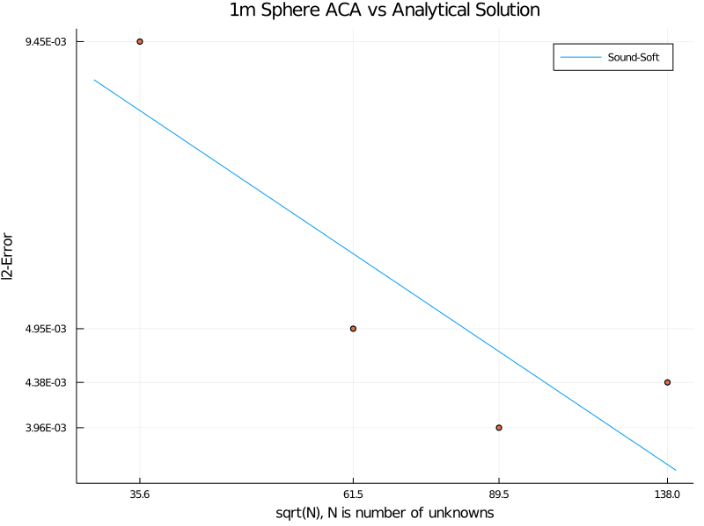
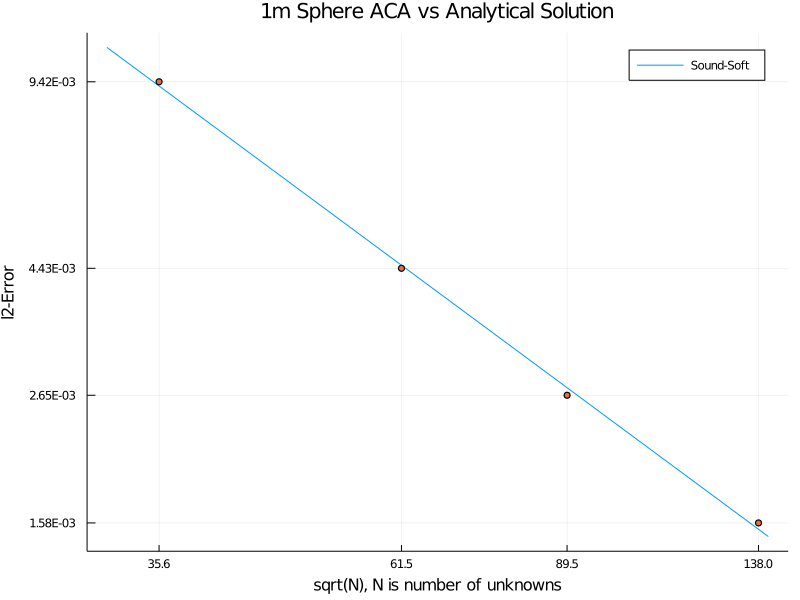
# ACA Testing

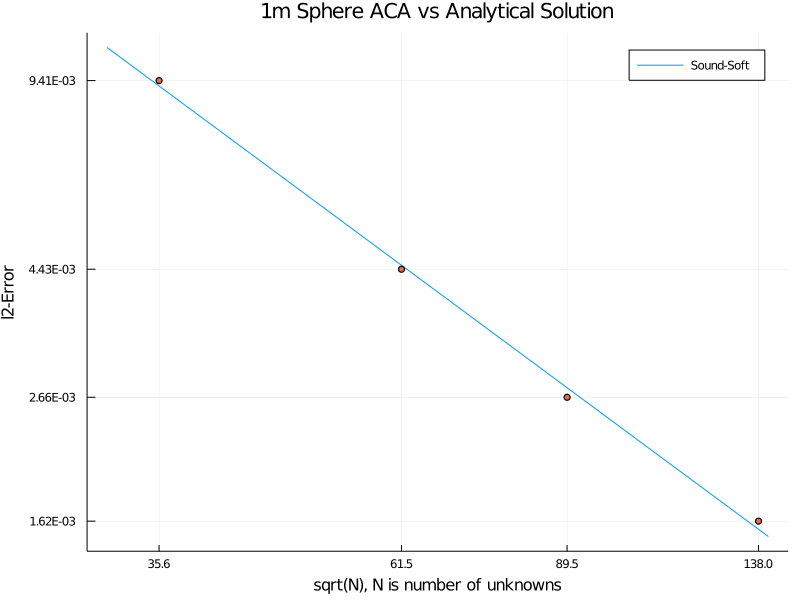
This document summarizes results of testing the ACA solver in my code and what insights they reveal

