**Interactive Orbitals Simulation Description**

*This project was fuelled by two students’ determination to create a computer simulation of orbitals that was interactive, easy-to-use and could be employed as a teaching tool by their Chemistry professor.*

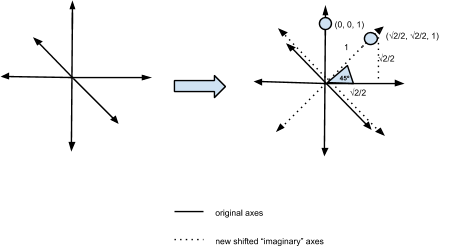
We can never know for sure where an electron will find itself at a given moment. This is why scientists have come up with atomic orbitals, three-dimensional maps of the probability distribution of the location of electrons in atoms. In other words, they are shapes that show where an electron is most likely to be found in the space around the nucleus. Schrodinger’s equations can be used to create these shapes or their wave functions but since they were too complicated for our mathematical understanding we approximated them using **quadric surfaces**. After doing some research (the [mathispower4u](https://www.youtube.com/watch?v=x6c2DdOrkQI&index=9&list=LLttK84oFlWDpBLrTEe5P5gA) series on the subject was especially useful) we decided to combine several different quadric surfaces to create the orbitals. Using grapher, a free graphing utility on Mac, we came up with the equations.

Refer to separate document for equations.

Refer to grapher files for first ‘draft’ of equations in action.

For example, for the 1s orbital, which is close to being a sphere, we simply used the equation for a sphere. For a 2p orbital, on the other hand, we had to combine a two sheet hyperboloid and two ellipsoids. We had to use inequalities to restrict the domains of the equations in order for them to fit together.

A complicated problem arose once we got to the 3d orbitals - until then the work had been a matter of testing and figuring out the basics. The problem with 3 out of the 5 3d orbitals is that the shapes are not aligned with the axes but rather at 45º with respect to them (forming an X). In order to resolve this problem we got help from a Mathematics teacher at our school. We used a rotation matrix to get our new equations. The main line of thinking behind this procedure is that we wanted to shift our “imaginary axes” by 45º.



The coding process was quite complex as we had to use tools that we had never utilized before. Using the matplotlib library in Python we digitized the graphs. However, we were not able to successfully code the three 3d orbitals that required the transformation in the time that we had for the project (they will be coming soon). We then bought this domain, interactiveorbitals.com, and we used the mpld3 library to put the Python code online.