Computer Bulletin Board

edited by
Steven D. Gammon
University of Idaho
Moscow, ID 83844

Orbital Plots Using Gnuplot

Brian G. Moore
School of Science, Penn State Erie–The Behrend College, Erie, PA 16563-0203; *bgm4@psu.edu*

About Gnuplot

Various means of displaying the one-electron hydrogen atom orbitals have been discussed in this *Journal* (*1*–*8*). In the process of exploring this topic with a physical chemistry class at Behrend, we looked at possible ways of creating illustrative views of the orbital functions. The technique we finally arrived at makes use of a free and ubiquitous general-purpose plotting program called *Gnuplot*. Since Gnuplot is free, available on a variety of platforms, and easy to use, perhaps using it to create orbital views might be of use to some readers of this *Journal*.

In our case the big advantage of using Gnuplot was that it was already installed on all the computer systems in our lab, which use Linux as an operating system. In addition, the simple syntax for specifying a plot in Gnuplot means that writing a plotting file for an orbital is simply a matter of transcribing the analytical form of the function to be plotted. Gnuplot is a fairly small, mature, general-purpose plotting program. Another advantage of using Gnuplot to display the orbitals is that the skills learned to master the program may be useful for other purposes, such as displaying data from a lab report.

One problem of moving to a new academic or industry position is encountering new computer hardware and operating systems. In my career, I forced myself to learn several different plotting programs to accommodate each move before I finally settled on Gnuplot as a good free product that works well in several environments. Gnuplot is a very widespread program. It is part of the standard operating system package in most UNIX installations. For example, Gnuplot is part of many commonly available Linux packages and is very easily installed on any Linux version. Versions of Gnuplot exist for many other operating systems as well, including PC/DOS, PC/Windows(3.1), Windows 95/98/2000 and NT, VAX/VMS, to name a few. I have personally used Gnuplot on these platforms with minimal problems. There is also a version for the Macintosh, which I have not tried.

Gnuplot is copyrighted but freely distributed. A central site with much of the crucial information about Gnuplot as well as binaries, a reference manual, example plotting files, and much more is at *http://www.cs.dartmouth.edu/gnuplot_info.html*.[1] At the Dartmouth site, the binaries are currently at *ftp://ftp.dartmouth.edu/pub/gnuplot/*. To get a version of Gnuplot that will work for Windows 95/98/2000 and NT, download the files *gnuplot3.7cyg.zip* and *gnuplot3.7cyg.zip.sig*. These files will get you version 3.7 of Gnuplot, which is the one I used to create the plots in this paper (under the Linux operating system).

When Gnuplot is started, a command prompt appears. (In the Windows version, a window appears with a command prompt inside it). Plotting commands may be entered interactively or from previously saved text files. A standard test of Gnuplot functionality is to enter a simple plot command such as `plot sin(x)`. If Gnuplot is installed properly and functioning, a plot of the sine function should appear.

For complicated plots, the best way to proceed is to save a series of plot commands in a text file. The command file may be edited with any text editor (the vi or emacs editor on a UNIX system, Notepad on a PC/Windows system, etc.). For example, if we have saved a text file of Gnuplot commands named `plotfig.gnu`, then at the Gnuplot command prompt we would type the command `load "plotfig.gnu"`. (The quotes are required. Filenames are specified in quotes in Gnuplot.)

The specifics of the appearance of the plots produced by Gnuplot are dependent on the display device. To achieve a consistency of output format in a reasonably portable form, I have chosen for this paper to use Gnuplot's postscript driver. (This is the way we use it in our physical chemistry class.) Previewers and printing software for postscript are present on most UNIX/Linux systems. These kinds of programs are also available for MS-Windows operating systems. See *http://www.cs.wisc.edu/~ghost* for more information on the free programs ghostscript and GSview. If you don't have a postscript viewer, you can display to the screen (remove the commands `set term postscript` and `set output "filename"` in my example command files) and in the MS-Windows version of Gnuplot you can save the plot to the clipboard (right click on the display of the graph to bring up the menu), then paste into another application such as Word for printing.

