MECH 222 Computer Lab 2—Triangulated Mesh Surfaces

Read and understand the briefing notes and lab instructions before your lab period. Each lab activity corresponds to one of the main ideas in the briefing notes.

Learning Objectives By the end of this lab, you should be able to:

- Understand and manipulate Matlab's standard data structure for describing triangulated mesh surfaces in space.
- Make an informed choice between Matlab's command `trisurf` and `trimesh` for drawing such surfaces.
- Describe two possible ways of representing a triangle in Matlab, and know which of the two is "correct" (i.e., standard).
- Write a Matlab function to calculate an area-scaled normal vector for a given triangle.
- Use a `for`-loop to iterate over the triangle data structure and do something to each triangle in the figure.
- Use the accumulator idea to aggregate total surface area for a triangulated figure, or for selected subregions.
- Use the `get` and `set` functions to view and modify basic properties of graphics objects through their handles.

PRE-LAB ASSIGNMENT

PL1: Consider the shed shown in Figure 1 of the Briefing Notes.

(i) Calculate the area and the centroid for each of the five faces in the sketch.

(ii) Calculate the area and the centroid for the whole shed considered as a single object.

PL2: Given the points $A(2,1,3)$, $B(4,-2,3)$, $C(3,3,5)$, calculate by hand the area of $\triangle ABC$. Call the area dS. Also find a unit vector normal to the plane containing this triangle. Call the vector $\widehat{\mathbf{n}}$. Calculate the vector $\mathbf{N} = \widehat{\mathbf{n}}\, dS$.

PL3: Find two different vectors of length 1/10 that are perpendicular to the plane containing the triangle specified in PL2, above.

PL4: Read the description of function `n_dS` required in Activity 3. Write down by hand the Matlab vectors `Xtri`, `Ytri`, `Ztri` for which the command `n_dS(Xtri,Ytri,Ztri)` should instantly calculate the answer to problem PL2. (You will test this in the lab.)

ACTIVITY 1: *Display and manipulate a 3D surface.*

This activity produces nothing to hand in. Allow yourself 10 minutes just to get familiar with the display system that you will be using later.

1.1. Copy the files `hand.mat` and `drawhand.m` from Vista into your current workspace.

1.2. Type `drawhand` at the prompt to put a picture of Olivier's hand on the screen. Experiment with the tools that let you rotate the image, change its colour, zoom in and out, etc. Try turning the edge display off using the command discussed in the Briefing Notes, i.e., `set(Ohandle,'EdgeColor','none')`. Can you find out how to turn it back on?

1.3. Use the command `type drawhand` or open Matlab's built-in editor to read the commands that make the script work. It's not complicated, and there are some useful hints on handle graphics in there.

ACTIVITY 2: *Give Caesar the measles.*

2.1. Write a Matlab script that does two things:

(i) draws a wire mesh image for the triangulated surface encoded by the whatever values of `X`, `Y`, `Z`, and `FACELIST` are currently in the active workspace;

(ii) makes red dot at the centroid of each triangle in the mesh.

Test your script on something simple, like the 5-triangle shed in Figure 1 of the Briefing Notes. (You can copy-paste three displayed Matlab sequences from the Briefing Notes to set up the workspace variables.)

2.2. Copy the file `jc3d.mat` from Vista into your workspace. Adapt the script file `drawhand.m` from Activity 1 so that it draws Julius Caesar's face instead of Olivier's hand. (Give the new script a different name.) The variables `X`, `Y`, `Z`, `FACELIST` have the same interpretation for both objects. Play around briefly with the image you get. There is nothing to hand in yet.

2.3. Run your measle script from item 2.1 on Caesar. Hand in a printed copy of the picture you get and the script that produced it.

