CITS2401 Computer Analysis and Visualization 2017 Semester 2 Assignment 2

Due date: **06 October 2017 @ 5:00PM**

Total marks: 10

Assignment submission instructions:
This assignment is to be completed in Cody Coursework (https://coursework.mathworks.com/) in the similar manner as you did for Assignment 1. Failure to submit it in Cody Coursework will result in zero grade. Your code will be tested on Cody Coursework. However final grade will be awarded based on testing your code with different testing data and observing your output plots which will be checked manually.
Assignment consists of a single problem divided into two sets of test cases. You are required to run both test cases in Cody Coursework otherwise your submission will not be considered. The same code should also be submitted in both test sets of test cases.

Plagiarism.
All work submitted should be your own. I am sure that you agree that this is for your own good!! If you do not agree, please note that we have ways to detect plagiarism in code. Incidences of plagiarism will be taken seriously and will involve follow-up with the Head of the School and consequences to academic results.

Problem

Drones are getting very popular and expected to play an important role in future. One of the companies has the problem of counting the number of drones in an area and their access and manoeuvrings in a specific area. The solution to this problem can help to use drones in different types of missions including drones traffic control, drone delivery optimization and drone medical assistance.

Consider a field having many drones. Each drone can move in three dimensions but needs some area around it for its stability to overcome factors such as wind and gravity. This required area can be represented by a sphere where the centre of the sphere is the centre of the drone. The radius of the sphere is dependent on the size and stability of the drone and is taken as the maximum distance between any two points on the surface of drone.

We have a 3D imaging system which can locate the centre points of spheres (drones) and the radii of the spheres around each drone. The 3D imaging system records these details in a CSV file and upload it on a server. The problem becomes interesting when multiple drones need to move in the field without collision for which we need to know the intersection of the spheres.

Figure 1: Multiple drones in an area

Write a function drones to import a CSV file available online which is created by the 3D imaging system. On each line of the CSV file, it contains information about a sphere around drone. The first three integers on each line represent the centre of the sphere in three dimensional space, which are x, y and z. The fourth integer on the line represents the radius of the sphere. All dimensions are provided in meters.

The function needs to provide the output that informs which of the spheres are colliding or touching each other and which are not touching with any other sphere. In addition, how many overlaps between two spheres exist? One overlap is considered when two spheres around drones touch or intersect each other. Therefore, one sphere around a drone can have multiple overlaps or touches with many other spheres around other drones. The distance between two points (x_1, y_1, z_1) and (x_2, y_2, z_2) in three dimension can be found by:

$$distance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

In addition, the function also needs to plot the spheres in three dimensional spaces [use surf or mesh] in one figure having two plots [use subplot]. In the first plot all spheres returned are plotted. While in the second plot only non-touching spheres are plotted. Don't forget to label the axes and include the title for each plot. Follow all the guidelines of plotting and visualization explained to you during the lectures. Ensure that the view of your plots look similar to the referenced plots shown in testing results in Cody Coursework.
[Hint: You need to study these commands: subplot, surf, mesh, sphere,view]

Below is the sample function definition mentioning the inputs and required outputs.

```
[touches, touchlist, ntouchlist] = drones(fileLink)
```

where input fileLink is the link of the CSV file and the outputs are:

touches: Number of overlaps/touches among all spheres present in the CSV file.

touchlist: A structure containing information of all spheres which have overlap with any other sphere. The fields of the structure are: x, y, z and radius. The order of spheres data must remain the same as provided in the CSV file excluding non-touching spheres.

