

# CS 109 – C/C ++ Programming for Engineers w. MatLab– Summer 2012

## Homework Assignment 2

### Calculation of Barycentric Coordinates & Point Classification

**Due: Wednesday June 27th by 8:00 a.m., via Blackboard.** Optional hard copy may be turned in during lab on Thursday.

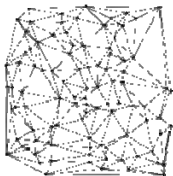
#### Overall Assignment

For this assignment you are to write a simple computer program that calculates the barycentric coordinates of a point based upon its Cartesian ( x, y ) coordinates, and the Cartesian coordinates of the three vertices of a triangle. Then the point will be classified as either an interior point, exterior point, edge point, or vertex based upon the barycentric coordinates.

#### Background: Triangulation ( from HW1 )

Engineers and scientists are often faced with the problem of interpolating function values over an area, based on measurements taken at irregularly spaced locations. For example, weather data is measured principally at airports, which are distributed somewhat randomly about the country, and oil well test sites are also irregularly spaced. Modern computer graphics is also based on interpolating color values between the vertices of triangle meshes.

One approach to this problem is to first *triangulate* the space based on the known data points, which basically means to subdivide the space into triangles as shown in the following diagram:



Delaunay triangulation of a random set of 100 points in a plane. ( Source: [http://en.wikipedia.org/wiki/Delaunay\\_triangulation](http://en.wikipedia.org/wiki/Delaunay_triangulation) )

There are many different algorithms for triangulating a space, all of which are beyond the scope of this assignment.

#### Background: Barycentric Coordinates ( from HW1 )

Once a triangulation has been determined, and the particular triangle that contains the point of interest has been identified, then the next step is to interpolate the function values from the vertices of the triangle to the point of interest. This is where barycentric coordinates come in.

Given a triangle defined by vertices  $r_1$ ,  $r_2$ , and  $r_3$ , then any point  $r$  interior to the triangle can be defined by a weighted sum of the three vertices, as given by the equations:

$$r = \lambda_1 r_1 + \lambda_2 r_2 + \lambda_3 r_3$$

where  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are the barycentric coordinates, subject to the constraints:

$$\lambda_1 + \lambda_2 + \lambda_3 = 1.0$$

$$0.0 \leq \lambda_i \leq 1.0 \quad \forall i \text{ in } 1, 2, 3 \text{ for points interior to the triangle.}$$

A function  $f$  whose values are known at the vertices can be interpolated as:+

$$f(r) = \lambda_1 f(r_1) + \lambda_2 f(r_2) + \lambda_3 f(r_3)$$

## Calculation of Barycentric Coordinates

Given the Cartesian coordinates of a point of interest,  $(x, y)$ , and the Cartesian coordinates of the three vertices of a triangle,  $((x_1, y_1), (x_2, y_2), \text{ and } (x_3, y_3))$ , the barycentric coordinates of the point can be calculated as<sup>1</sup>:

$$\lambda_1 = \frac{(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)}$$

$$\lambda_2 = \frac{(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)}$$

$$\lambda_3 = 1 - \lambda_1 - \lambda_2$$

## Classification of Points Based on Barycentric Coordinates

Given the Barycentric coordinates of a point of interest the point may be classified as follows:

Conditions	Classification
$0.0 < \lambda_i < 1.0 \forall i \text{ in } 1, 2, 3$	Interior. ( Note strict less-than operators )
One $\lambda = 0.0, 0.0 < \lambda_i < 1.0$ for the other 2 $\lambda$ s	Edge.
One $\lambda = 1.0$ , The other two $\lambda$ s = 0.0	Vertex
At least one $\lambda < 0.0$ or $> 1.0$	Exterior

Note: It may be more efficient to check for the classifications in an order other than the one presented in the table above.

## Program Details

For this assignment you are to write a simple computer program that calculates the barycentric coordinates of a series of points, given the Cartesian coordinates of the three vertices of a triangle and the Cartesian coordinates of the points in question. Your program should also classify each point as interior, exterior, edge, or vertex. Your program should first print out your name and ACCC account name ( e.g. astudent ), and explain to the user what the program does.

