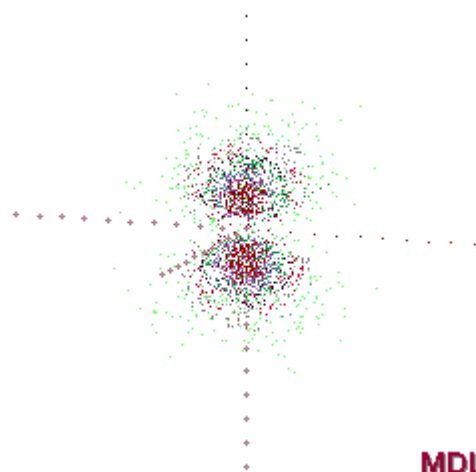


ORBITAL.EXE is a Visual Basic 4.0 program that runs under Microsoft Windows 95 and later versions. It allows the production of probability-based three-dimensional representations of the atomic orbitals of the hydrogen atom and other single-electron species.

**MDL**

Click on the orbital to view it using
Chime

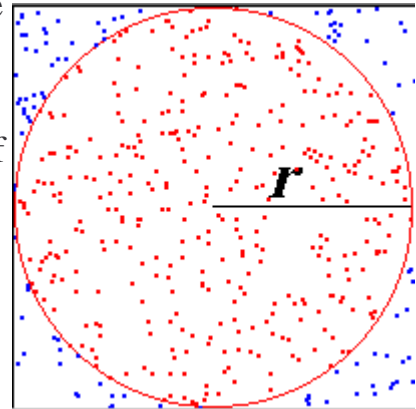
Unlike traditional representations of orbitals, which depict a surface containing a fixed percentage of the electron density, the orbitals created by ORBITAL.EXE depict the electron density itself, using dots of various spatial density and color. The orbitals are produced using a Monte Carlo technique and saved as Chime XYZ files. These files are suitable for uploading to web sites and are displayed automatically using the default browser, provided it has the **Chime plug-in** installed.

The Monte Carlo Method

ORBITAL.EXE uses the "Monte Carlo" technique to display a function that has different values in three-dimensional space. In this technique, named after a city famous for high-stakes gambling, a mathematical function is explored using probability. The Monte Carlo process is much like throwing darts at a dart board. If the dart is within the target area, it is registered as a "hit." By throwing many darts, and registering whether a hit was made or not, we can depict a function without actually drawing its curve.

The classic example of the use of the Monte Carlo method is the determination of the value of π by throwing darts at a dart board consisting of a circle contained in a square.

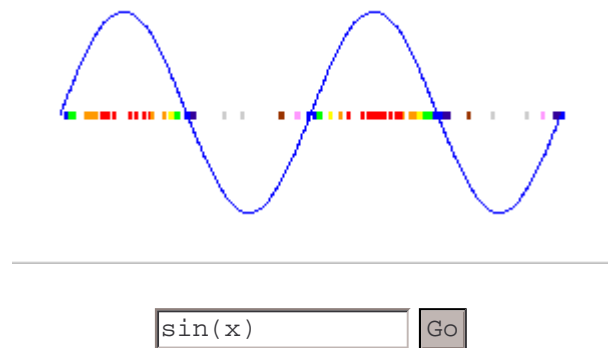
If $r = 1$, then the area of the square is 4, and the area of the circle should be $\pi r^2 = \pi$. The fraction of darts landing inside the circle (as thrown by a blindfolded player) should therefore be $\pi/4$, and from this fraction the value of π can be approximated.



Click [here](#) to watch π being calculated using the Monte Carlo method. Notice that the solution is not exact, but gains precision as more darts are thrown.