One nice feature of the free availability of Gnuplot is that it can then be run through a Web page interface (*9*). I have set up a Web page that shows this capability; it is located at *http://onsager.bd.psu.edu/~moore/orbitals_gnuplot*. This page includes links to forms that will execute all the examples given in this article, as well as additional plotting examples. The forms may be edited before submission, allowing the user to see the effects of changing various parameters. Upon submission, a CGI script runs the Gnuplot program and the output is converted to a GIF file and displayed in the browser window. (For extended work with Gnuplot it will be easier to download the free program and run it on your own computer.) I have included a listing of the code for the CGI script program as a reference; the script is written for a Linux server.

Two-Dimensional Plots

For a very simple first example, let's use the radial distribution function for an electron in a hydrogen atom 3s orbital. Using the table of hydrogen-like wave functions from Pauling (*10*) we find

$$R_{30}(r) = \left(\frac{Z}{a_0}\right)^{3/2} \frac{1}{9\sqrt{3}} \left(6 - 6\rho + \rho^2\right) e^{-\rho/2} \qquad (1)$$

where $\rho = (2Z/na_0)r$, a_0 is the Bohr radius (0.529 Å), and n is the principal quantum number (3, in this case). Below I show a sequence of Gnuplot commands to plot $4\pi r^2(R_{30})^2$ vs r/a_0.

```
# File Rad3s.gnu
#   Creates an x-y plot of the radial probability density
#   4pi(r^2)R^2 as a function of r/a_0.
#
reset
set samples 400
set xlabel "r/a_0"
set ylabel "Radial distribution function"
set title "Radial probability, H-atom 3s wavefunction: "
set xrange [0:30]
# Note: using formulas from Pauling, Z=1, n=3;
# by setting a_0=1, resulting plot will have r/a_0
for x-axis.
Z=1.
n=3.
rho(x)=(2.*Z/n)*x
R30(x)=( 1./(9.*sqrt(3.)) )*(6. - 6.*rho(x) +
rho(x)**2)*\
exp(-rho(x)/2.)
P(x)=4*pi*(x**2)*R30(x)**2
set term postscript enhanced
set output "Rad3s.ps"
plot P(x) with lines lw 3
```

The plot produced by the above command file is shown in Figure 1. I have tried to write the plotting file in a way that closely parallels the analytical representation of the functions in question (eq 1). Note that many standard mathematical functions are provided and the user may define functions (such as `rho(x)`, etc., above). Comments are indicated by a # symbol at the beginning of a line. The backslash character (\) can be use to continue a command on the following line. By default, the function is generated on a grid of 100 points on the *x*-axis. A higher sampling produces a smoother plot. The default *x* range of [-10:10] is reset in the example above. The *y* range is automatically scaled.

Three-Dimensional Surface Plots

The three-dimensional surface plotting features of Gnuplot are especially good. An example of this is a plot of the angular variation of the probability density of an orbital. On this three-dimensional surface, the distance from the origin represents the probability density at that particular angle.

The angular part of the wave function is the product of two terms, one which depends on the polar angle θ, the other ϕ. Using again the notation of Pauling (*10*), choosing the case of a d_{yz} orbital we have

$$\Theta_{2\pm1}(\theta) = \frac{\sqrt{15}}{2}\sin\theta\cos\theta\,, \qquad \Phi_{1\sin}(\phi) = \frac{1}{\sqrt{\pi}}\sin\phi \quad (2)$$

where $\Phi_{1\sin}(\phi)$ represents the real linear combination $(1/i\sqrt{2})(\Phi_1 - \Phi_{-1})$ (and $\Phi_m(\phi) = (1/i\sqrt{2\pi})e^{im\phi}$). The function we wish to plot is the probability density $(\Theta\Phi)^2$ as a function of the angles θ and ϕ. In Gnuplot a 3-D surface is most easily defined parametrically. In this mode, the surface plotting command (`splot`) requires three arguments, which are the three parametric functions giving the *x*, *y*, and *z* location of the surface. Thus we need to also include the spherical polar coordinate transformations,

