ME 481 – H15 **Name** _____

1) Local coordinates for vertices on bodies 2 and 3 are provided below in CW closed boundary chains. Time samples for position and attitude of each body are provided as data in the attached MATLAB code. A sample MATLAB plot of motion for body 2 is shown below.

$$\{s_2\}^{'ALL} = \begin{bmatrix} \begin{Bmatrix} 2 \\ 3 \end{Bmatrix} & \begin{Bmatrix} 3 \\ -1 \end{Bmatrix} & \begin{Bmatrix} -1 \\ -2 \end{Bmatrix} & \begin{Bmatrix} -1 \\ 2 \end{Bmatrix} & \begin{Bmatrix} 2 \\ 3 \end{Bmatrix} \end{bmatrix} \qquad \{s_3\}^{'ALL} = \begin{bmatrix} \begin{Bmatrix} 2 \\ 3 \end{Bmatrix} & \begin{Bmatrix} 1 \\ -2 \end{Bmatrix} & \begin{Bmatrix} -1 \\ 0 \end{Bmatrix} & \begin{Bmatrix} 2 \\ 3 \end{Bmatrix} \end{bmatrix}$$

2) Run the MATLAB code and then modify it to additionally show motion of body 3 on the same MATLAB plot. Manually assess when collision occurs based on the plot and record in the table below. Attach hardcopy of your MATLAB plot and code.

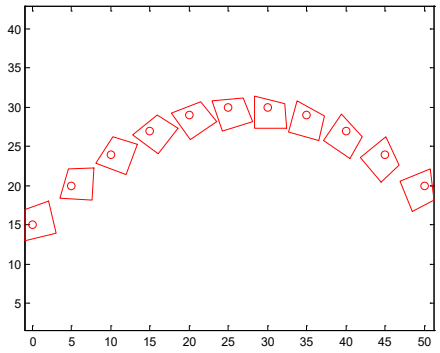
3) Use the bounding circles test to determine when the objects would be candidates for collision detection and record your results below. Attach copy of your code.

4) Use the point-in-polygon test with MATLAB "inpolygon" to detect collision and record your results below. Attach copy of your code.

5) Use the edge intersection test with MATLAB "polyxpoly" to detect collision and record your results below. For example, if edge 4 on body 2 intersects edge 1 on body 3, record $E4_2/E1_3$. Attach copy of your code.

Time [sec]	5	6	7	8	9	10
Manual C = collision blank = no						
Bounding circles C = candidates blank = no						
Point-in-polygon Which vertices on body 2 are inside polygon 3 ?						
Point-in-polygon Which vertices on body 3 are inside polygon 2 ?						
Edge intersection Which edges on 2 intersect which edges on 3 ?						

EXTRA CREDIT
Use Savitsky-Golay interpolants to compute the time for initial collision.




```
% show_body2.m - show motion for collision detection
% HJSIII, 19.04.24

clear

% motion data
%     time        x2        y2      phi2        x3        y3      phi3
%    [sec]      [in]      [in]     [rad]      [in]      [in]     [rad]
all = [  0         0   15.0000         0         0         0         0 ;
    1.0000    5.0000   20.0000   -0.3000    6.6000         0    0.2000 ;
    2.0000   10.0000   24.0000   -0.6000   13.0684    1.3112    0.4000 ;
    3.0000   15.0000   27.0000   -0.9000   19.1474    3.8814    0.6000 ;
    4.0000   20.0000   29.0000   -1.2000   24.5947    7.6080    0.8000 ;
    5.0000   25.0000   30.0000   -1.5000   29.1929   12.3426    1.0000 ;
    6.0000   30.0000   30.0000   -1.8000   32.7589   17.8963    1.2000 ;
    7.0000   35.0000   29.0000   -2.1000   35.1505   24.0477    1.0000 ;
    8.0000   40.0000   27.0000   -2.4000   38.7165   29.6014    0.8000 ;
    9.0000   45.0000   24.0000   -2.7000   43.3147   34.3360    0.6000 ;
   10.0000   50.0000   20.0000   -3.0000   48.7620   38.0626    0.4000 ];
time = all(:,1)';      % size 1 x nt
r2_all = all(:,2:3)';  % size 2 x nt
phi2_all = all(:,4)';  % size 1 x nt
ntime = length( time );

% define object 2 in local coordinates
s2p_poly = [ 2   3  -1  -1  2 ;                    % local x2p [in]
             3  -1  -2   2  3 ];                   % local y2p [in]
rho2 = max( sqrt( diag( s2p_poly'*s2p_poly ) ) );  % maximum radius
[ nr, n2 ] = size( s2p_poly );                     % number of points for body 2

% new figure
figure( 1 )
  clf

% plot origin and outline at each time sample
  for itime = 1 : ntime,
    t = time(itime);

% body 2 = red
    r2 = r2_all(:,itime);                  % origin
    phi2 = phi2_all(itime);                % angle
    A2 = [ cos(phi2)  -sin(phi2) ;         % attitude matrix
           sin(phi2)   cos(phi2) ];
    r2_poly = r2*ones(1,n2) + A2*s2p_poly;   % global locations for vertices
    plot( r2(1),r2(2),'ro' )                 % plot origin
    axis equal
    hold on
    plot( r2_poly(1,:), r2_poly(2,:), 'r' )  % closed curve

  end  % bottom - for itime

% bottom - show_body2.m
```