ACTIVITY 3: *Write a function that analyzes flying triangles.*

Let's agree to represent a general triangle $\triangle ABC$ with vertices at $A(x_1, y_1, z_1)$, $B(x_2, y_2, z_2)$, and $C(x_3, y_3, z_3)$ by storing the x-coordinates in one vector, the y-coordinates in another, and the z-coordinates in a third:

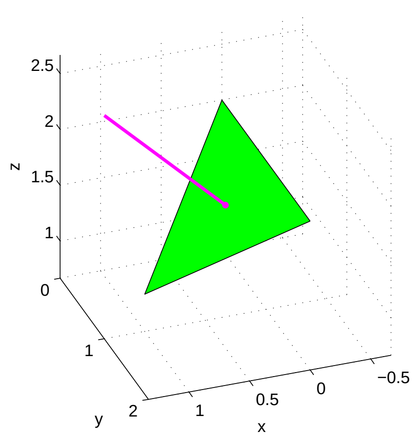
$$\texttt{Xtri} = (x_1, x_2, x_3), \qquad \texttt{Ytri} = (y_1, y_2, y_3), \qquad \texttt{Ztri} = (z_1, z_2, z_3).$$

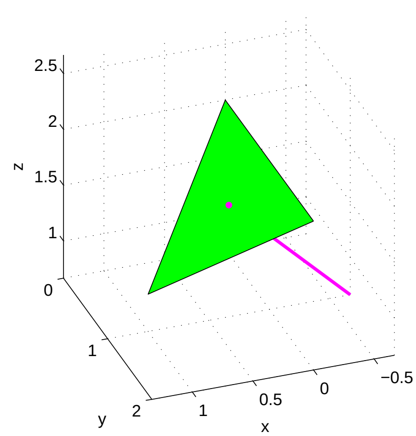
Create a function named `n_dS` for which the command `N = n_dS(Xtri,Ytri,Ztri)` returns a 3-element vector `N` with the following properties:

(i) The vector magnitude (norm) of `N` equals the area of $\triangle ABC$.

(ii) The direction of `N` is perpendicular to the plane containing $\triangle ABC$.

(Make sure the command works even if the triangle's area equals 0.) Test your code on the triangle specified in prelab problem PL2.





Notes: Every triangle of positive area has two opposite normal directions, so the result of your work could be illustrated by either of the sketches shown above. Both are fully acceptable at this stage. Here are some highlights from the code that produced those sketches.

```
Xtri = [0, 1, 0];
Ytri = [2, 1, 0];
Ztri = [2, 1, 2];

th = fill3(Xtri,Ytri,Ztri,'g');  % draw filled triangle, keep handle
set(th,'FaceAlpha',0.6);         % make it partially transparent
hold on;                         % prepare figure to draw more

N = n_dS(Xtri,Ytri,Ztri);        % normal N has norm(N)=Area

% Draw magenta vector emerging from triangle centroid
xbar = mean(Xtri); ybar = mean(Ytri); zbar = mean(Ztri);
n1 = plot3([xbar,xbar+N(1)], [ybar,ybar+N(2)], [zbar,zbar+N(3)], 'm');
```

Warning. The three input arguments for your function `n_dS` are NOT three (x, y, z) points in space. They are separate lists of three x-coordinates, three y-coordinates, and three z-coordinates. Both interpretations provide a way to inject 9 real numbers into `n_dS` for calculation, but *they are not the same, and only the second one is correct.* Many student-hours have been lost to confusion on this point!

Hand-in Checklist:

- ☐ A printed copy of your function `n_dS`.
- ☐ A transcript showing how to apply your function to triangle $\triangle ABC$ from Prelab PL1, and confirming that it returns the same vector as your manual calculation in the Prelab.

ACTIVITY 4: *Find Surface Areas and Centroids for Surfaces in Space*

Write a Matlab script that operates on workspace variables (much as in Activity 2). Use it to solve all three problems below. Remember that the Briefing Notes provided a simple procedure based on the results of Computer Lab 1.

4.1. For the shed in the Briefing Notes, calculate
   - (i) the surface area [one side only], and
   - (ii) the coordinates of the centroid.

4.2. For Olivier's hand, calculate
   - (i) the surface area [one side only], and
   - (ii) the coordinates of the centroid.

4.3. For the part of Caesar's head supplied in our model, calculate
   - (i) the surface area, and
   - (ii) the coordinates of the centroid.

Hand-in Checklist:

- ☐ A printed copy of your script.
- ☐ Your calculated results.