

Computing Project 3 *Properties of Areas*

The computing project *Properties of Areas* concerns the computation of various properties of cross sectional areas. In each of our theories (i.e., axial bar, torsion, and beams) we arrive at a point where we need certain properties of the cross section to advance the analysis. For the axial bar we needed only the area; for torsion we needed the polar moment of inertia; for the beam we will need moment of inertia of the cross section about the centroid of the cross section.

We can develop an algorithm that allows the computation of all of the properties of a cross section if the cross section can be described as a polygon. The algorithm is built on formulas for the properties of a triangle. What that program will do is create a triangle from the origin and the two vertex nodes associated with a side of the polygon. Whether this polygon adds or subtracts from the accumulating properties will be determined from the vectors defining the sides of the polygon (see the *CP Notes* for further clarification). If you loop over all of the sides, the end result will be the properties of the entire cross section.

The general steps are as follows:

1. Develop a routine that allows you to describe the cross section with a sequence of points numbered in a counterclockwise fashion starting with 1. The last point should be a repeat of the first one in order to close the polygon. Some suggestions:
 - a. Store the (x,y) coordinates of each point in a single array $\mathbf{x}(N,2)$, where N is the number of points required to describe the cross section (including the repeat of the first point as the last point) and the first column contains the x values of the coordinate and the second column contains the y values of the coordinate and the second column contains the y value.
 - b. It will eventually be a good idea to put the input into a MATLAB function and call the function from your main program. That way you can build up a library of cross sectional shapes without changing your main program.
 - c. If you need a negative area region (for a cutout section like in an open tube) then number the points in that region in a counter-clockwise fashion. Just keep numbering the vertices in order (no need to start over for the negative areas).
2. Develop a routine to loop over all of the edges of the polygon and compute (and accumulate) the contributions of the triangle defined by the vectors from the origin to the two vertices of the current side of the triangle (that gives two sides) and the

vector that points from the first to the second vertex (in numerical order). Calculate the area, centroid, and outer-product contributions to the properties (see the CP Notes for clarification of this issue).

3. Compute the orientation of the principal axes of the cross section using the eigenvalue solver in MATLAB (`eig`) on the moment of inertia matrix \mathbf{J} . See the CP Notes for more information on this task.
4. Create an output table (print into the Command Window) giving the relevant cross sectional properties. Develop a routine to plot the cross section. Include the location of the centroid of the cross section in the graphic (along with lines defining the principal axes if you can figure out how to do that).
5. Generate a library of cross sections, including some simple ones (e.g., a rectangular cross sections) to verify the code. Include in your library as many of the following cross sections as you can get done:
 - a. Solid rectangle with width b and height h .
 - b. Solid circle of radius R .
 - c. Rectangular tube with different wall thickness on top and bottom.
 - d. I-beam with flange width b , web depth d , flange thickness t_f , and web thickness t_w .
 - e. Angle section with different leg lengths and leg thicknesses.
 - f. Circular tube with outside radius R and wall thickness t .
 - g. T-beam.
6. Use the program to explore aspects of the problem. For example,
 - a. Why is it more efficient to use an open circular tube for torsion rather than a solid cylinder?
 - b. For beam bending we can control deflections and reduce stresses with a large moment of inertia about the axis of bending. Show the trade-offs available in an I-beam when you can select different web and flange depths and thicknesses. What is the ideal allocation of material? Why would we never actually do that in practice?
 - c. Demonstrate that the principal axes of a symmetric cross section lie along the lines of symmetry. You can do this by showing that the off-diagonal elements of \mathbf{J} are zero for symmetric sections with axes so chosen.
 - d. Explore any other feature of the problem that you find interesting.

Write a report documenting your work and the results (in accord with the specification given in the document *Guidelines for Doing Computing Projects*). Post it to the *Critviz* website prior to the deadline. Note that there is only one submission for this problem (the final submission).

Please consult the document *Evaluation of Computing Projects* to see how your project will be evaluated to make sure that you can get full marks. Note that there is no peer review process for reports in this course.