# NUMERICAL NOTES

Steps to success:

## Create edgexy and Modify Initial Mesh Options

Add new variable edgexy to extraxt points from v

```
% Prepare distance-based mesh refinement data for `size_fun`
edgexy = [v(e(:,1), 1), v(e(:,1), 2), v(e(:,2), 1), v(e(:,2), 2)];

% Initial mesh options
hdata = [];
hdata.hmax = 0.6; % Maximum element size
hdata.fun = @(x, y) size_fun(x, y, edgexy); % Distance-based refinement
options = [];
options.output = false;
options.stats = true;
```

Specifically, from,

```
hdata.fun = @(x,y)
```

Change to:

```
hdata.fun = @(x, y) size_fun(x, y, edgexy);
```

## `size_fun`:

In the *Airfoil* code, change from `size_fun(x,y)` to `size_fun(x,y,edgexy)`

```
function h = size_fun(x, y, edgexy)
    % Calculate distances from each point to the airfoil boundary
    % Ensure (x, y) are combined correctly for p_poly_dist
    xv = edgexy(:,1);  % x coordinates of polygon vertices
    yv = edgexy(:,2);  % y coordinates of polygon vertices

    % Initialize the output distance array
    n = size(x, 1); % number of points
    L = zeros(n, 1);  % distance array

    % Calculate distance for each point
    for i = 1:n
```

```
        [L(i), ~, ~] = p_poly_dist(x(i), y(i), xv, yv);  % get the distance from point (x(i), y(i)) to the polygon
    end

    % Minimum and maximum element sizes
    hmin = 0.05;
    hmax = 0.3;
    decay_factor = 10; % Controls how quickly h decreases near the airfoil

    % Calculate target element size as a function of distance
    h = hmax - (hmax - hmin) * exp(-L / decay_factor);
    h = max(h, hmin); % Ensure h doesn’t go below hmin
end
```

This is converted from the orignal:

```
function h=size_fun(x,y)

  hmin=0.1; hmax=0.3; xmin=-0.5; xmax=6; ydist=0.6;
  hinc=(hmax-hmin)/ydist;

  n=size(x,1);
  h=hmax*ones(n,1);

%  return

  for i=1:n
   if xmin <= x(i,1) & x(i,1) <= xmax & abs(y(i,1)) < ydist; % Hmin can't be 0
     h(i,1) = hmin+hinc*abs(y(i,1));
   end
  end

end
```

## poly_to_dist

Insert the p_poly_dist function from the file after the size_fun() unchanged:

```
function [d,x_poly,y_poly] = p_poly_dist(x, y, xv, yv)
...
..
.
```