1. Calculate the inertial matrix, J^b, for a brick with dimensions 5, 10 and 20 cm (along the x-, y- and z-axis) when the density is $2\frac{kg}{dm^3}$. Place the b-frame in the center of mass with axis parallell to the edges.

The unit of brick dimensions and density is adjusted as a standard unit [m] and $[\frac{kg}{m^3}]$.
Let suppose the dimension is described as below with the variables respectively,

$$a = 5[cm] = 0.05[m]$$

$$b = 10[cm] = 0.1[m]$$

$$c = 20[cm] = 0.2[m]$$

and the density

$$d = 2[\frac{kg}{dm^3}] = 2000[\frac{kg}{m^3}]$$

The inertial matrix, J^b can be described as below,

$$J^b = \begin{bmatrix} J^b_{xx} & -J^b_{xy} & J^b_{xz} \\ -J^b_{yx} & J^b_{yy} & -J^b_{yz} \\ -J^b_{zx} & -J^b_{zy} & J^b_{zz} \end{bmatrix}$$

As we place the b-frame in the center of mass with axis parallell to the edges and according to the dimension a, b and c written above, it is obvious that $a < b < c$. In that case, we can describe that the inertia matrix J^b is real, symmetric and positive definite matrix (Det($J^b > 0$)) meaning that the product of inertia equals zero and the momentia of inertia matrix will be diagonal. Therefore the inertia matrix will be as below.

$$J^b = \begin{bmatrix} J^b_{xx} & 0 & 0 \\ 0 & J^b_{yy} & 0 \\ 0 & 0 & J^b_{zz} \end{bmatrix}$$

where
$J^b_{xx} = \frac{1}{12}M(b^2 + c^2)$
$J^b_{xx} = \frac{1}{12}M(a^2 + c^2)$
$J^b_{xx} = \frac{1}{12}M(a^2 + b^2)$
where M = a*b*c*d

Place the b-frame in the center of mass with axis parallel to the edges.

2. Define and find the kinetic and angular momentum ellipsoids. Draw the ellipsoids with a common center (3D, opaque and with two different colours) for three cases which give intersection lines near the three axis (the kinetic energy ellipsoid is constant). See the attachment for how to choose the three cases.

By using the guidelines which are given at the task attachment, the kinetic energy and angular momentum ellipsoids have been defined and find.

The listing of the programs used in the figures are described as follows.

```
%Kinetic energy ellipsoids

%b3 as main-axis
w3_tilde_b3 = 2*pi;
K0 = 0.5*Jzz*w3_tilde_b3^2;
% angular velocity near b1 (or between b1 and b3)
w2_b3 = 0;
w3_b3 = w3_tilde_b3/10;
w1_tilde_b3 = sqrt(2*K0/Jxx);
w1_b3 = w1_tilde_b3*sqrt(1-(w3_b3^2)/(w3_tilde_b3^2));
```

```
%b2 as main-axis
w2_tilde_b2 = 2*pi;
% angular velocity near b3 (or between b2 and b3)
w1_b2 = 0;
w2_b2 = w2_tilde_b2/10;
w3_tilde_b2 = sqrt(2*K0/Jzz);
w3_b2 = w3_tilde_b2*sqrt(1-(w2_b2^2)/(w2_tilde_b2^2));
```

```
%b1 as main-axis
w1_tilde_b1 = 2*pi;
% angular velocity near b2 (or between b1 and b2)
w3_b1 = 0;
w1_b1 = w1_tilde_b3/10;
w2_tilde_b1 = sqrt(2*K0/Jyy);
w2_b1= w2_tilde_b1*sqrt(1-(w1_b1^2)/(w1_tilde_b1^2));
```

Then the Kinetic energy ellipsoid can be defined as below.