## Test Group 1:

These tests run a geometric sphere with a spherical incident wave. As the mesh is refined, the wavelength is set to ~200 element edge lengths.

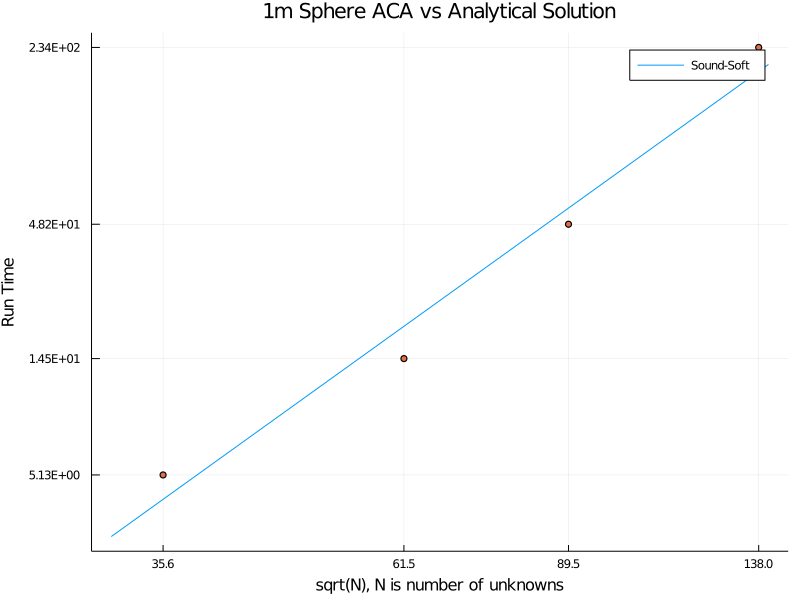
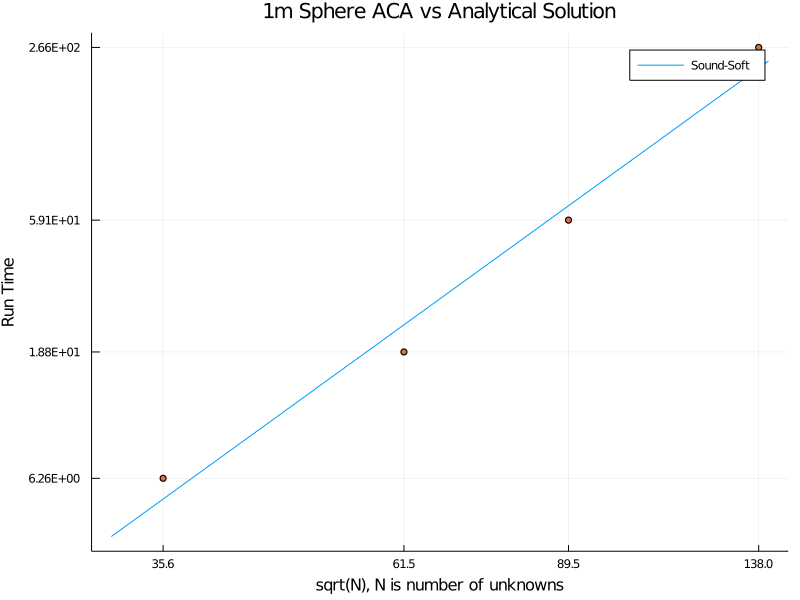
### ACA Approximation Tolerance Testing

Chart, line chart, scatter chart

Description automatically generatedThe ACA approximation tolerance, , determines when the decomposed sub-Z matrices are high enough rank to approximate the true sub-Z matrices. This test incrementally decreases the parameter to observe its effect on timing and error when compared to an analytical solution.