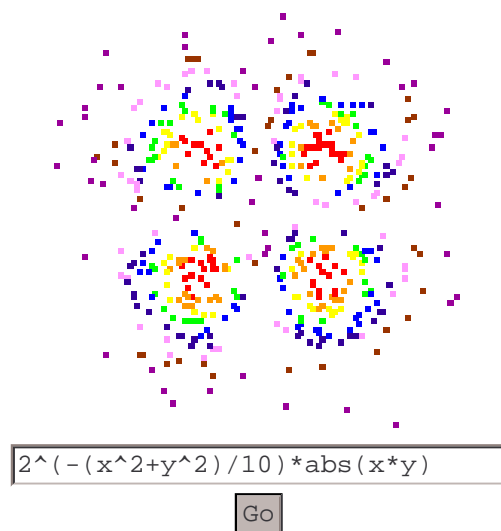
Now consider a function of x such as shown here. In this case, the Monte Carlo method can be used to depict the two-dimensional curve using only one dimension.

Click on "Go" to see this graph produced. (If you like, you can try different functions [here](#).)



The idea is to throw the darts and only register hits when the dart lands below the value of the function at that coordinate. Thus, the value of the function itself determines the probability of a point being registered at a given location. (This is similar to registering darts as hitting inside the circle in the first example.) The higher the function is in a given region, the more darts will be hits. Since more darts will be hits in regions where the function is higher, the density of dots will be higher wherever the function is higher. In addition, we can use the value of the function itself to color the dots. These two pieces of information--**density** and **color**--reinforce each other in the final picture.

In the above example, the curve itself shows the function just fine. That's because we had a function of x alone, $f(x)$. When the function is of two variables, $f(x,y)$, then it gets trickier to visualize.



Shown here is a function that depends upon both x and y plotted using the Monte Carlo method. The pattern resembles something like a cloverleaf.

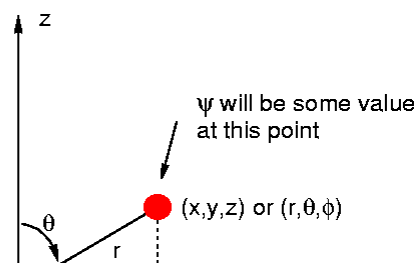
Click on "Go" to see this function or a function of your choice developed using the Monte Carlo method.

Thus, normally, if we want to depict a function of x we need two dimensions--one for x and one for $f(x)$. Likewise, for a function of two variables, x and y , we need three dimensions: one for x , one for y , and one for $f(x,y)$. As shown above, the Monte Carlo method of displaying functions allows us to reduce the number of dimensions by one. The cloverleaf graph above is basically a colorized "topological" map, like the ones used for showing terrain, where $f(x,y)$ is altitude above sea level.

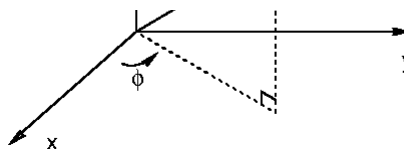
While functions that depend upon only two dimensions can be depicted using topographical maps, a real problem arises when we try to depict a function that takes different values in three-dimensional space. One solution is to depict "slices" through the function, as is done in magnetic resonance imaging (MRI). Another solution is to show only the *surface* where the function goes through a certain value, such as 5% of its maximum height. This is the method commonly used for displaying atomic orbitals. The problem with these solutions, of course, is that both depict the function only along a specified plane or surface. They give some idea of what the function looks like without showing its true four-dimensional characteristics.

The Schroedinger Equation for the Hydrogen Atom

The **solution to the Schroedinger equation for hydrogen** is a *wave function*, Ψ , that is exactly the sort of function that is difficult to depict. It depends on *three* variables, x ,



y , and z (or, as it is typically written using spherical coordinates, on r , θ , and ϕ).



It is the square of the wave function, Ψ^2 , that relates to the probability of finding the electron at a specific location relative to the nucleus. Ψ^2 is often referred to as the *orbital*. If we want to "see" an orbital, we need four dimensions, because Ψ has a different value at every point in three-dimensional space. With the Monte Carlo method, we can use just three dimensions plus dot color and dot density.

Visualizing the solution to the Schroedinger equation using the Monte Carlo method requires the placement of colored dots at points in three-dimensional space. This is where **Chime** comes in, because Chime allows just such a representation (usually in the context of molecules). As a bonus, Chime allows us to rotate the visualization around to see it from different perspectives.