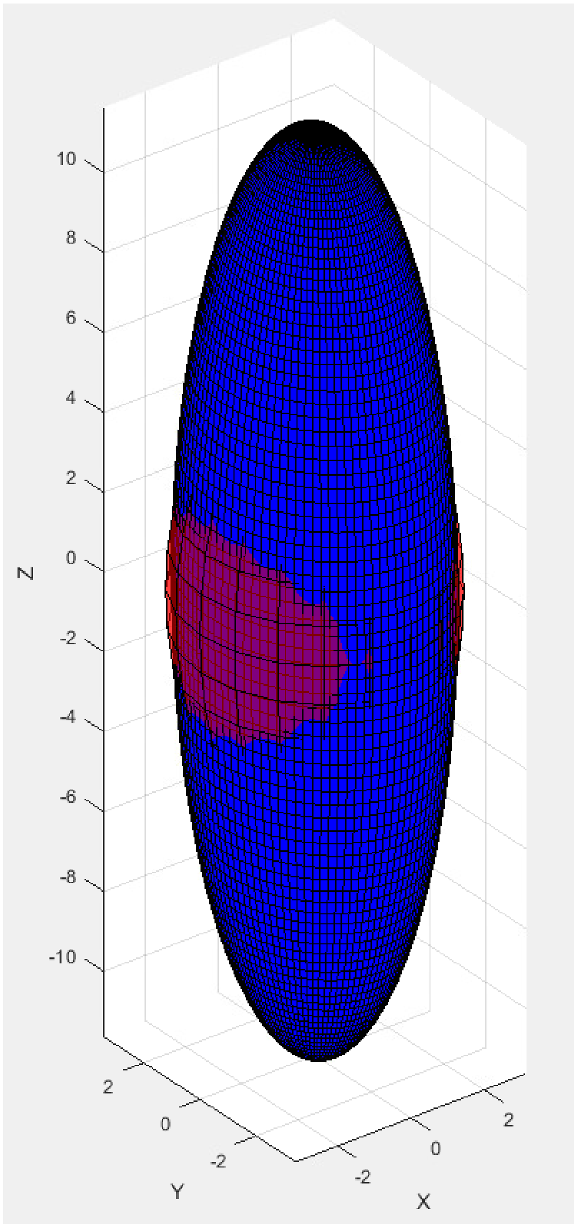
```
%Kinetic energy ellipsoid
K0_b1 = sqrt(2*K0/Jxx);
K0_b2 = sqrt(2*K0/Jyy);
K0_b3 = sqrt(2*K0/Jzz);
```

After that, three different kinds of angular momentum ellipsoids from three different axes can be found as follow.

```
%3 different angular momentum ellipsoid from 3 different axes
H0_b1 = sqrt(Jxx^2*w1_b1^2 + Jyy^2*w2_b1^2 + Jzz^2*w3_b1^2);
H0_b2 = sqrt(Jxx^2*w1_b2^2 + Jyy^2*w2_b2^2 + Jzz^2*w3_b2^2);
H0_b3 = sqrt(Jxx^2*w1_b3^2 + Jyy^2*w2_b3^2 + Jzz^2*w3_b3^2);
```

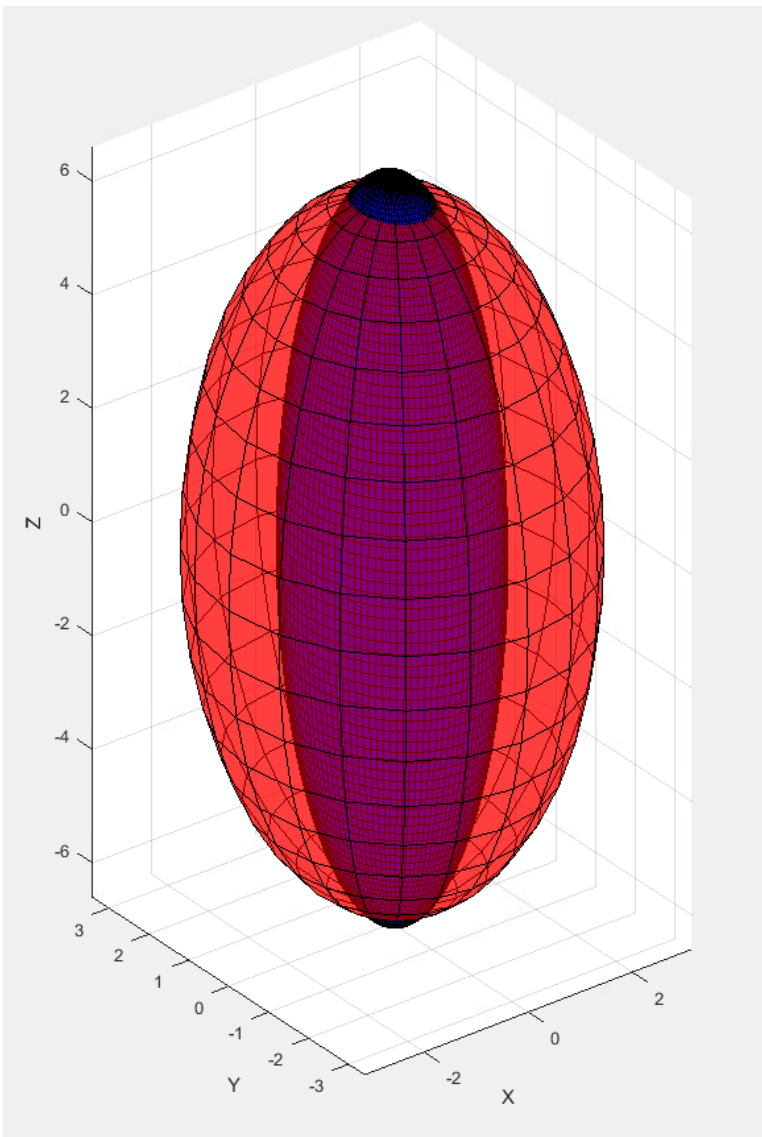
Rotating at b1 axes

```
%Rotating at b1 axes
figure(1)
[x,y,z] = ellipsoid(0,0,0,K0_b1,K0_b2,K0_b3);
surf(x,y,z,'FaceColor','r','FaceAlpha',0.5)
hold on
[x,y,z] = ellipsoid(0,0,0,H0_b1/Jxx, H0_b1/Jyy, H0_b1/ Jzz, 100);
surf(x,y,z,'FaceColor','b')
axis equal
xlabel('X')
ylabel('Y')
zlabel('Z')
hold off
```