- Your program should then ask the user to enter the Cartesian coordinates of the three vertices of the triangle. This step will only be done once for the program. First get the x and then y coordinates for the first vertex, then for the second vertex, and then for the third vertex. ( Six questions all together. )
- Next ask the user how many points they would like to calculate the barycentric coordinates for. This answer should be checked to verify that it is not negative.

---

<sup>1</sup> Source: [http://en.wikipedia.org/wiki/Barycentric\\_coordinates\\_\(mathematics\)](http://en.wikipedia.org/wiki/Barycentric_coordinates_(mathematics))

- For however many points the user has requested, your program should then perform the following steps:
  - Ask the user for the x and then y Cartesian coordinates of the point desired.
  - Calculate the barycentric coordinates.
  - Classify the point as interior, exterior, edge, or vertex. This should be done using an if-else block, and should assign a value to a variable depending upon the classification. ( This may be done with a simple int, though an enum is actually a better choice. )
  - Report the barycentric coordinates, and the classification of the point. Use a switch block based on the variable set in the previous step to report the classification.
- After completing the loop through the points, report how many total points were calculated, as well as how many were interior, exterior, edge, or vertex ( 4 counters plus the total. These counters should be incremented in the switch block, not the if-else block. )

### Special Notes:

- **You should work out some sample problems by hand before writing any computer code.** You may want to use some graph paper to help identify test cases for interior, exterior, edge, and vertex points. Initially you may want to use “easy” triangle vertices, such as ( ( -1, 0 ), ( 1, 0 ), ( 0, 1 ) ), but is it also a good idea to test your program with randomly located vertices, including both positive and negative values.
- This problem does not involve any units. ( Normally you would have to be careful about units, and possibly convert units, such as feet to meters. )
- Now that you know about loops, you should always test user input to confirm it is in a valid range. For this particular problem, all possible values of coordinates are valid, so the only input that can be checked is the number of points to be calculated.

### Evolutionary Development

When dealing with large complex programs, it is generally best to work one step at a time, getting part of the program working and tested before attempting the next part. For this assignment you may want to do your development in the following steps:

1. Write a program to calculate a single set of barycentric coordinates, without doing any error checking, looping, or classification.
2. Add the code to ask the user for how many problems they want to solve, and check their input.
3. Modify the code to solve the number of problems requested by the user.
4. Add the if-else code to determine the classification and assign a value to a variable. For now just print out the stored value.
5. Add the switch to report the classification based on the stored variable value. Note that the counters for how many points were interior, etc. are incremented in the switch.
6. Any other polish and refinement that may be needed, including optional enhancements.

## What to Hand In:

- Your code, **including a user documentation file**, should be handed in electronically using Blackboard.
- The intended audience for the documentation file is a general end user, who might want to use this program to perform some work. They do not get to see the inner workings of the code, and have not read the homework assignment. You can assume, however, that they are familiar with the problem domain ( e.g. the data to be interpolated. )
- A secondary purpose of the documentation file is to make it as easy as possible for the grader to understand your program. If there is anything special the grader should know about your program, be sure to document it in the documentation file. In particular, if you do any of the optional enhancements, then you need to document what they are and anything special the TA needs to do to run your program and understand the results.
- If there are problems that you know your program cannot handle, it is best to document them as well, rather than have the TA wonder what is wrong with your program.
- Make sure that your name appears at the beginning of each of your files. Your program should also print this information when it runs.

## Optional Enhancements:

It is course policy that students may go above and beyond what is called for in the base assignment if they wish. These optional enhancements will not raise any student's score above 100 for any given assignment, but they may make up for points lost due to other reasons.

- Ask the user if they would like to solve additional triangles, and if so, repeat until they indicate they are done.
- Other enhancements that you think of – Check with TA for acceptability.

## Future Work:

Once we learn about functions, we will be breaking the work in this assignment up into separate functions – One to calculate the barycentric coordinates and one to classify points at a minimum. Once we learn about files and arrays, we will start working with a network of triangles, with the triangle vertices and the list of which vertices make up each triangle read in from data files. We can then use the classification function to identify which of our triangles contains the point of interest, and then interpolate function values based on the vertex values and barycentric coordinates, as in HW1.

For future assignments we will be using the concept of barycentric coordinates and triangulation to interpolate and plot weather conditions based on data from NOAA weather buoys, as shown in the following map, most likely limited to the southern half of Lake Michigan, e.g. stations SGNW3 to MKGM4 and below.

