Contents

```
function [v1,v2,v3,e1,e2,e3,ie1,ie2,area] = mylists(x,y,TRI)

    % Nikhil Jayswal
    % MATH 3890
    % Machine Problem 6
    % 01 Mar 2021

    % TRI = (n_t x 3) matrix, each row = vertices of a triangle
    % x = x-coordinates of vertices
    % y = y-coordinates of vertices
    % v1, v2, v3 = vertices of triangle in counter-clockwise order
    % ie1, ie2 = edges of triangulation (ie1 < ie2)
    % e1, e2, e3 = edges of triangle in counter-clockwise order
    % area = area of all triangles

    nt = size(TRI, 1);
```

construct v1, v2, v3

initialisation

```
    v1 = TRI(:, 1);
    v2 = TRI(:, 2);
    v3 = TRI(:, 3);

    % check for counter-clockwise order
    % fix vertex labeling in case of clockwise order
    for i = 1:nt
        % create edge vectors
        edge1 = [x(v2(i)) y(v2(i)) 0] - [x(v1(i)) y(v1(i)) 0];
        edge2 = [x(v3(i)) y(v3(i)) 0] - [x(v2(i)) y(v2(i)) 0];
        % check to see if the normal vector is in the +ve z-direction
        b = cross(edge1, edge2);
        % if b(3) < 0 -> clockwise orientation -> swap any 2 vertices
        % we choose to swap v2 and v3
        if b(3) < 0
            tmp = v3(i);
            v3(i) = v2(i);
            v2(i) = tmp;
        end
    end
```

construct ie1 and ie2

```
    ie1 = [];
    ie2 = [];
    eindex = 1;
    for i = 1:nt
        ie1(eindex) = v1(i);
        ie2(eindex) = v2(i);
        eindex = eindex + 1;
        ie1(eindex) = v2(i);
        ie2(eindex) = v3(i);
        eindex = eindex + 1;
        ie1(eindex) = v3(i);
        ie2(eindex) = v1(i);
        eindex = eindex + 1;
    end
    % total number of edges (common edges counted twice) = eindex - 1
    eindex = eindex - 1;
    % ensure ie1 < ie2
    for i = 1:eindex
        if ie1(i) > ie2(i)
            tmp = ie1(i);
            ie1(i) = ie2(i);
            ie2(i) = tmp;
        end
    end
    % remove duplicates of shared edges
    ie =[ie1' ie2'];
    ie = unique(ie, 'rows');
    ie1 = ie(:, 1);
    ie2 = ie(:, 2);
    % number of edges
    eindex = length(ie1);
```

construct e1, e2, e3

since vertices have been ordered counterclockwise edges are automatically ordered if we follow the vertices

```
    e1 = zeros(nt, 1);
    e2 = zeros(nt, 1);
    e3 = zeros(nt, 1);
    for i = 1:nt
        % create edge matrix for each triangle
        ie = zeros(3, 2);
        ie(1, :) = [v1(i) v2(i)];
        ie(2, :) = [v2(i) v3(i)];
        ie(3, :) = [v3(i) v1(i)];
        % ensure ie(:, 1) < ie(:, 2)
        for j = 1:3
            if ie(j, 1) > ie(j, 2)
                tmp = ie(j, 1);
                ie(j, 1) = ie(j, 2);
                ie(j, 2) = tmp;
            end
        end
        % get edge labels from ie1 and ie2 vectors
        E = [ie1 ie2];
        [~, Locb] = ismember(ie, E, ’rows’);
        % fill out e1, e2, e3 vectors
        e1(i) = Locb(1);
        e2(i) = Locb(2);
        e3(i) = Locb(3);
    end
```

compute area of all triangles

```
    area = zeros(nt, 1);
    for i = 1:nt
        edge1 = [x(v2(i)) y(v2(i)) 0] - [x(v1(i)) y(v1(i)) 0];
        edge2 = [x(v3(i)) y(v3(i)) 0] - [x(v2(i)) y(v2(i)) 0];
        area(i) = norm(0.5*cross(edge1, edge2));
    end

end
```