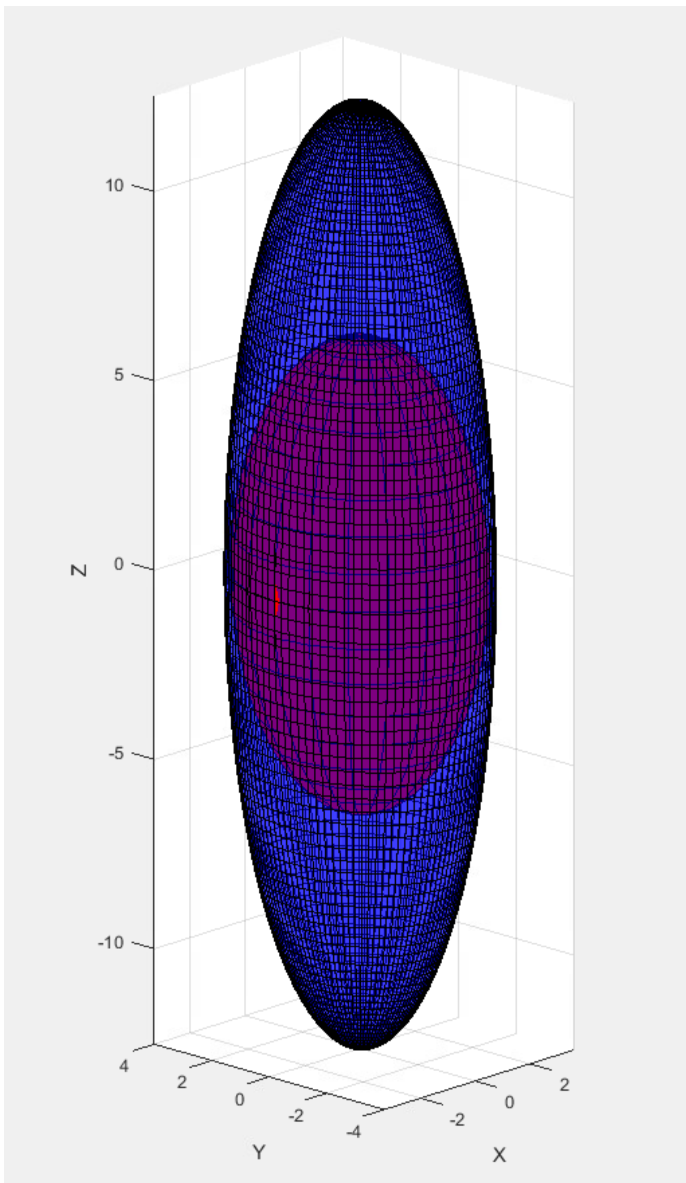
Rotating at b2 axes

```
%Rotating at b2 axes
figure(2)
[x,y,z] = ellipsoid(0,0,0,K0_b1,K0_b2,K0_b3);
surf(x,y,z,'FaceColor','r','FaceAlpha',0.5)
hold on
[x,y,z] = ellipsoid(0,0,0,H0_b2/Jxx, H0_b2/Jyy, H0_b2/ Jzz, 100);
surf(x,y,z,'FaceColor','b')
axis equal
xlabel('X')
ylabel('Y')
zlabel('Z')
hold off
```



