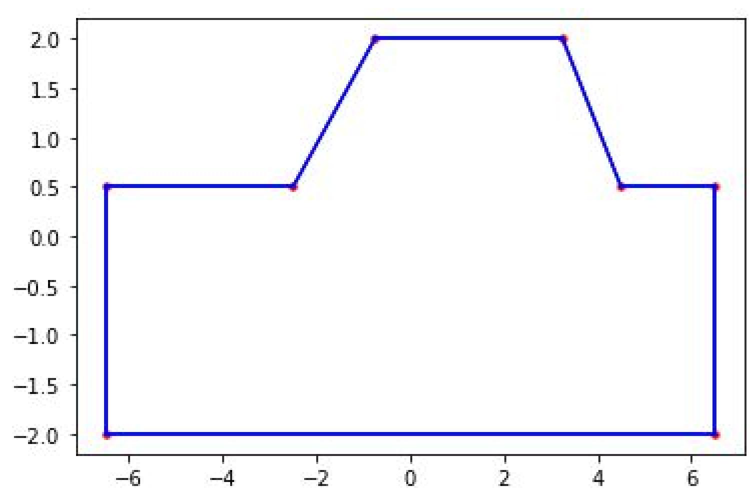
1. To find the perspective projection I started by creating an np array of the 16 by 4 matrix of the toyota as well as the 16 by 16 adjacency matrix which is used to determine which points are connected with lines when the projection is later graphed

```
N=np.array([[-6.5,-6.5,-6.5,-6.5,-2.5,-2.5,-0.75,-0.75,3.25,3.25,4.5,4.5,6.5,6.5,6.5,6.5],[-2,-2,0.5,0.5,0.5,0.5,2,2,2,2,0.5,0.5,0.5,0.5,-2,-2],[-2.5,2.5,2.5,-2.5,-2.5,2.5,-2.5,2.5,-2.5,2.5,-2.5,2.5,-2.5,2.5,2.5,-2.5],[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]])
adj=np.array([[0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,1],[1,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0],[0,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0],[1,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,0],[0,0,1,0,1,0,0,1,0,0,0,0,0,0,0,0],[0,0,0,0,1,0,0,1,1,0,0,0,0,0,0,0],[0,0,0,0,0,1,1,0,0,1,0,0,0,0,0,0],[0,0,0,0,0,0,1,0,0,1,1,0,0,0,0,0],[0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,0],[0,0,0,0,0,0,0,0,1,0,0,1,1,0,0,0],[0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0],[0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1],[0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,0],[0,1,0,0,0,0,0,0,0,0,0,0,1,0,1],[1,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0]])
f, ax1=plt.subplots(1)
ax1.plot(N[0,:],N[1,:],'r.')
for i in range(16):
        for j in range(i):
        if adj[i,j]==1:
        ax1.plot([N[0,i],N[0,j]],[N[1,i],N[1,j]],'b')
```

The graph which is produced appears to have no depth but that is merely because we are looking at this image from a perspective where depth is not projected.



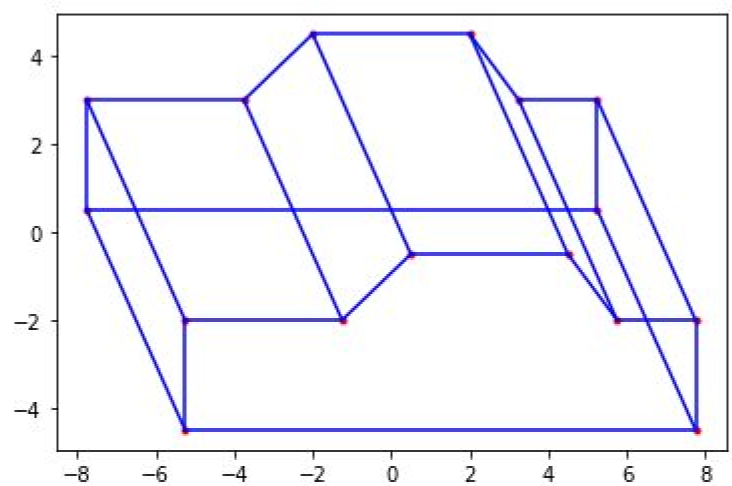
I evaluated perspective projection matrices for the given centers of projection and performed matrix multiplication with the original toyota matrix N to view depth