The trick that ORBITAL.EXE uses is to "fool" Chime into thinking it is displaying a huge molecule consisting of "atoms" of different type (which are thus displayed using different colors). For example, oxygen is displayed by Chime as **red** and nitrogen is displayed as **blue**. By judicious use of atoms, we can represent all of the colors of the rainbow using Chime.



Calculation Options

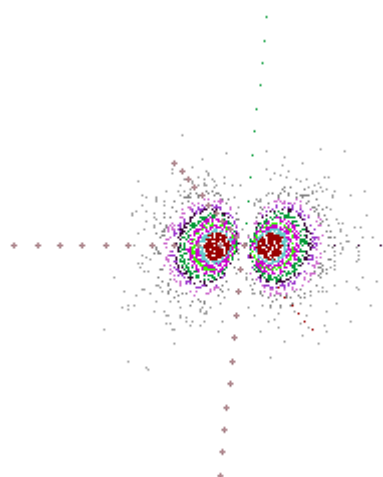
You will need to specify the quantum numbers n , l , and m before the calculation starts. Not all possible combinations are valid, because the solution to the Schroedinger equation imposes certain restrictions on the values of n , l , and m . Clicking on **J** or **aJ** displays the energy of the orbital in Joules or attoJoules (10^{-18} Joules), respectively.

- Calculation Options for the H Atom [Z=1] _____

n	<input type="text" value="3"/>	<input type="button" value="←"/>	<input type="button" value="→"/>	region:	<input checked="" type="radio"/> No Restrictions
l	<input type="text" value="1"/>	<input type="button" value="←"/>	<input type="button" value="→"/>		<input type="radio"/> XY-plane only
m	<input type="text" value="0"/>	<input type="button" value="←"/>	<input type="button" value="→"/>		<input type="radio"/> YZ-plane only
				energy:	<input type="radio"/> XZ-plane only
				maximum number of points:	<input type="text" value="10000"/>

maximum number of points: 10000

orbital extent (Angstroms): 13

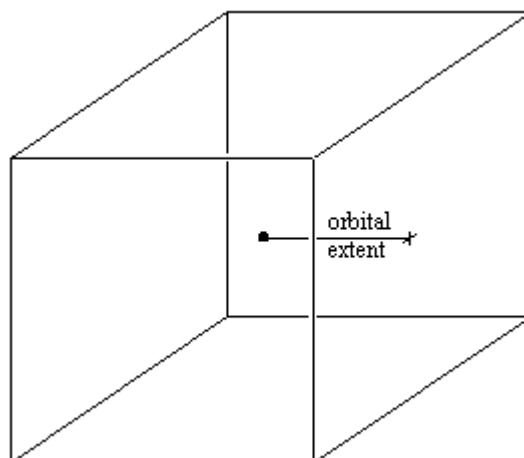


In addition, you can choose whether to restrict the points to specific planes. This produces a slice through the orbital.

[Click to view this slice using Chime](#)

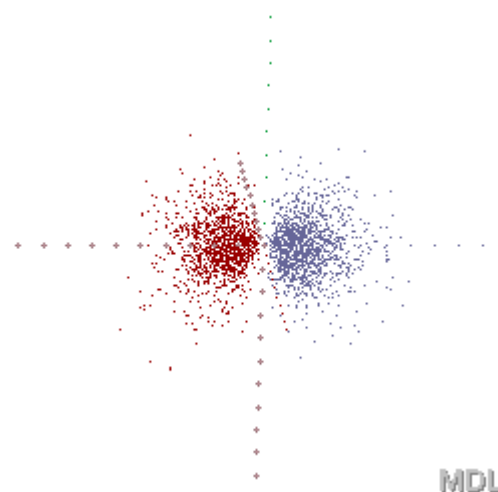
Also select the maximum number points to display before automatically pausing. (You can then continue with more points if you like.) The orbital shown here involves 2500 points. For smaller orbitals, 2500-10000 points may be sufficient, but for larger orbitals, you may want to select a higher number.

A critical parameter is the *orbital extent*, which is defined here as the distance from the center of the atom to the faces of a cube that defines the volume of space to investigate. Matching the orbital extent parameter to an orbital is important. If it is too large, then the process will be slow; if too small, the orbital will be clipped.