Rotating at b3 axes

```
%Rotating at b3 axes
figure(3)
[x,y,z] = ellipsoid(0,0,0,K0_b1,K0_b2,K0_b3);
surf(x,y,z,'FaceColor','r')
hold on
[x,y,z] = ellipsoid(0,0,0,H0_b3/Jxx, H0_b3/Jyy, H0_b3/ Jzz, 100);
surf(x,y,z,'FaceColor','b','FaceAlpha',0.5)
axis equal
xlabel('X')
ylabel('Y')
zlabel('Z')
hold off
```



3. Solve the Euler equations numerically for the same three cases (with initial values on the intersection lines). Draw the solutions in the same figures as above.

We can also use Euler equations and integrate it in MatLab program for the three cases with the same initial values as in question.2, as it also gives the same solutions.

Euler equation can be used to find the motion
($\underline{\omega}_b^{ib}$) as it is given the force ($\underline{n}_c^b$),
$\dot{\omega}_x = \frac{1}{J_{xx}}[(J_{yy} - Jzz)\omega_y\omega_z + n_x]$
$\dot{\omega}_y = \frac{1}{J_{yy}}[(J_{zz} - Jxx)\omega_x\omega_z + n_y]$
$\dot{\omega}_z = \frac{1}{J_{zz}}[(J_{xx} - Jyy)\omega_x\omega_y + n_z]$
where $\underline{n}_c^b = 0$ or torque free motion in this case.

The ode45 function in MatLab is a function for solving numerical differential equation where we can draw the ellipsoid trajectory with a common center for three cases which gives the intersection nears the main axes. After using the Eular equation and plotting the result in MatLab, we can see that the trajectory as a result as describe at the image below.

```
%% Question 3

%b1 axes

syms b1 b11 b2 b22 b3 b33
syms t1 t11 t2 t22 t3 t33

t0 = 0:0.01:10;


%b1 axes
omgb1 = @(t1, b1)  [(Jyy-Jzz)*b1(2)*b1(3)/Jxx;...
                    (Jzz-Jxx)*b1(3)*b1(1)/Jyy;...
                    (Jxx-Jyy)*b1(1)*b1(2)/Jzz];
[t1, b1] = ode45(omgb1,t0,[w1(1), w1(2), w1(3)]);

omgb11 = @(t11, b11)  [(Jyy-Jzz)*b11(2)*b11(3)/Jxx;...
                      (Jzz-Jxx)*b11(3)*b11(1)/Jyy;...
                      (Jxx-Jyy)*b11(1)*b11(2)/Jzz];
[t11, b11] = ode45(omgb11,t0,[-2.8745, 0.42722, -2.2108]);
```

```
%b2 axes
omgb2 = @(t2, b2) [(Jyy-Jzz)*b2(2)*b2(3)/Jxx;...
                   (Jzz-Jxx)*b2(3)*b2(1)/Jyy;...
                   (Jxx-Jyy)*b2(1)*b2(2)/Jzz];
[t2, b2] = ode45(omgb2,t0,[w2(1), w2(2), w2(3)]);

omgb22 = @(t22, b22) [(Jyy-Jzz)*b22(2)*b22(3)/Jxx;...
                      (Jzz-Jxx)*b22(3)*b22(1)/Jyy;...
                      (Jxx-Jyy)*b22(1)*b22(2)/Jzz];
[t22, b22] = ode45(omgb22,t0,[-0.53281, 0.3446, -6.1427]);
```

```
%b3 axes
omgb3 = @(t3, b3) [(Jyy-Jzz)*b3(2)*b3(3)/Jxx;...
                   (Jzz-Jxx)*b3(3)*b3(1)/Jyy;...
                   (Jxx-Jyy)*b3(1)*b3(2)/Jzz];
[t3, b3] = ode45(omgb3,t0,[w3(1), w3(2), w3(3)]);

omgb33 = @(t33, b33) [(Jyy-Jzz)*b33(2)*b33(3)/Jxx;...
                      (Jzz-Jxx)*b33(3)*b33(1)/Jyy;...
                      (Jxx-Jyy)*b33(1)*b33(2)/Jzz];
[t33, b33] = ode45(omgb33,t0,[3.1298, 0.2312, 0]);
```