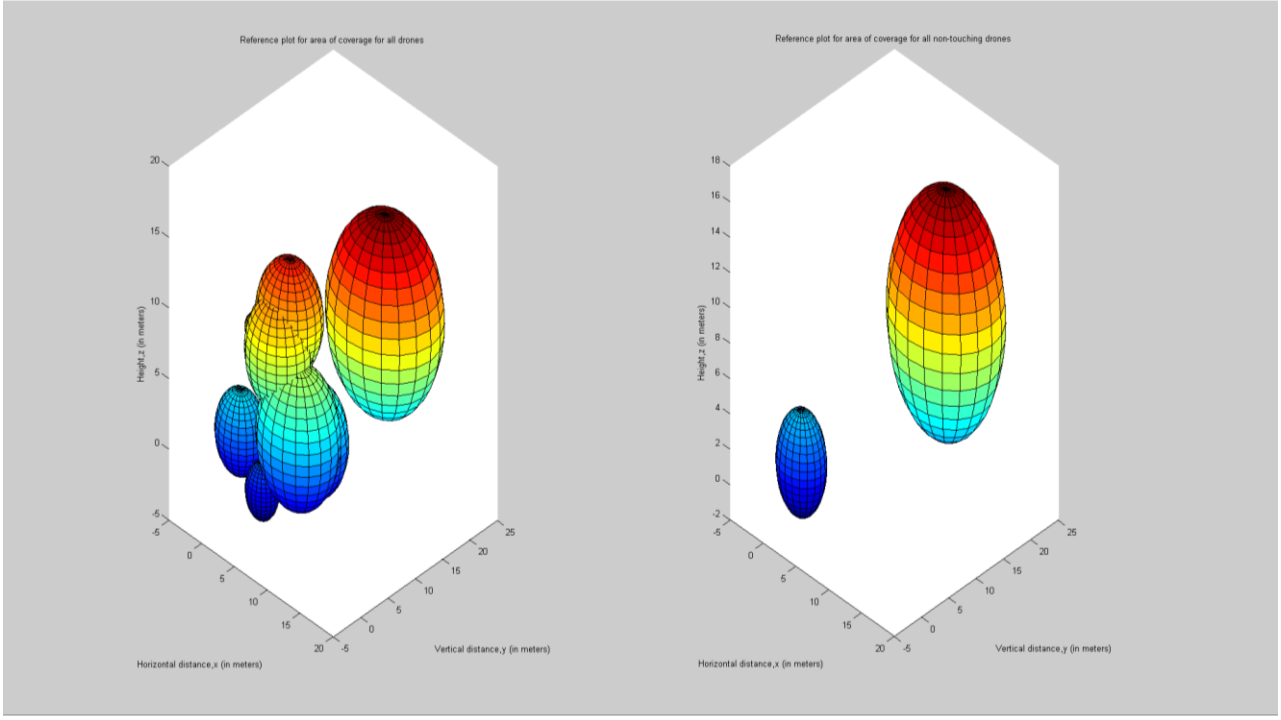
ntouchlist: A structure containing information of all spheres which are not touching any other sphere. The fields of the structure are: x, y, z and radius. The order of spheres data must remain the same as provided in the CSV file excluding touching spheres.

The sample CSV file provided is drone.csv which is available on LMS and at the link (http://teaching.csse.uwa.edu.au/units/CITS2401/drone.csv). Following are the sample commands and their results for validating your code. The sample data is provided for testing purpose and values of answers need to be matched. Ignore text formatting of the commands and their outputs.

The sample plots generated by the execution of the sample command are also given below. Remember your default plots may look different depending on the use of surf or mesh. You need to rotate plots in 3D so that they look like the reference plots. For Cody Coursework, change the view of the plot to ensure that it looks similar to the reference testing plots.

```
>> fileLink = 'http://teaching.csse.uwa.edu.au/units/CITS2401/drone.csv';

>> [touches, touchlist, ntouchlist] = drones(fileLink)

touches = 15

touchlist = 1x8 struct array with fields:
    x
    y
    z
    radius

ntouchlist = 1x2 struct array with fields:
    x
    y
    z
    radius


>> struct2table(touchlist)
ans =
    x    y    z     radius

    _    _    __    ______
    5    3     8    4
    4    7     3    3
    2    2     9    1
    5    5    10    4
    5    0    -1    2
    7    4     2    3
    8    6     2    4
    8    3     3    5

>> struct2table(ntouchlist)
ans =
    x     y     z     radius

    __    __    __    ______
    12    14    10    7
     0     2     1    3
```

Note: Your submitted code will be tested with different CSV files and may be graded manually.



Reference plot for area of coverage for all drones

Reference plot for area of coverage for all non-touching drones

Figure 2: Sample output