$$x = r\sin\theta\cos\phi, \quad y = r\sin\theta\sin\phi, \quad z = r\cos\phi$$

The Gnuplot input file to plot this surface is given below.

```
# File dyz.gnu
#   Displays the d_{yz} orbital
#   Output to postscript file dyz.ps
#
set parametric
set isosamples 40,40
set hidden
set urange [0:pi]
set vrange [0.:2.*pi]
box=0.45
set zrange [-box:box]
set xrange [-box:box]
set yrange [-box:box]
set xlabel "x axis"
set ylabel "y axis"
set zlabel "z axis" -8,-8
set view 65,25,,2.5
# Define theta functions
theta20(x)=(sqrt(10.)/4.)*(3.*(cos(x)**2) - 1.)
theta21(x)=(sqrt(15.)/2.)*sin(x)*cos(x)
theta22(x)=(sqrt(15.)/4.)*(sin(x)**2)
# Define phi functions
phi0(x)=1./sqrt(2.*pi)
phi1cos(x)=(1./sqrt(pi))*cos(x)
phi1sin(x)=(1./sqrt(pi))*sin(x)
phi2cos(x)=(1./sqrt(pi))*cos(2.*x)
phi2sin(x)=(1./sqrt(pi))*sin(2.*x)
# Define orbital functions
```

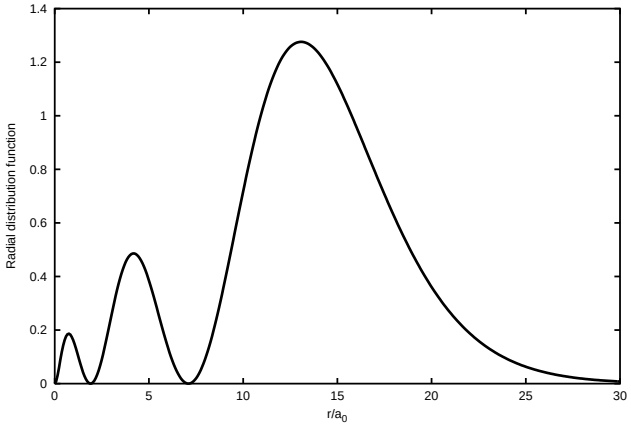

Figure 1. Radial distribution function for an electron in a hydrogen 3s orbital.

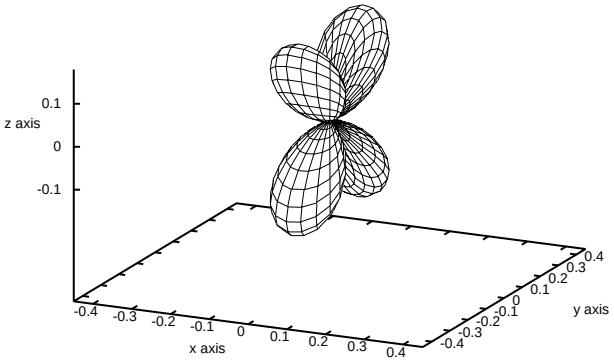

Figure 2. Probability density surface as a function of angle for a hydrogen atom d_{yz} orbital. At each angle θ, ϕ—the distance of the surface from the origin—is proportional to the probability density at that angle.

```
dz2(x,y)=theta20(x)*phi0(y)
dxz(x,y)=theta21(x)*phi1cos(y)
dyz(x,y)=theta21(x)*phi1sin(y)
dxy(x,y)=theta22(x)*phi2cos(y)
dx2y2(x,y)=theta22(x)*phi2sin(y)
# Set output to postscript file
set output "dyz.ps"
set term postscript enhanced
# Create the plot
splot (dyz(u,v)**2)*sin(u)*cos(v)\
,(dyz(u,v)**2)*sin(u)*sin(v)\
,(dyz(u,v)**2)*cos(u) title "d_{yz} orbital"
```

The plot produced by the above Gnuplot input file is shown in Figure 2. The command file shown defines *all* the d orbitals. To plot one of the other orbitals simply substitute the desired orbital function (e.g., `dz2`) for the one actually plotted (`dyz`) in the final splot command and change the name of the output postscript file.