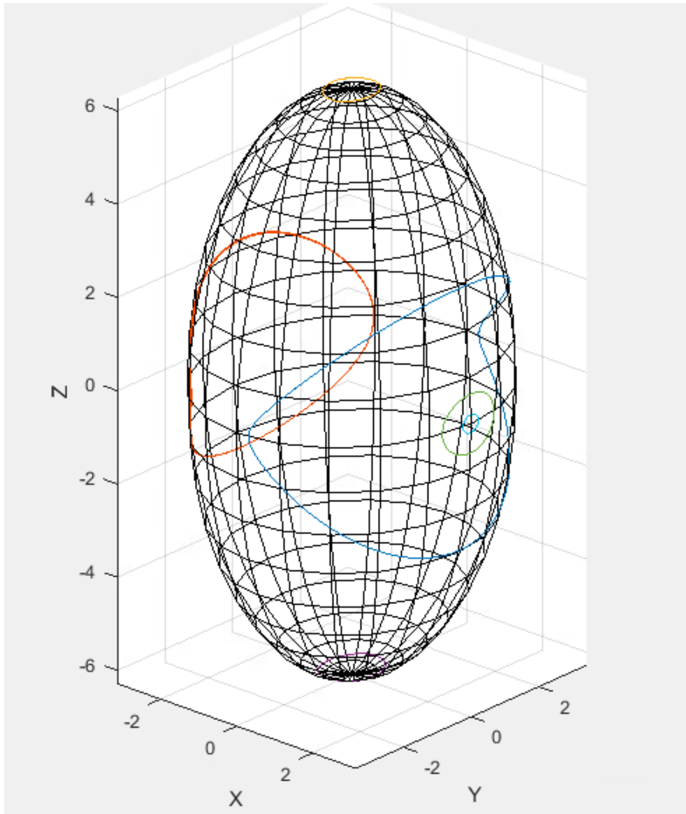
```
figure(4)
[x,y,z] = ellipsoid(0,0,0,K0_b1,K0_b2,K0_b3);
surf(x,y,z,'FaceColor','r','FaceAlpha',0)
hold on

plot3(b1(:,1),b1(:,2),b1(:,3))
grid on
plot3(b11(:,1),b11(:,2),b11(:,3))
grid on

plot3(b2(:,1),b2(:,2),b2(:,3))
grid on
plot3(b22(:,1),b22(:,2),b22(:,3))
grid on

plot3(b3(:,1),b3(:,2),b3(:,3))
grid on
plot3(b33(:,1),b33(:,2),b33(:,3))
grid on

axis equal
xlabel('X')
ylabel('Y')
zlabel('Z')
hold off
```



6
4
2
Z 0
-2
-4
-6
-2
0
2
X
-2
0
2
Y

4. Write the kinematic equations for 3-2-1 Euler angles and solve the kinematic and angular momentum equations numerically for the three cases above. Make three animations showing how the brick rotates as seen from inertial space.

The kinematic equation for 3-2-1 Euler angels can be written as below,
Euler equation can be used to find the motion
$(\underline{\omega}_b^{ib})$ as it is given the force $(\underline{n}_c^b)$,
$\dot{\omega}_x = \frac{1}{J_{xx}}[(J_{yy} - Jzz)\omega_y\omega_z + n_x]$
$\dot{\omega}_y = \frac{1}{J_{yy}}[(J_{zz} - Jxx)\omega_x\omega_z + n_y]$
$\dot{\omega}_z = \frac{1}{J_{zz}}[(J_{xx} - Jyy)\omega_x\omega_y + n_z]$
where $\underline{n}_c^b = 0$ or torque free motion in this case.

By using the equation and solving for the kinematic and angular momuntum equations numericall for the three cases above using MatLab, we can create animations showing how the brick rotates as seen from inertial space. The simulation file has been attach with the report file and shown the result.