Color Options

The orbitals shown above employ color along with the density of dots to show the probability of finding an electron at a particular location around the hydrogen atom. Alternatively, you can display the orbital in a way that emphasizes the nodes for the wave function (where the value of Ψ equals 0). This particular case has one angular (planar) node.

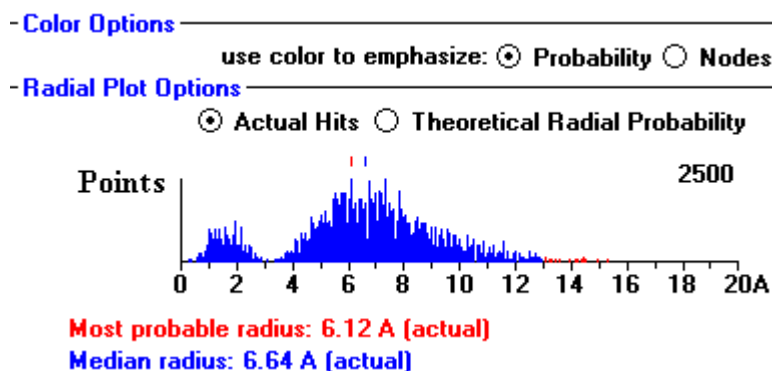


MDL

[Click to view the nodes using Chime](#)

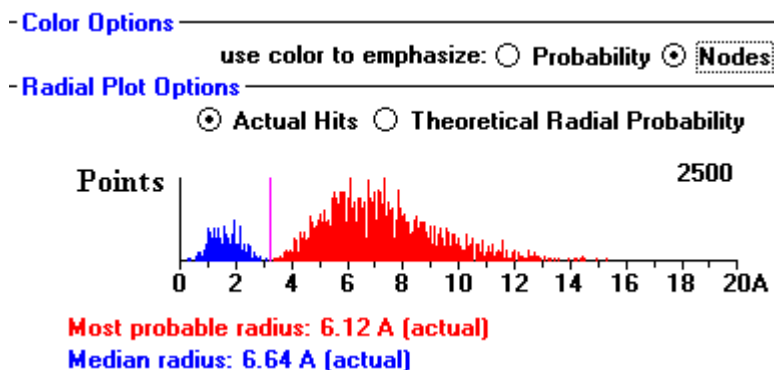
Radial Plot Options

Every 100 points, a bar graph is produced showing the number of points found (relative probability) vs. radius in Angstroms.



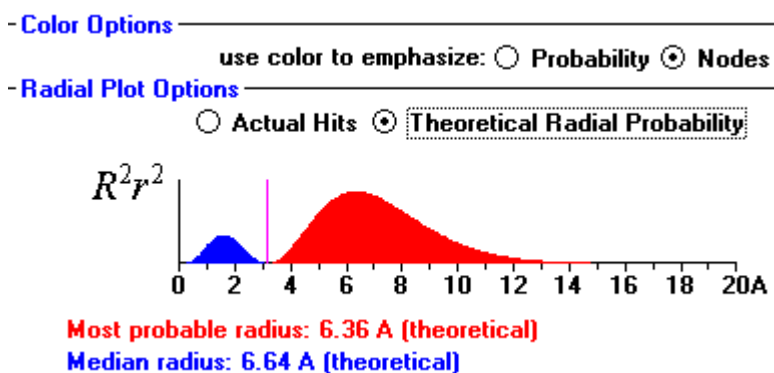
Two short vertical lines are dropped--one at the **most probable radius** (the highest bar) and one at the **median radius** (the radius at which half of the points are within). Red bars in the tail of the graph indicate points that were found in the corners of the cube. If there are many red points, then the orbital extent cube is too small.

If color is being used to indicate nodes, then a **violet-colored vertical line** is drawn between changes in the sign of the radial component of the wave function.



In this case, blue indicates where the wave function is positive; red indicates where it is negative.

An additional display option (for the radial plot only) depicts the theoretical radial probability distribution plot for the wavefunction. For a three-dimensional plot, this is defined as R^2r^2 , where R is the radial part of the wave function. Multiplication by r^2 takes into account the fact that at a certain radius there is a shell with area proportional to the square of the radius.

