```matlab
classdef ElemWithAxes < matlab.System
    % ElemWithAxes:  An antenna element with a local frame of reference
    %
    % This class combines an antenna element from the phased array toolbox
    % along with a local coordinate system to facilitate geometric
    % computations.
    %
    % In addition, it provides smooth interpolation of the directivity
    % which is not performed in the phased array toolbox
    properties
        % The antenna object from the phased array toolbox
        ant = [];

        % Azimuth and elevation angle of the element peak directivity
        axesAz = 0;
        axesEl = 0;

        % Axes of the element local coordinate frame of reference
        axesLoc = eye(3);

        % Frequency in Hz
        fc = 28e9;
        vc = physconst('lightspeed');

        % Directivity interpolant
        dirInterp = [];

        % Velocity vector in 3D in m/s
        vel = zeros(1,3);
    end
    methods
        function obj = ElemWithAxes(fc, ant)
            % Constructor
            % Inputs:  fc is the carrier frequency in Hz and ant is
            % an antenna compatible with the phased array toolbox.  It must
            % support the ant.pattern() method.

            % TODO:  Assign fc and ant to the class variables
            % obj.fc and obj.ant
            obj.fc = fc;
            obj.ant = ant;
        end

        function alignAxes(obj,az,el)
            % Aligns the axes to given az and el angles

            % TODO:  Set the axesAz and axesEl to az and el
            obj.axesAz = az;
            obj.axesEl = el;

            % TODO:  Use the azelaxes() function to create a 3 x 3 array
            % corresponding to an orthonormal basis for the local
            % coordinate system of the array aligned in the direction
            % (az,el).  Save this in the axesLoc property.
            obj.axesLoc = azelaxes(obj.axesAz, obj.axesEl);
        end

        function dop = doppler(obj,az,el)
            % Computes the Doppler shift of a set of paths
```

```matlab
            % The angles of the paths are given as (az,el) pairs
            % in the global frame of reference.

            % TODO:  Use the sph2cart method to find unit vectors in the
            % direction of each path.  That is, create an array where
            % u(:,i) is a unit vector in the angle (az(i), el(i)).
            % Remember to convert from degrees to radians!
            [x,y,z] = sph2cart(deg2rad(az),deg2rad(el),ones(size(az)));
            u = [x; y; z]';
            % TODO:  Compute the Doppler shift of each path from the
            % velocity vector, obj.vel.  The Doppler shift of path i is
            %     dop(i) = vel*u(:,i)*fc/vc,
            % where vc = speed of light
            dop = u*obj.vel*(obj.fc/obj.vc);
        end

    end

    methods (Access = protected)
         function setupImpl(obj)
            % setup:  This is called before the first step.
            % We will use this point to interpolator

            % TODO:  Get the pattern from ant.pattern
            [dir,az,el] = obj.ant.pattern(obj.fc, 'Type', 'Directivity');

            % TODO:  Create the gridded interpolant object.  You can follow
            % the demo in the antennas lecture
            %     obj.dirInterp = griddedInterpolant(...)
            obj.dirInterp = griddedInterpolant({el,az},dir);

         end

        function dir = stepImpl(obj, az, el)
            % Computes the directivity along az and el angles
            % The angles are given in the global frame of reference
            % We do this by first rotating the angles into the local axes

            % TODO:  Use the global2localcoord function to translate
            % the gloabl angles (az(i), el(i)) into angles
            % (azLoc(i),elLoc(i)) in the local coordinate system.  use
            % the 'ss' option along with the local axes obj.axesLoc.
            locCoord = global2localcoord([az; el; ones(size(az));],'ss',[0;0;0],obj.axesLoc);
            azLoc = locCoord(1,:);
            elLoc = locCoord(2,:);
            % TODO:  Run the interplationn object to compute the directivity
            % in the local angles
            dir = obj.dirInterp(elLoc,azLoc);

        end

    end

end
```

```
Not enough input arguments.

Error in ElemWithAxes (line 40)
            obj.fc = fc;
```