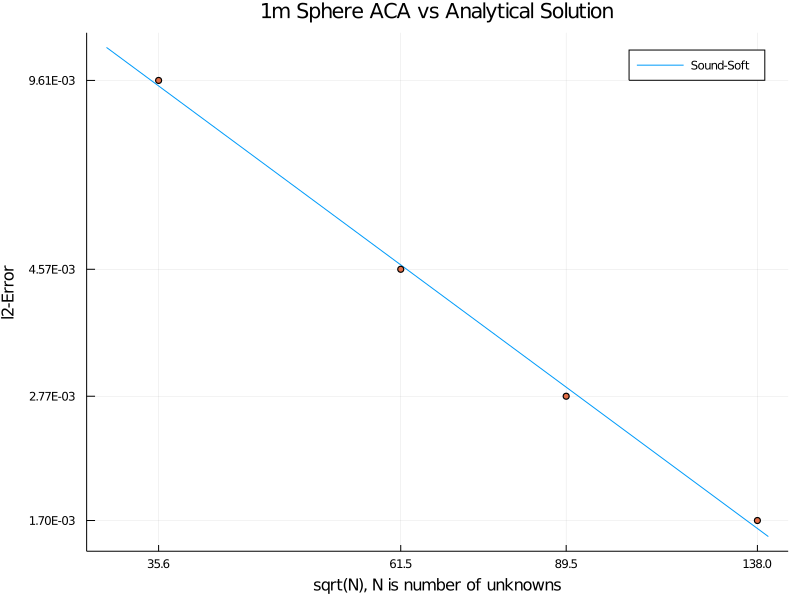
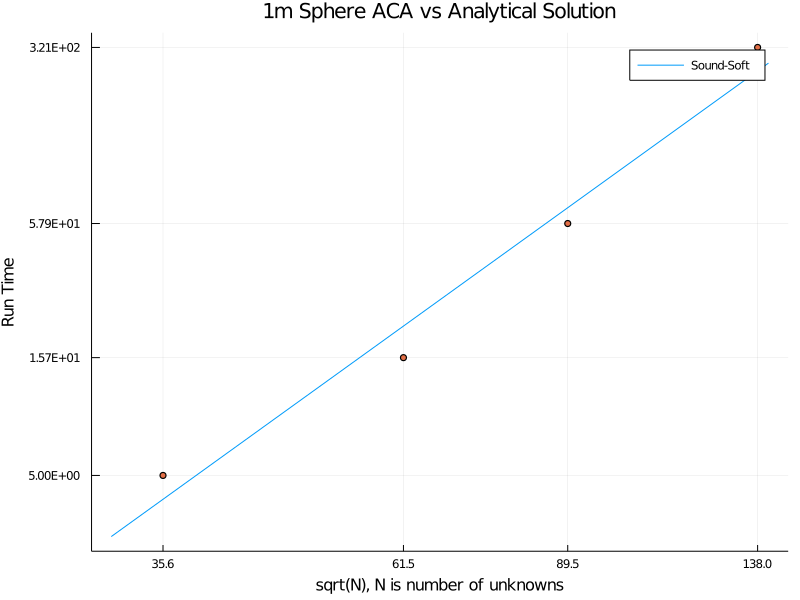
**Figure 1**: From left to right and top to bottom: ,, and . The slope of the logarithmic fit for the third and fourth plots are -1.31 and -1.32, respectively.

The above plots reveal that the ACA approximation tolerance needs to hit at least to see convergence of the error as mesh is refined. Beyond that, there is little to no benefit of decreasing the tolerance. In terms of run time, see below.

