

```

function plotPolhode(I, w_0, out)

I1 = I(1, 1);
I2 = I(2, 2);
I3 = I(3, 3);

T = (I1*w_0(1)^2 + I2*w_0(2)^2 + I3*w_0(3)^2)*0.5;
h = (I1^2*w_0(1)^2 + I2^2*w_0(2)^2 + I3^2*w_0(3)^2)^0.5;
Ngrid = 500;

w1em = sqrt(2*T/I1); w2em = sqrt(2*T/I2); w3em = sqrt(2*T/I3);
[w1e,w2e,w3e]=ellipsoid(0,0,0,w1em,w2em,w3em,Ngrid);

w1hm = h/I1; w2hm = h/I2; w3hm = h/I3;
[w1m,w2m,w3m]=ellipsoid(0,0,0,w1hm,w2hm,w3hm,Ngrid);

figure
surf(w1e,w2e,w3e,'FaceColor','r','EdgeColor','none','FaceAlpha',.5)
x = 20;
axis([-x x -x x -x x]);
hold on;

% plot momentum ellipsoid
surf(w1m,w2m,w3m,'FaceColor','b','EdgeColor','none','FaceAlpha',.3)

% plot principal axis vectors
plot3([0 x + 0.5],[0 0],[0 0],'k<')
plot3([0 x + 0.5],[0 0],[0 0],'k-')
plot3([0 0],[0 x + 0.5],[0 0],'k-')
plot3([0 0],[0 x + 0.5],[0 0],'k>')
plot3([0 0],[0 0],[0 x + 0.5],'k-')
plot3([0 0],[0 0],[0 x + 0.5],'k^')

% set the view perspective
view(135,20)

% Initialize the vector as a quiver plot (arrows in 3D)
angular_velocity = out.omega;
num_steps = size(angular_velocity, 1);
h = quiver3(0, 0, 0, angular_velocity(1, 1), angular_velocity(1, 2), angular_velocity(1, 3), 'MaxHeadSize', 0.5, 'LineWidth', 2);

% Animation loop
for t = 2:num_steps
    % Update the vector with new data for this time step
    set(h, 'UData', angular_velocity(t, 1), 'VData', angular_velocity(t, 2), 'WData', angular_velocity(t, 3));

    % Pause to control the speed of the animation
    pause(out.tout(2)-out.tout(1)); % Adjust pause duration to change animation speed
end

```

```
hold off;
```

```
end
```