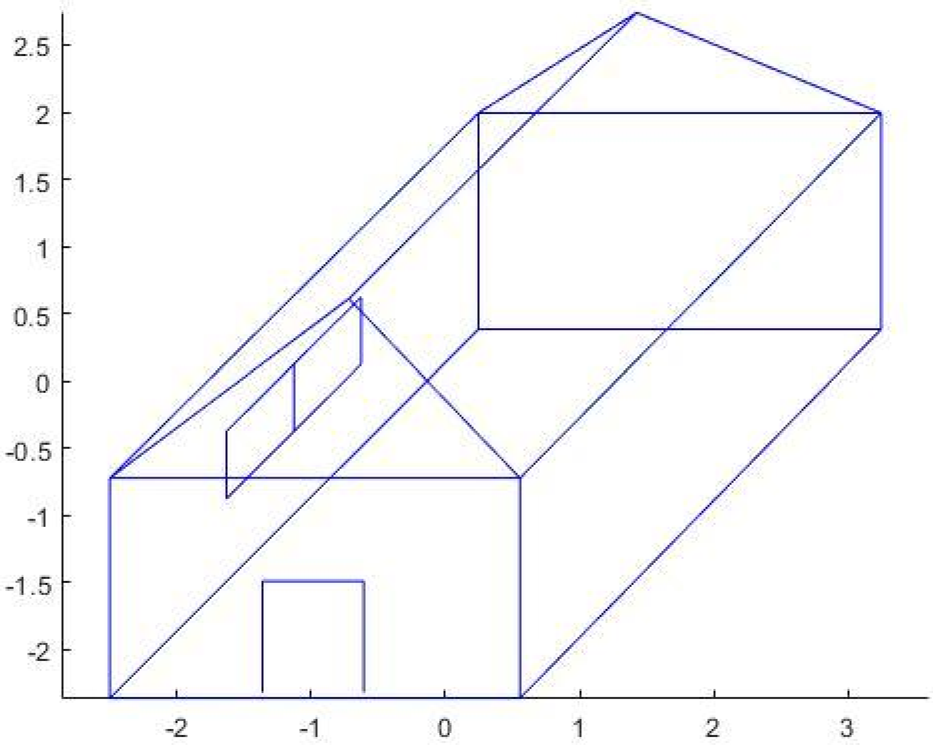
2. A perspective projection with one vanishing point (use the camera/eye arrangement)

Center of Projection (xyz):[0 0 3]
z0: 0
Scaling amounts [sx sy sz]: [1 1 1]
Rotation angles [x y z]:[0 0 180]



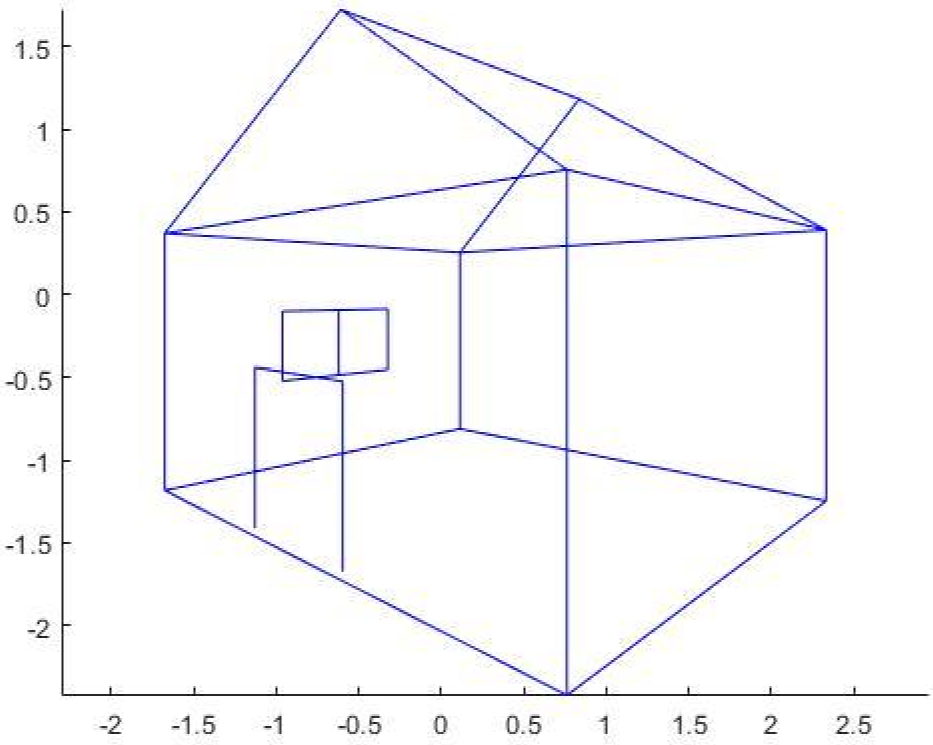
3. An oblique projection

Center of Projection (xyz):[150 150 150]
z0: 0
Scaling amounts [sx sy sz]:[1 1 1]
Rotation angles [x y z]:[0 0 180]