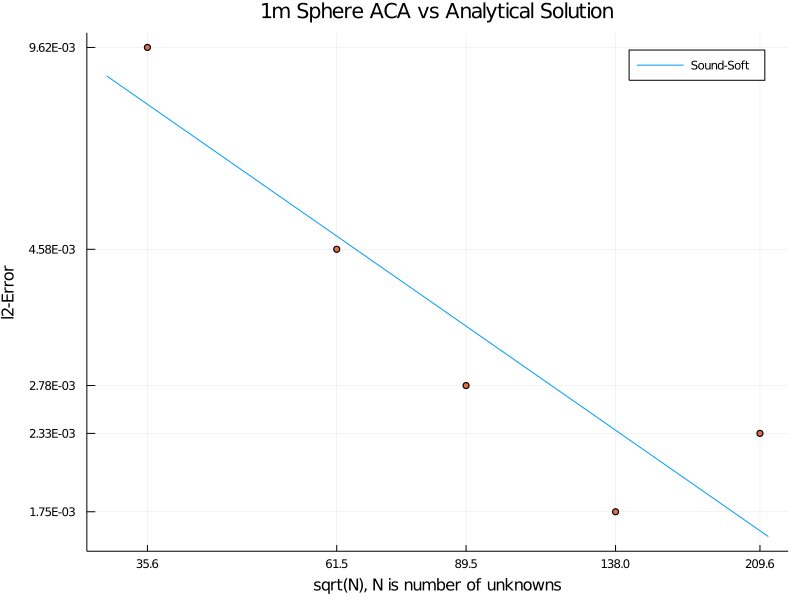
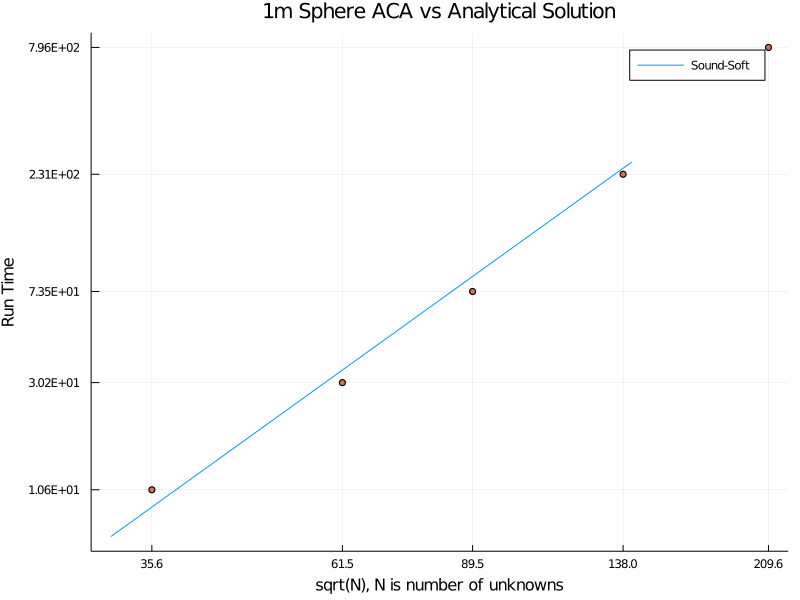
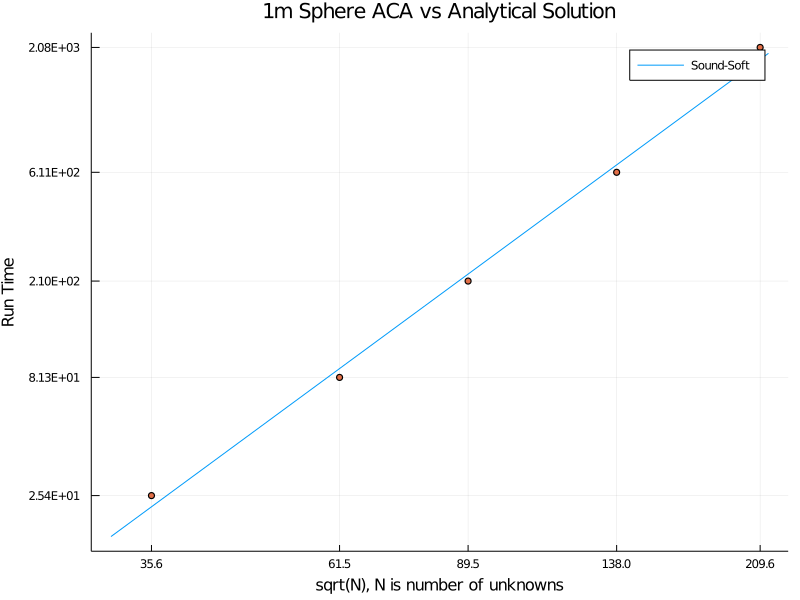
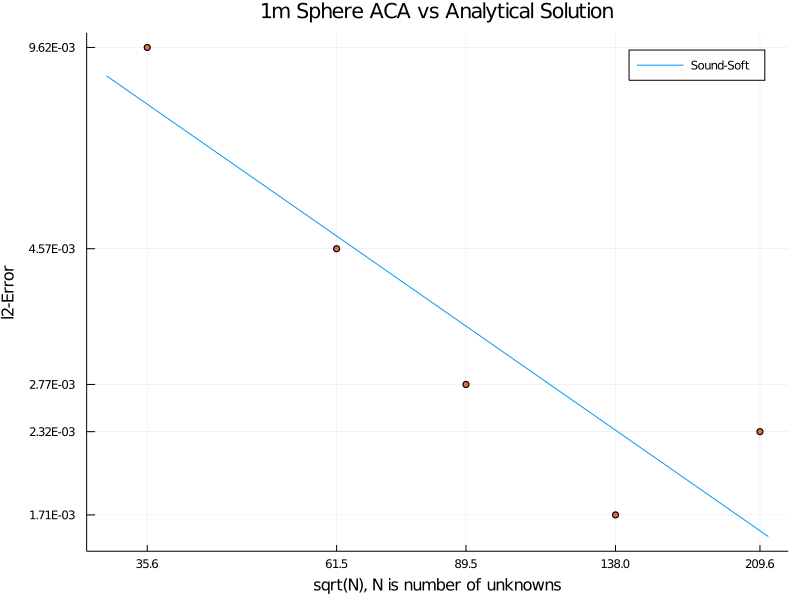
Clearly shows a noticeable impact on run time when tuning the tolerance down. Since the decrease in error is quite small, I think : will, generally, be the best tolerance to use.