The parameters such as the scales for the axes and the view settings were set by trial and error. In the set view command the first two arguments change the view angle and the last argument sets the scaling for the *z*-axis. For this plot, the value of 2.5 was arrived at by making a surface plot of a sphere at the same view angles and viewing the surface at various values of the scale parameter until a distortion-free spherical surface appeared. The particular value of this parameter is dependent on the display device (in this case the Gnuplot "postscript" driver). Note that again the actual specification of the functions closely parallels the analytical representation of the orbital functions, the variables *u* and *v* taking the role of θ and ϕ.

Contour Plots

A surface plot such as that produced in Figure 2 is good for visualizing the angular dependence, but it does not include any information about the radial variation of the probability density. Gnuplot has the capability of producing contour plots that can be used to present this information. To do this we take a two-dimensional slice of an orbital and plot the probability density; Gnuplot can automatically generate contours on such a surface.

Let's take the example of a $3d_{xy}$ orbital. The functions in question are

$$\Theta_{2\pm2}(\theta) = \frac{\sqrt{15}}{4}\sin\theta\,, \qquad \Phi_{2\cos}(\phi) = \frac{1}{\sqrt{\pi}}\cos 2\phi \qquad (3)$$

and

$$R_{32}(r) = \left(\frac{Z}{a_0}\right)^{3/2}\frac{1}{9\sqrt{30}}\rho^2 e^{-\rho/2} \qquad (4)$$

with ρ as defined in eq 1. We will take the plane $z = 0$ and plot the probability density in the xy plane. To do this we set $\theta = \pi/2$ and parametrically plot the probability density as a function of *r* and ϕ. The corresponding Gnuplot input file is given below.

```
# File 3dxy_cont.gnu
#   Displays a contour plot and a surface of the
#   probability density of a 3d_{xy} orbital
#   in the xy plane (theta = pi/2.).
#   Output to postscript file 3dxy_cont.ps
#
set parametric
set contour
set isosamples 25,50
set cntrparam levels 5
set cntrparam levels incremental 1.0e-04,1.0e-04
box=15.
set urange[0.:17.5]
set vrange[0.:2*pi]
set xrange [-box:box]
set yrange [-box:box]
set xlabel "x axis"
set ylabel "y axis"
set view 70,30,,
# Define theta functions
theta22(x)=(sqrt(15.)/4.)*(sin(x)**2)
# Define phi functions
phi2sin(x)=(1./sqrt(pi))*sin(2.*x)
# Define orbital functions
dxy(x,y)=theta22(x)*phi2sin(y)
dxy_cont(x)=dxy(pi/2.,x)
# Define radial functions
n=3.
rho(x)=(2./n)*x
R32(x)=(1./(9.*sqrt(30.)))*(rho(x)**2)*exp(-rho(x)/2.)
# Define the plotting function
orbital(x,y)=R32(x)*dxy_cont(y)
# Set output to postscript file
set output "3dxy_cont.ps"
set term postscript enhanced
# Create the plot
splot u*cos(v),u*sin(v),(orbital(u,v)**2) title "3d_{xy}"
```

The plot generated by this input file is shown in Figure 3. Note the contours shown in the base of the plot generated by the "set contour" command.

The contours of a surface may be presented alone by removing the surface and changing the view angle. The internal radial nodes of orbitals may be visualized in this way. An example is the $3p_z$ orbital.

```
set contour
set nosurface
set key -27,10
set cntrparam levels 5
set cntrparam levels incremental 2.e-5,2.e-5
set isosamples 50
set title "3px orbital – z=0 plane"
set xrange [-20:20]
set yrange [-20:20]
set xlabel "x axis"
set ylabel "y axis"
set view 0,0,1.15,
set size 0.62,1
rfun(x,y)=sqrt(x**2 + y**2)
phifun(x,y)=atan(y/x)
```