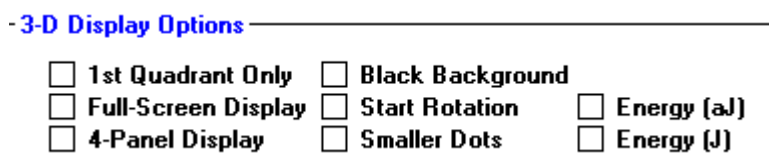


The actual hits and the theoretical radial probability should be quite similar, because the probability of a randomly chosen point to be at a certain radius from the nucleus is also proportional to r^2 .

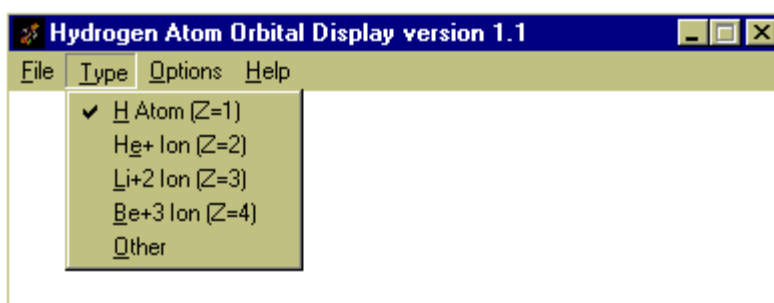
For a two-dimensional slice through an orbital, the theoretical radial probability is defined as R^2r . In the two-dimensional case, the probability is proportional to r instead of r^2 , because at a certain radius there is an arc with length directly proportional to the radius. Again, the actual hits and the theoretical radial probability should match reasonably well.

3-D Display Options

Several options allow you to depict the three-dimensional orbital different ways. Feel free to experiment with these options for the result you like best.

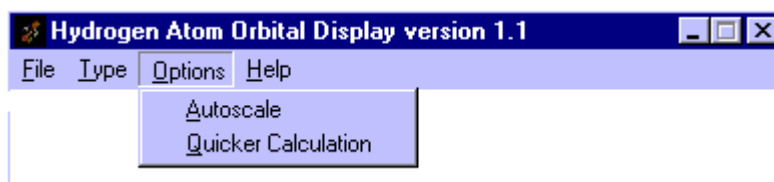


Other Single-Electron Species



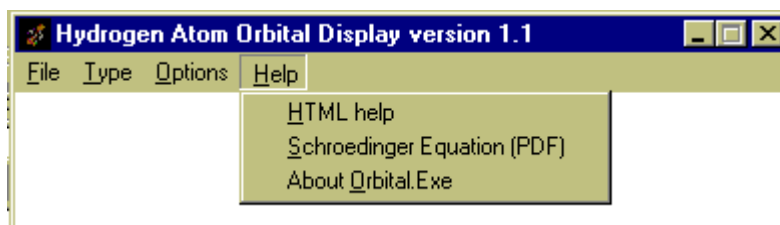
You can use ORBITAL.EXE to depict other single-electron species, such as He^+ ($Z=2$), Li^{+2} ($Z=3$), Be^{+3} ($Z=4$), and B^{+4} ($Z=5$). Simply select the species of interest using the **Type** menu selection.

Other Options



Two additional options include the "autoscale" option and the "quick calculation" option. If **Autoscale** is checked, then the orbital extent is automatically set when the value of n , l , and m are changed. The **Quick Calculation** option allows the Monte Carlo calculation to run faster, especially for s orbitals. If this option is checked, then the resulting orbital may not quite match the theoretical values in terms of dot density and radial probability. However, colors and node positions will still be correct.

Help Available



From the help menu you can access this this document or, if you have a default PDF file reader (such as Adobe Acrobat) installed, you can view [the complete solution to the Schroedinger equation](#). This document also includes extensive discussion of the solution.