**Figure 2**: From left to right: and . The lower tolerance has a noticeable affect on run time.

### Octree Levels Testing

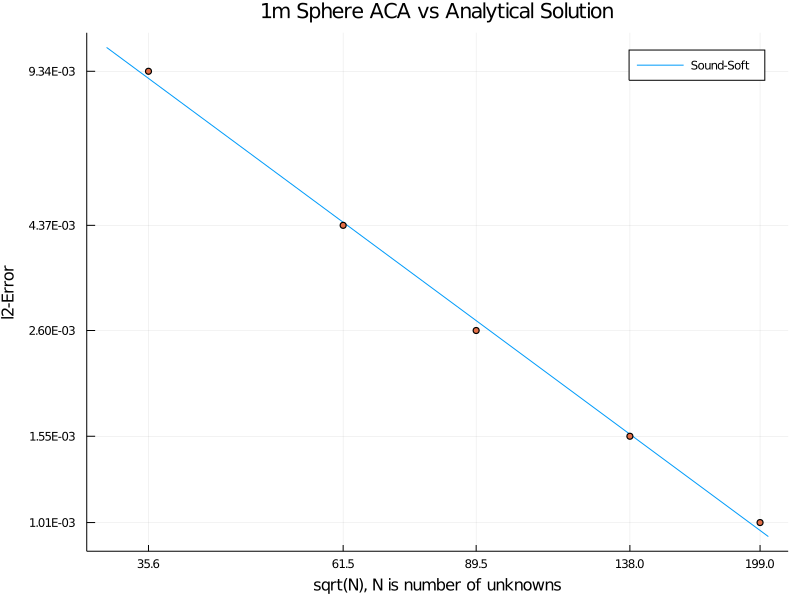
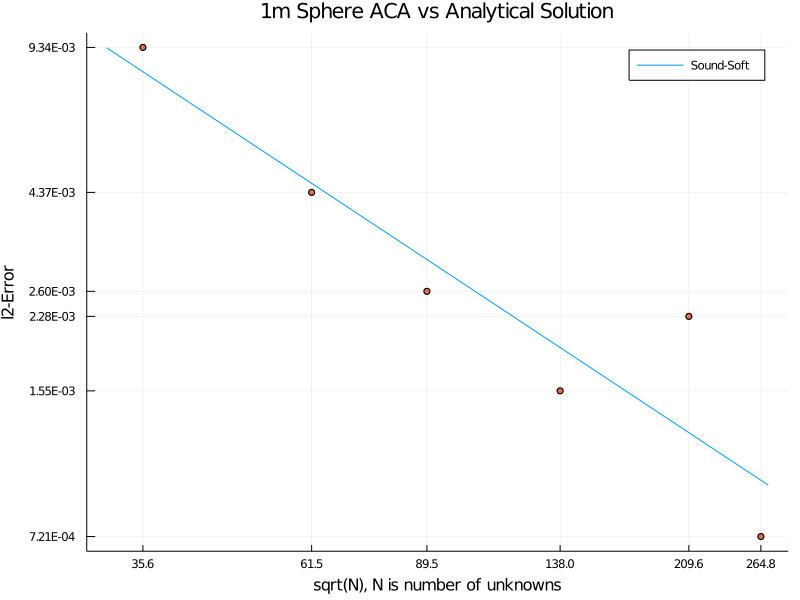
These tests run five meshes with 3, 4 and 5 octree levels to compare the effects of changing octree levels on error and runtime.