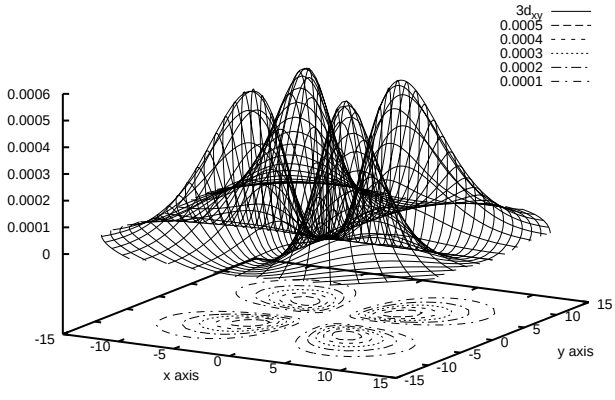

Figure 3. A plot of the probability density along the z = 0 surface for a hydrogen atom $3d_{xy}$ orbital, showing the contour plotting capability of Gnuplot.

```
theta11=(sqrt(3.)/2.)
phi1cos(x,y)=(1./sqrt(pi))*cos(phifun(x,y))
rho(x,y)=(2./3.)*rfun(x,y)
Rad(x,y)=(1./(9.*sqrt(6.)))*(4.                -
rho(x,y))*(rho(x,y))*exp(-rho(x,y)/2.)
f(x,y)=(Rad(x,y)*theta11*phi1cos(x,y))**2
set term postscript enhanced
set output "3px_cont.ps"
splot f(x,y)
```

The plot is shown in Figure 4.

Use of Gnuplot in Physical Chemistry

In our undergraduate physical chemistry class, I try to create a sequence of Gnuplot exercises that lead the students to a better understanding of the underlying mathematics of quantum chemistry. We do a few simple plotting exercises during the first semester to acquaint the students with the program. Gnuplot can be a good tool for data plotting as well as mathematical visualization. During the second semester we study quantum chemistry, eventually reaching the detailed illustration of the mathematics of the single-electron orbitals. As we proceed through this, I assign relevant plotting exercises, for example related to the particle-in-a-box or harmonic oscillator models. I provide the students with example plotting files and example output and they progressively work toward more independent exercises.

Simple visualization of the orbital functions can be useful as an exercise in itself. We will take a plot of a familiar orbital—say one of the d set—and view it from different angles, with change in scales, in combination with other orbitals, etc., or we will examine the difference in shape seen when plotting the square of the orbital as opposed to the orbital function itself.

Once the students are comfortable with these skills, I give them a few more challenging final exercises that require more independent work. These exercises usually change from semester to semester; the goal is to try and visualize an orbital in a unique way or an orbital function that the student (or perhaps the instructor!) has not seen before.

This "capstone" exercise might be something as simple as rendering a view of a hybrid orbital. (The first semester that I tried these Gnuplot exercises I had not worked out plotting files for any hybrids.) In this case we had to decide on the best method for creating the plot. A surface plot of the probability density (similar to Fig. 2) provides a plausible image, very similar to the renderings seen in some texts. However, in this case, since two orbitals must be added together, the relative size of each orbital will affect the resultant shape. For this reason, the contour plot method is a more accurate depiction. An example is shown in Figure 5 for an sp^2 hybrid, in particular, the combination

$$\frac{1}{\sqrt{3}}2s + \frac{\sqrt{2}}{\sqrt{3}}2p_x \qquad (5)$$

Surface plots of the f and higher orbitals are a natural exercise to undertake. One can qualitatively predict the shapes of all these higher orbitals fairly easily, as described by Breneman in this *Journal* (*8*). In looking at the f orbitals one should also be aware of the linear combination described by Cotton (*11*), which is shown in some general chemistry books. Displayed in Figure 6 is one of the 8-lobed orbitals from the f set.

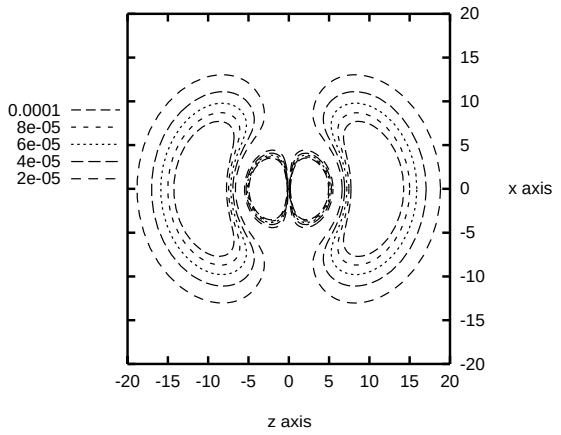

Figure 4. Contour plot of the $3p_z$ orbital.