(a) (b,c,d) = (-5,10,10)

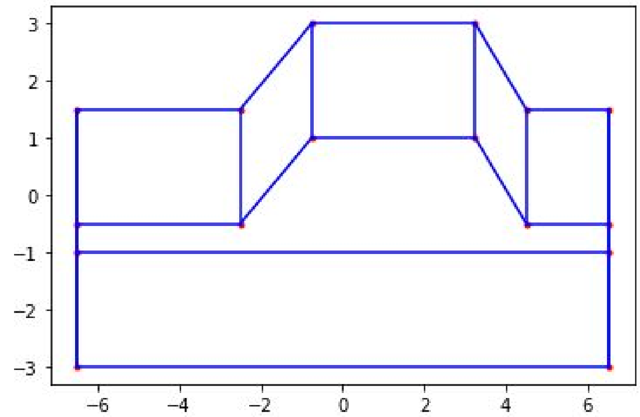
(b) (b,c,d) = (0,10,25)

$$\begin{bmatrix} 1 & 0 & -\frac{b}{d} & 0 \\ 0 & 1 & -\frac{c}{d} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{d} & 1 \end{bmatrix}$$

```
#Perspective projection (a)
aP = np.array([[1,0,0.5,0],[0,1,-1,0],[0,0,0,0],[0,0,-0.1,1]])
aN = np.matmul(aP,N)
f, ax1=plt.subplots(1)
ax1.plot(aN[0,:],aN[1,:],'r.')
for i in range(16):
        for j in range(i):
        if adj[i,j]==1:

ax1.plot([aN[0,i],aN[0,j]],[aN[1,i],aN[1,j]],'b')
```



```
#Perspective projection (b)

bP = np.array([[1,0,0,0],[0,1,-0.4,0],[0,0,0,0],[0,0,-0.04,1]])

bN = np.matmul(bP,N)

f, ax2=plt.subplots(1)

ax2.plot(bN[0,:],bN[1,:],'r.')

for i in range(16):

    for j in range(i):

    if adj[i,j]==1:

    ax2.plot([bN[0,i],bN[0,j]],[bN[1,i],bN[1,j]],'b')
```



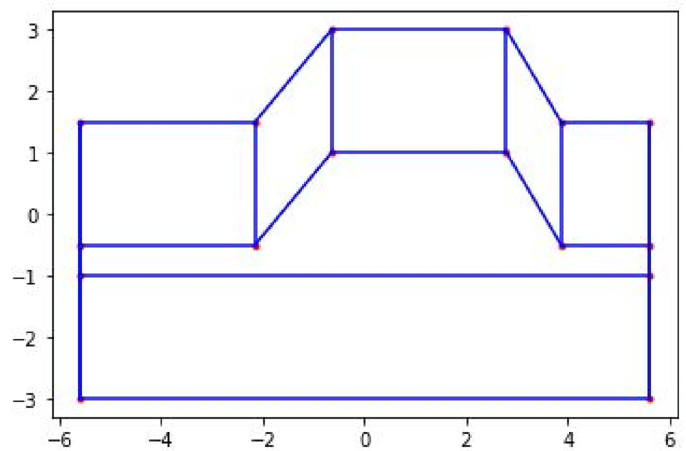
In projection (b) only a vertical perspective is applied so we get a top down view of the vehicle

2. I used the same bN matrix to evaluate the perspective of the matrix after being rotated 30 degrees over the y axis since bN is already in the center of projection (0,10,25) . This time I multiplied bN by Ay:

Theta is replaced by 30(degrees)

$$A_y = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
#Rotation 30 degrees about the y axis
A = np.array([[.86,0,.5,0],[0,1,0,0],[-.5,0,.86,0],[0,0,0,1]]) #this is our Ay matrix
N1 = np.matmul(A,bN)
f, ax3=plt.subplots(1)
ax3.plot(N1[0,:],N1[1,:],'r.')
for i in range(16):
    for j in range(i):
    if adj[i,j]==1:
    ax3.plot([N1[0,i],N1[0,j]],[N1[1,i],N1[1,j]],'b')
```

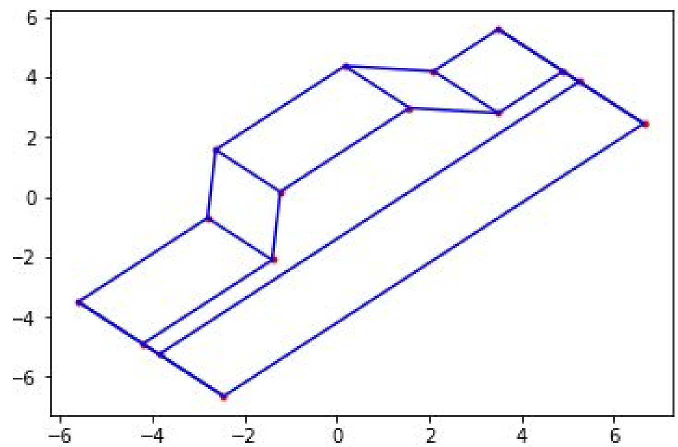


Not much about our graph changes visually but by assessing the numerical information given by the axes of the graph we can see the coordinates have changed with respect to a slight rotation around the y axis.

3. A similar process is done to find rotation along the z axis but this time multiplying bN by Az:

$$A_z = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
#Rotation 45 degrees about the z axis
B = np.array([[0.7,-0.7,0,0],[0.7,0.7,0,0],[0, 0,1,0],[0,0,0,1]]) #this is our Az matrix
N2 = np.matmul(B,bN)
f, ax4=plt.subplots(1)
ax4.plot(N2[0,:],N2[1,:],'r.')
for i in range(16):
    for j in range(i):
    if adj[i,j]==1:
    ax4.plot([N2[0,i],N2[0,j]],[N2[1,i],N2[1,j]],'b')
```



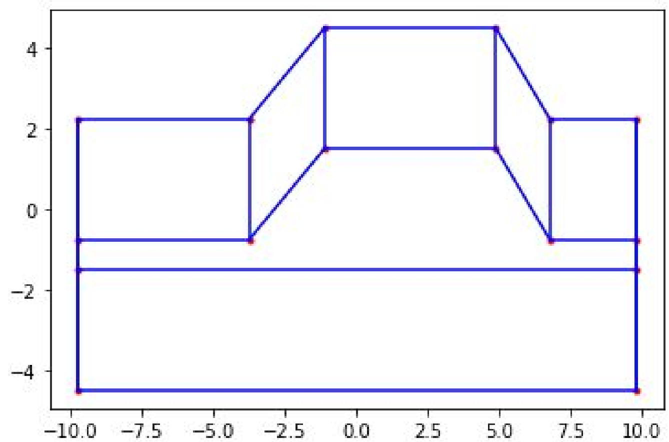
4. To zoom in on the toyota we put the factor we are zooming in on the toyota in (p) along the diagonal of the following matrix

$$A = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since we are zooming 150% p = 1.5

This matrix is then multiplied by bN

```
#Zoom by 150%
S = np.array([[1.5,0,0,0],[0,1.5,0,0],[0,0,1.5,0],[0,0,0,1]])
N3 = np.matmul(S,bN)
f, ax5 = plt.subplots(1)
ax5.plot(N3[0,:],N3[1,:],'r.')
for i in range(16):
        for j in range(i):
        if adj[i,j]==1:
        ax5.plot([N3[0,i],N3[0,j]],[N3[1,i],N3[1,j]],'b')
```



Once again, looking at the numeric data in the axes reveals that the vehicle has been expanded comparative to the graph of our vehicle in question 1 projection (b)