**Figure 3**: Error and runtime plots with 3 octree levels

These plots raise a concern about the error of the finest mesh. To figure out what is going on, I first ran another level of refinement and produced the following plot

**Figure 5**: Error and runtime plots with 5 octree levels

**Figure 4**: Error and runtime plots with 4 octree levels

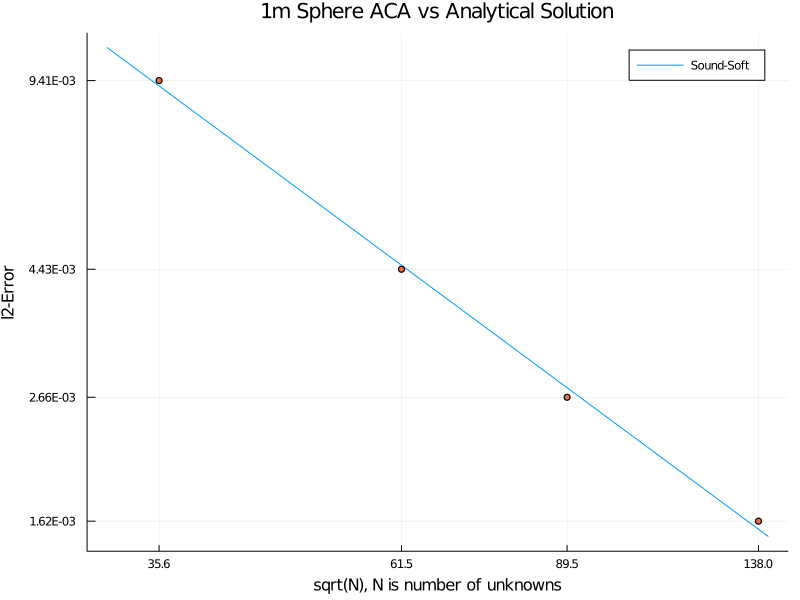
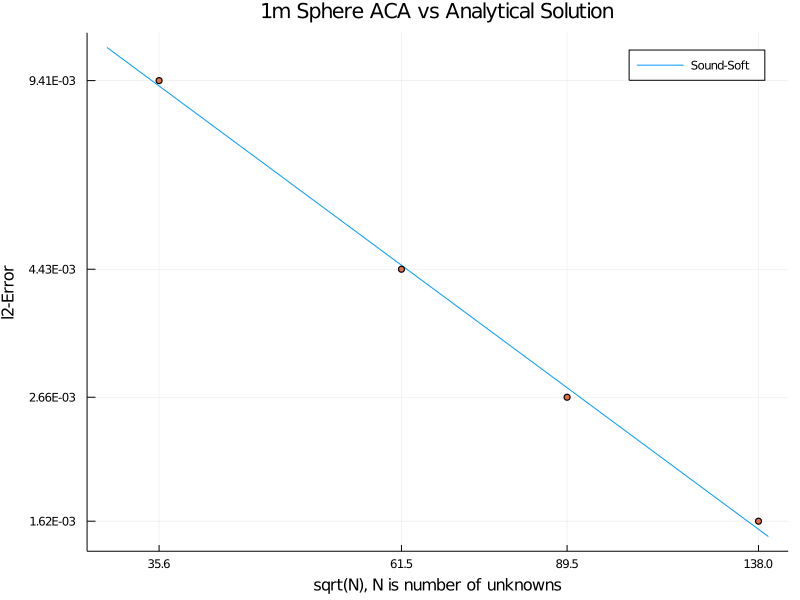
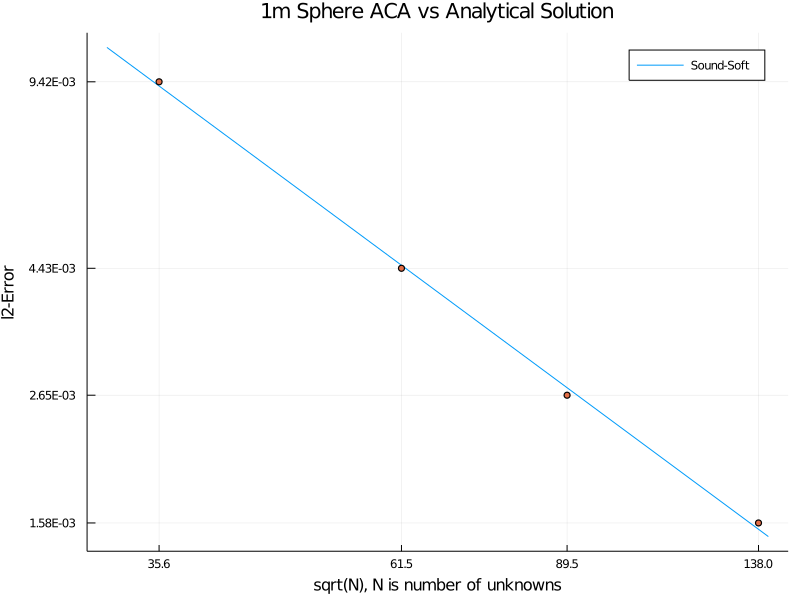
Figures 6 and 7 suggest the spike in error for the fifth mesh in Figure 5 is a fluke. Figure 6 shows that the convergence trend continues at the next level of refinement and Figure 7 shows that a different mesh with approximately the same number of elements () does not suffer from the same issue. This is sufficient evidence to me that the error spike comes from a poor mesh.

**Figure 7**: Error plot with 4 octree levels and a different 5th mesh

**Figure 6**: Error plot with 4 octree levels and a 6th mesh

### GMRES Tuning

I tested changing various GMRES parameters trying to decrease error further such as relative convergence tolerance and maximum number of inner iterations. This was probably a moot test because the ACA solver already has reached the errors obtained by the old solver and therefore I shouldn’t expect GMRES to be able to push things down any further.

While I am not sure what the relative tolerance of GMRES defaults to, it seems the changes in Figure 8 have no effect on error.

**Figure 9**: GMRES relative tolerance set to and ACA tolerance at

**Figure 8**: From left to right: GMRES relative tolerance set to and

For Figure 9, I cranked down the ACA tolerance and see little to no change in error. Lastly, I also doubled the max number of inner iterations GMRES can take and saw no changes to error. I think the default GMRES settings are good enough because other sources of error are dominating such as discretization and other algorithmic errors not including ACA because these errors are very close to the errors when using my traditional solver.