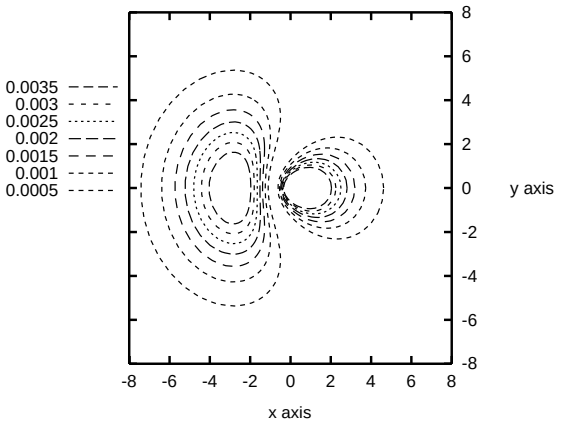

Figure 5. Contour plot of an sp^2 hybrid formed from 2s and 2p orbitals.

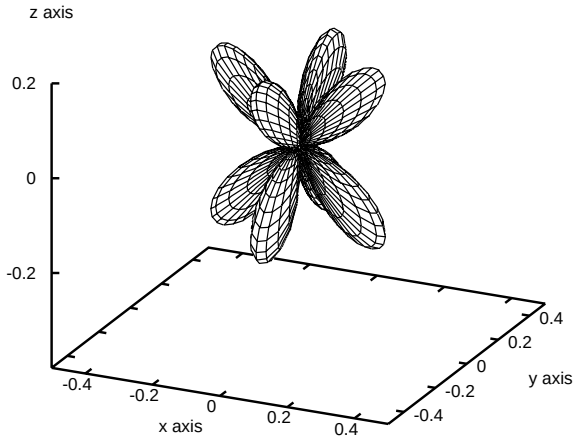

Figure 6. A surface plot of the angular part of the probability density for the $f_{z(x^2-y^2)}$ orbital.

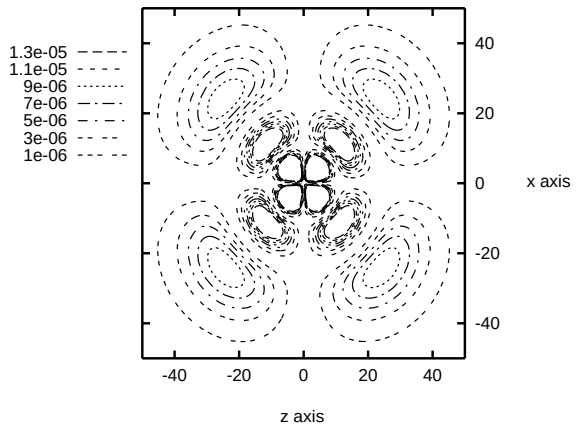

Figure 7. Contour plot of the $5d_{xz}$ orbital in the $y = 0$ plane.

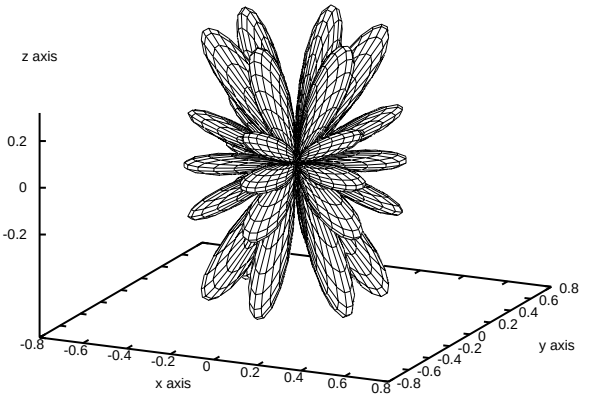

Figure 8. Surface plot of the angular variation of the absolute value of the orbital function for $\ell = 7$, $|m| = 3$ using the sine function of ϕ.

