

CS 109 – C/C ++ Programming for Engineers w. MatLab– Fall 2011

Homework Assignment 1

Solving the SSS Triangle Problem for Use in Trilateration

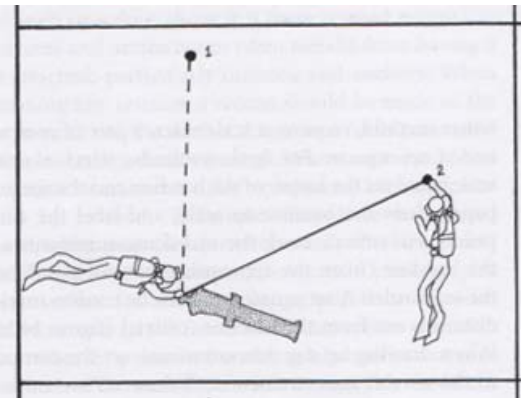
Due: **Wednesday August 31st by 8:00 a.m., via Blackboard.** Optional hard copy may be turned in during class.

Overall Assignment

For this assignment you are to write a simple computer program that finds the angles of a triangle given the lengths of the three sides.

Background: Trilateration

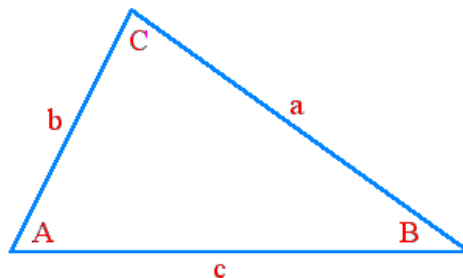
There are many situations in science and engineering where it is necessary to determine the locations of points in a 2-D plane, such as the archaeological survey of an underwater shipwreck site. One approach to this is *trilateration*, which involves measuring the distances from the unknown point to two known reference points. Assuming the distance between the two known points is also known, this forms a unique triangle, which can be used to calculate the position of the unknown point from the known positions of the two reference points¹:



Background: Solution to the SSS problem

The first step in calculating the unknown position using trilateration is to determine the angles of the triangle, which involves solving the SSS, or side-side-side problem.

Given a triangle with sides of known lengths labeled a , b , and c , having unknown opposing angles labeled A , B , and C respectively as shown in the figure below, and furthermore given that side c is the longest of the three sides, solve for the unknown angles²:



The first step is to solve for angle C, using the cosine law:

$$c^2 = a^2 + b^2 - 2ab \cos(C) \quad (1)$$

which can be rearranged to solve for the angle C:

$$C = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \quad (2)$$

Then the next step is to solve for the remaining two angles using the sine law:

$$\frac{\sin(A)}{a} = \frac{\sin(B)}{b} = \frac{\sin(C)}{c} \quad (3)$$

which can be rearranged to solve for the angles A and B:

$$A = \sin^{-1}\left(\frac{a}{c} \sin(C)\right) \quad (4)$$

$$B = \sin^{-1}\left(\frac{b}{c} \sin(C)\right) \quad (5)$$

Program Details

For this assignment you are to write a simple computer program that calculates the three angles of a triangle, given the lengths of the three sides, and reports the angles in radians and degrees.

- 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 length of the longest side and then the lengths of the other two sides, **IN THAT ORDER**.
 - Each data item should be entered separately, with its own question and answer. I.e. with three data values to enter, your program should ask three separate questions and get a separate response for each one, rather than asking the user to enter multiple values in response to a single question.
- After performing the necessary calculations, the program should report the results, **including an echo of the user's input.** For this program the results can be shown nicely in a table format:

Side Length	Opposing Angle (radians)	(degrees)
20.8	1.309	74.998
15.7	0.817	46.809
18.3	1.016	58.192

- Don't worry excessively about making the output "pretty", since we have not yet covered that. However you should label the table and report the angles in both radians and degrees.

Special Notes:

- **You should work out some sample problems by hand before writing any computer code.** The sample results shown above are believed to be accurate, but you should verify this by working the problem out by hand.
- **You will need to report dimensional units.** For this particular problem the input dimensions are unimportant, so long as they are all in the same units. You will have to convert the angles for output.
- You will need to use the math library for this assignment, which means you must `"#include <cmath>"` at the top of your program. The math library functions for sine, cosine, arcsine, and arccosine are `sin()`, `cos()`, `asin()`, and `acos()` respectively, **with angles in radians.**
- The variable names A, B, C, a, b, and c used in the mathematical discussion above are **REALLY BAD** names for variables in your program. Better names would be something like `side1` or `length1` and `angle1` (or `angle1Radians` and `angle1Degrees`), etc.
- The basic assignment should not use any loops. You may use loops in your program only if you implement one of the optional enhancements (see below), e.g. to allow the user to solve multiple problems without restarting the program.
- Since you haven't learned how to test things yet, you can just assume that all of the user input is good, i.e. that the user enters the largest side first and doesn't enter any non-positive lengths or impossible triangles.

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. trilateration, the SSS problem.)
- 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.

- Check the accuracy of the calculations by subtracting the sum of the three angles from 180 degrees or pi radians. The result should be zero, so any difference would be the result of round-off errors. (As an added check, repeat the calculations using float data types instead of doubles, and see how the round-off error changes. You should be able to do this with a single program that originally reads in the data as doubles, and then later copies the data into float variables to perform the alternate calculations.) In your documentation file you should discuss your findings
- Calculate the "heights" of the triangle measured from each of the three sides, defined as the perpendicular distance from that side to the opposing vertex, and include those results in your table. You could also calculate the area based on each of the three sides and compare the results. (If they are not equal, then any difference is the result of round-off error.)
- Check the data entered to verify that it is valid.
- Ask the user if they would like to solve additional problems, and if so, repeat until they indicate they are done.
- Other enhancements that you think of – Check with TA for acceptability.

¹ Amanda Bowens, "Underwater Archaeology, the NAS Guide to Principles and Practice", Second Edition, The Nautical Archaeology Society, 2009.

² http://www.teacherschoice.com.au/maths_library/trigonometry/solve_trig_sss.htm, viewed 24 August 2011.