

Coursework Report

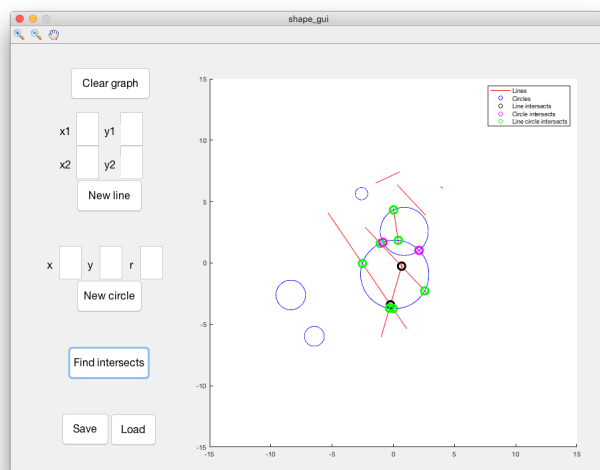
JACK FURBY

CM2104

December 10, 2017

1 Overview

1.1 Interface



My application for displaying line segments and circles with their intersection point allow circles to be added by entering the X and Y coordinates of the center along with the radius. Lines are added by specifying the X and Y coordinates of each end of the segment. If a circle or line is entered incorrectly (missing a value, not a number or a negative radius) an error message is displayed. If nothing is entered a random line or circle is created. Once a line or circle has been created it cannot be edited. The user will have to clear the axis to remove any element. There is also the option to save and load lines and circles. For both lines and circles the values the user enters are saved into matrices. Once required these values will be converted to parametric form for lines remain untouched for circles.

1.2 Finding intersects

When finding intersects I am using a function which takes in two matrices one containing lines and the other circles. From here lines are put into parametric form with one point saved with the change in X and Y. circles stay in the same format as they are in the matrix. To get

all comparisons I am using `nchoosek` for line intersects and circles intersects and `combvec` for lines and circles. This ensures I am not comparing the same 2 lines, circles or combos more than once. To get the intersection points I am using `lines_par_int_2d` (Burkardt 2005c), `circle_imp_line_par_int_2d` (Burkardt 2005a), `circles_imp_int_2d` (Burkardt 2005b) and `r8_epsilon` (Burkardt 2004) which return intersection points along with a few other bits of information which I then use to validate the result e.g. make sure the intersection with a line is within the line segment.

2 Main elements

To find intersects my program takes in two matrices. One of these contains rows of lines containing the values for x_1 , y_1 , x_2 , and y_2 which represent the two end points for line segments. Circles are also entered one per row but contains the values x , y and r representing the center of the circle and radius. The output is in the form of 3 matrices containing interception points. The matrices are line intersection points, line and circle intersection points and circle interception points.

My GUI allows a user to place lines and circles onto the axis by specifying the required points for lines and circles. There is no limit to the number of points the user can place and they can zoom out of the axis if they need to change the range. Once the 'find intersects' button has been pressed all interceptions will be found and circled on the axis. The GUI was developed using GUIDE.

3 Extra features

For extra features I have made it possible for users to save and load what they have done. This includes a file explorer and will save lines and circles to a .mat file. Loading will load the lines and circles but will not run find intersects as I am assuming the user can do that themselves if they need to.

I also added advance controls making it possible for the user to generate random lines and circles. This can be done by leaving the input fields blank and just clicking on new line or circle. It is also possible to clear the graph and reposition what part of the axis the user wishes to view.

References

Burkardt, J. (2004), 'r8_epsilon'. [Accessed: 9 December 2017].

Burkardt, J. (2005a), 'Circle_imp_line_par_int_2d'. [Accessed: 9 December 2017].

Burkardt, J. (2005b), 'Circles_imp_int_2d'. [Accessed: 9 December 2017].

Burkardt, J. (2005c), 'Lines_par_int_2d'. [Accessed: 9 December 2017].