It can be interesting to look at a familiar orbital and vary the principal quantum number, viewing the internal radial nodes. This is shown in Figure 7. While the appearance agrees with one's expectation, I had never seen this particular view of this orbital.

In viewing the higher orbitals, a nice challenge for the student (and instructor) is "how high can we go?" While the very high ℓ orbitals may have little application to physical problems, the challenge of displaying them can be worthwhile as an exercise in the mathematics underlying the quantum mechanics. For example, the table of orbitals we have used for creating the plots (from Pauling's quantum mechanics book [*10*]) stops at $\ell = 5$. We can use recursion relations to obtain the orbital equations for any of the higher orbitals, for example

$$P_{\ell+1}^{|m|}(x) = \frac{2\ell+1}{\ell-|m|+1} x P_{\ell}^{|m|}(x) - \frac{\ell+|m|}{\ell-|m|+1} P_{\ell-1}^{|m|}(x) \quad (6)$$

a rearrangement of eq 19-16 in Pauling. Thus if one has the functions $P_4^3(x)$ and $P_5^3(x)$ (which only differ from the Θ functions listed in Pauling's table by a prefactor), the recursion relation may be used to get any higher associated Legendre polynomial with $|m| = 3$. In Figure 8 we see the surface plot one of the orbitals for $\ell = 7$ and $|m| = 3$; this is one of the *j* orbitals. By my count there are 30 angular lobes! When plotting surface views of the higher orbitals it is easier to display the absolute value of the orbital function itself rather than the square. The square of the wave function has more physical meaning, as it is the probability density for the electron; but the small lobes become much narrower (and hence require a much finer display grid) when the function is squared. If one is interested in just qualitative shapes of the orbitals—how many lobes, how are they arranged—plotting the orbital function rather than the square is easier.

In all work in the physical chemistry class that we do with Gnuplot, the emphasis is on reaching a better comprehension of the underlying quantum mechanical ideas and the related mathematical concepts. In this sense it can be an advantage that Gnuplot does not "do it all". I have seen software packages that will produce spectacular pictures of all the orbitals, but the procedure we undertake, which involves using a general-purpose plotting program as a tool to view complicated three-dimensional functions, helps to develop much more generalizable skills.

Acknowledgments

The orbital plots were developed with the participation of the spring 1997 and 1998 physical chemistry classes at Behrend: Afraa Al-Quraishi, Matt Blazevich, Shawn Burkholder, Colleen Gritzen, Keegen Guyer, Jennifer Herrmann, Anthony Jordan, Rebecca Mack, Shital Patel, Amy Pickwick, Christina Sayson, John Schwendeman, Tom Stevenson, Neil Vogeley, Toni Yezzi and Matthew Wilson.

Note

1. This information about the location of resources on the Internet can of course only be correct at the time of the writing of this article. For more up-to-date or other information about Gnuplot, readers might try a net search with keyword Gnuplot (I recently did such a search using the AltaVista search engine. It yielded hits in the tens of thousands).

Literature Cited

1. Cooper, R.; Casanova, J. *J. Chem. Educ*. **1991,** *68*, 487–488.
2. Ramachandran, B. *J. Chem. Educ.* **1995,** *72*, 1082–1083.
3. Rioux, F. *J. Chem. Educ.* **1992,** *69*, A240–A242.
4. Ramachandran, B.; Kong, P. C. *J. Chem. Educ.* **1995,** *72*, 406–408.
5. Liebl, M. *J. Chem. Educ.* **1988,** *65*, 23–34.
6. Allendoerfer, R. D. *J. Chem. Educ.* **1990,** *67*, 37–39.
7. Douglas, J. E. *J. Chem. Educ.* **1990,** *67*, 42–44.
8. Breneman, G. L. *J. Chem. Educ.* **1988,** *65*, 31–33.
9. Pruett, M.; *Linux J.* **1998,** *58*(Aug), 68–71.
10. Pauling, L.; Wilson, E. B. Jr. *Introduction to Quantum Mechanics with Applications to Chemistry*; Dover: New York, 1935; pp 133–136.
11. Cotton, F. A. *Chemical Applications of Group Theory*, 2nd ed.; Wiley: New York, 1971; pp 368–369.