Files Created by ORBITAL.EXE

ORBITAL.EXE automatically creates files as it works. All created files are placed in the subdirectory `C:\orbital\files`. The names of these files start with the values of n , l , and m . Three types of files are created:

- **Chime coordinate files (*.xyz).** These files contain the "atoms" used to depict the points in space where the Monte Carlo "darts" landed. Colors (atom types) are based on the value of Ψ^2 .
- **Hypertext "wrapper" files (*.htm).** These files specify the way the Chime orbital should be displayed.
- **Comma-Separated Value Files (*.csv).** These files save information for later quick loading using the "Load" button. They consist of an initial line indicating the number of points, the orbital extent (in Angstroms) and the maximum value for the square of the wave function. Each additional line consists of the x, y, and z coordinates of the point and the value of the wave function at that position, separated by commas.

If you move files to a web site, be sure to transfer both the *.xyz files and the *.htm files. The *.csv files are not necessary except for quick loading of an orbital back into ORBITAL.EXE.

Batch Orbital Generation

If you run ORBITAL.EXE with a filename on the command line, then the program will carry out the sequence of calculations indicated in that file. In this way, many

orbitals can be generated at once. For more information about this option, see the sample file [input.txt](#).

Installation of ORBITAL.EXE

ORBITAL.EXE requires the Windows operating system, version 95 or better. In addition, the default browser on the system must have the Chime plug-in installed. (See <http://www.mdli.com>.) Instructions for the installation of ORBITAL.EXE are given in [install.txt](#). The program, along with this HTML help file, is available as a zip file, [orbital1.zip](#).

About ORBITAL.EXE

ORBITAL.EXE was written by Bob Hanson (hansonr@stolaf.edu and <http://www.stolaf.edu/people/hansonr>) at [St. Olaf College](#) using Visual Basic 4.0. Any comments or suggestions for program improvement are welcome.

Context for Using ORBITAL.EXE

At St. Olaf College, ORBITAL.EXE has been used by first-year students both in [Chemistry 123](#), *Atomic and Molecular Structure* (a January interim course), and [Chemistry 126](#), *Energies and Rates of Chemical Reactions* (a Spring semester course).

Not all first-year chemistry students take Chemistry 123, which is not a laboratory course. In this case, ORBITAL.EXE is used by students on a day when the class meets in a computer laboratory rather than in a classroom. Students explore the various shapes and orientations of orbitals and learn about the concept of a node and its relationship to the quantum numbers n , l , and m .

Chemistry 126, which is a laboratory course, is somewhat unusual for a first-year course in that for the first seven weeks students use a nonstandard textbook, [Introduction to Molecular Thermodynamics](#). This book uses an accessible, probabilistic approach to develop ideas of quantum states, energy levels, entropy, enthalpy, and free energy.

In Chemistry 126, students use ORBITAL.EXE as part B of their [second laboratory experience](#), *Energy Levels and Spectra: Atomic Spectroscopy*. In Part A of this laboratory, students investigate the emission spectrum of hydrogen experimentally. In Part B, students experiment with different values of n , l , and m , exploring their

relation to the number of radial and spherical nodes, similarly to what students in Chemistry 123 do. However, in this case, students are asked to look through the solution of the Schroedinger equation for places where the "rules" for n , l , and m originate. At the end of the experiment, students answer the following questions:

1. Describe how the lines of the hydrogen emission spectrum you observed arise from hydrogen atoms falling from one state to another. Why can't we see the transitions from level 5 to level 1 or from level 6 to level 3?
2. Compare your determination of the ionization energy of the hydrogen atom, R_H , with that given in your text.
3. Using your value for R_H , predict the wavelength of the fifth Balmer line (level 7 to level 2).
4. How would you describe the "meaning" of n ? of l ? of m ?
5. Where exactly in the solution to the Schroedinger equation would you find the "rules" for n , l , and m ?
6. What is the relationship between the number of nodes (total and of each type) and the quantum numbers n , l , and m , and how is this supported by your observations?
7. Describe in your own words precisely what it would mean for the hydrogen atom to drop from the $n = 2$ level to the $n = 1$ level in terms of probability. (For example, would the electron suddenly jump from one place to another? Or would it be more subtle than that?)

This help file uses divgraph.js for its real-time graphics. divgraph.js was developed at St. Olaf College by Robert M. Hanson (<http://www.stolaf.edu/people/hansonr/divgraph>)