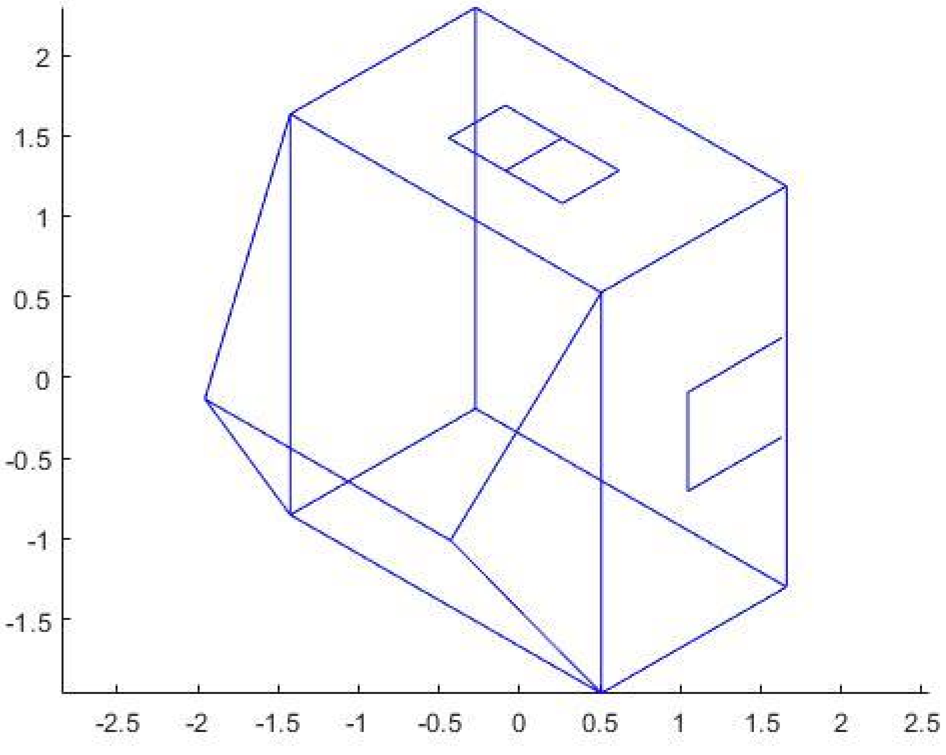
4. A perspective projection with two vanishing points (use the camera/eye arrangement)

Center of Projection (xyz):[0 0 4]
z0: 0
Scaling amounts [sx sy sz]:[1 1 1]
Rotation angles [x y z]:[0 45 180]



5. An isometric projection

Center of Projection (xyz):[0 0 1e6]
z0: 0
Scaling amounts [sx sy sz]:[1 1 1]
Rotation angles [x y z]:[135 35 90]



```
% homogenousCubedemo.m was created on 22 Feb 2020
% and last modified on 22 Feb 2020 by R. Adrezin
% This is a sample cube that calls cube.txt, point.txt and cubeEdges.txt
% It only creates a rotation about the z-axis
% This is a simplified version of the full code created by Prof. Marshall
% Long

%%%%%% load data
%first the vertices
clear
clc
close all;
load house.txt
% house=house;
[npts,~] = size(house);

% load the point around which the object is centered
load point.txt
% get the edge information
load Edges.txt
houseEdges=Edges;
[nedges,~] = size(houseEdges);
% [angx angy angz]=input("Enter the angle to rotate about z in degrees: ");
% for a=1:3

%% Input
cop = input("Center of Projection (xyz):");
z0 = input("z0: ");
scale = input("Scaling amounts [sx sy sz]:");
ang = input("Rotation angles [x y z]:");
o = ones(npts, 1);
house = [house o];

%% translation
dx=-point(1);
dy=-point(2);
dz=-point(3);

% T=[1 0 0 dx; 0 1 0 dy; 0 0 1 dz; 0 0 0 1];
T=[1 0 0 0; 0 1 0 0; 0 0 1 0; dx dy dz 1];
% translation_mat_inv=[1 0 0 0; 0 1 0 0; 0 0 1 0; -dx -dy -dz 1];
% house=house*T

%% scaling
sx=scale(1);
sy=scale(2);
sz=scale(3);
S=eye(4).*[sx sy sz 1];
% scaling_mat_inv=eye(4).*[1/sx 1/sy 1/sz 1];
% house=house*S;

%% rotation around x axis
angx=ang(1);
Rx = [1, 0, 0, 0; 0, cosd(angx),sind(angx), 0; 0 -sind(angx) cosd(angx) 0; 0 0 0 1];
%% rotation around y axis
```

```
angy=ang(2);
Ry = [cosd(angy), 0,-sind(angy), 0; 0 1 0 0; sind(angy), 0, cosd(angy), 0; 0 0 0 1];
%% rotation around z axis
angz=ang(3);
Rz = [cosd(angz), sind(angz),0, 0;-sind(angz), cosd(angz), 0,0;0 0 1 0; 0 0 0 1];

% housenew = house*[cosd(angz), sind(angz),0, 0;-sind(angz), cosd(angz), 0,0;0 0 1 0; 0 0 0 1];


%% rotation center
% R_center=[0 1 2 1];

%% rotation axis
% rx=00; ry=0; rz=90;
% R_center=[cosd(rx) cosd(ry) cosd(rz) 1];

Ttotal=T*S*Rx*Ry*Rz;
housenew=house*Ttotal;
% housenew=housenew*S;
%%%%%% Plot the result
% newplot

%% Projection
houseproject = housenew;

for n=1:npts
    houseproject(n,1:2) = housenew(n,1:2)+ ((z0 - housenew(n,3))/(housenew(n,3)-cop(3)))*(housenew(n,1:2)- cop(1:2));
end

%% Ploting Results (new house)
newplot
hold on
for n= 1:nedges
    plot(houseproject(houseEdges(n,:),1),houseproject(houseEdges(n,:),2),'b-');
%     hold on;
%     plot(houseold(houseEdges(n,:),1),houseold(houseEdges(n,:),2),'r-')
end
axis equal
hold off


%% this is the original house
% figure (2)
% hold on
% for n= 1:nedges
% %     plot(housenew(houseEdges(n,:),1),housenew(houseEdges(n,:),2),'b-');
% %     hold on;
%     plot(houseold(houseEdges(n,:),1),houseold(houseEdges(n,:),2),'r-')
% end
% axis equal




% hold off
```