

0.1 3次元グラフの描き方

ここではパラメータ表示された3次元グラフをgnuplotでどのようにして描くのか?という点を水素原子の波動関数として現れる球面調和関数 $Y_2^0(\theta, \varphi)$ の2乗の描き方を例にご紹介します。

0.2 関数を用意する～球面調和関数～

ここではgnuplotの使い方を説明することを中心としますので、球面調和関数は天下り式に与えてしまいます。描きたい関数を用意する

球面調和関数 $Y_2^0(\theta, \varphi)$ は次の式で定義されます:

$$Y_2^0(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{5}{\pi}} (3 \cos^2 \theta - 1) \quad (1)$$

これは、極座標で表されていますので、次の式が成り立ちます:

$$x = r \sin \theta \cos \varphi, \quad (2)$$

$$y = r \sin \theta \sin \varphi, \quad (3)$$

$$z = r \cos \theta, \quad (4)$$

さて、水素原子の波動関数の場合、実際には3次元空間的に分布した確率密度の濃淡が今回の球面調和関数の2乗になるのですが、これを正確に描くには4次元空間が必要になってしまい、3次元で表せません。そこでよくやる方法なのですが、原点からの距離 r を球面調和関数の2乗とします。2乗する理由は波動関数の絶対値の2乗が確率密度を表す関数になるからです。

まとめると、グラフに描きたい関数は、

$$r = \frac{5}{16\pi} (3 \cos^2 \theta - 1)^2, \quad (5)$$

$$x = r \sin \theta \cos \varphi, \quad (6)$$

$$y = r \sin \theta \sin \varphi, \quad (7)$$

$$z = r \cos \theta, \quad (8)$$

となります。

0.3 パラメータは”u”と”v”

gnuplot で3次元グラフをパラメータ表示で描く場合、そのパラメータは”u”と”v”になります。まずパラメータ表示のグラフを描く場合には、次のコマンドを入力します:

パラメータ表示されたグラフを描くためのコマンド

```
gnuplots>set parametric
```

これを入力して Enter キーを押すと、次のように表示されます:

```
dummy variable is t for curves, u/v for surfaces
```

曲線のときは t, 曲面のときは u と v がパラメータですよって言う意味ですね。そこで、描きたい関数のセット, (5), (6), (7), (8) を u と v で表しておきます。

$\theta \rightarrow u$, $\varphi \rightarrow v$ として考えるとき、gnuplot は定義された関数を覚えることが出来るので次のように入力します:

パラメータ表示された関数を用意するためのコマンド

```
gnuplots>r(u) = 5/(16*pi)*(3*cos(u)**2 - 1)**2
gnuplots>x(u,v) = r(u)*sin(u)*cos(v)
gnuplots>y(u,v) = r(u)*sin(u)*sin(v)
gnuplots>z(u,v) = r(u)*cos(u)
```

上の入力例の中で * は掛け算を表し、通常の数式とはことなり省略は出来ないのに注意が必要です。また **n は n 乗を表し、大抵の数式ソフトでは ^n と入力するのですが、gnuplot ではこの入力方法に対応していませんのでこれも注意が必要です。最後の注意点としては、例えば 1/3 は整数割る整数の商を与える計算という扱いになってしまい $1/3 = 0$ と処理されてしまいます。このため gnuplot では整数同士の割り算は 1.0/3 などのように “.0” をつけて浮動少数点演算させます。ここでこの例はいま極座標を用いていますが、一般に陽に表現されてさえいれば任意のパラメータ表示が使えます。

さて、パラメータ u,v は適切な変域を指定しないと形が変わったり、きれいなグラフが書けなかったりしますので変域を指定しましょう。

パラメータの変域を指定する

```
gnuplots>set urange[0:pi]
gnuplots>set vrange[0:2*pi]
```

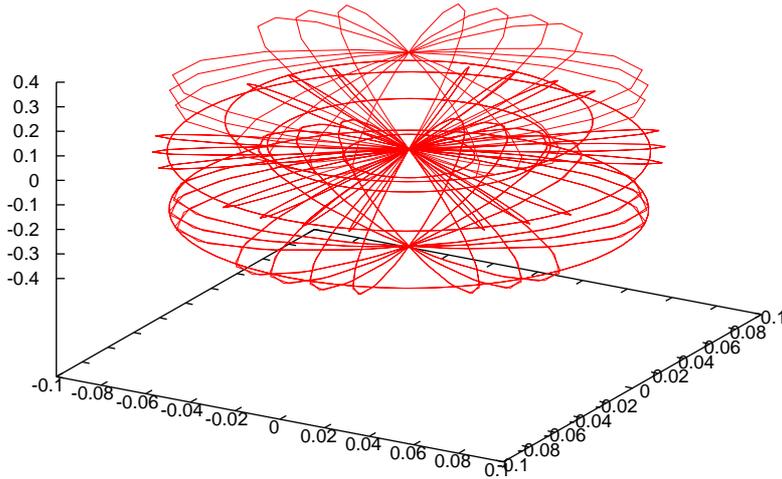
極座標なので上のような変域になります。

さて、ここまでくればあとはプロットの命令をするだけでグラフを描くことが出来ます。splot f(u,v),g(u,v),h(u,v) と入力すれば f(u,v) が変数 x, g(u,v) が変数 y, h(u,v) が変数 z になるようにグラフを描きますので、次のように入力します:

3次元グラフを描くコマンド

```
gnuplots>splot x(u,v),y(u,v),z(u,v)
```

x(u,v),y(u,v),z(u,v) ———



入力して Enter キーを押すと、なにやら上の図のようなグラフらしきものが出てきました。しかしどうも形が変ですね?実は gnuplot は勝手に x, y, z 軸を別々に伸ばしたり縮めたりしてしまう困った特徴があるのです。そこで次のように入力します:

変数の描画領域を指定するコマンド

```
gnuplots>set xrange[-1:1]
gnuplots>set yrange[-1:1]
gnuplots>set zrange[-1:1]
```

これでグラフの描画領域はそれぞれ等しい比率になっていますから、同じグラフを書いても形はいびつにならないはずですが、同じグラフを再描画するコマンドは、

前回の関数を再描画するコマンド

```
gnuplots>replot
```

大体の感じでこのように描画領域をとって見ましたが、今度は形は正しいようですが、肝心のグラフが小さくなってしまいました。もちろん描画領域を小さめに取り直して再描画させてもよいのですが、ここでは別の方法でやってみます。

いま、このグラフが小さくなってしまったのは r の定義式の定数係数部分 $\frac{5}{16\pi}$ が小さすぎるためです、ということはこの係数部分を変数 c にしてやってみて、 c を適当に動かして大きくしてやれば大きいグラフが得られそうです。勿論このやり方では、本来の数値が変わってしまうので厳密な関数の値が欲しい場合には使えません。しかし、ここでは、まあ気楽にグラフの形だけ分かればよいやぐらいの気持ちでこの方法を試してみます。そこで $r(u)$ を再定義します:

$r(u)$ を再定義する

```
gnuplots>r(u) = c*(3*cos(u)**2 - 1)**2
```

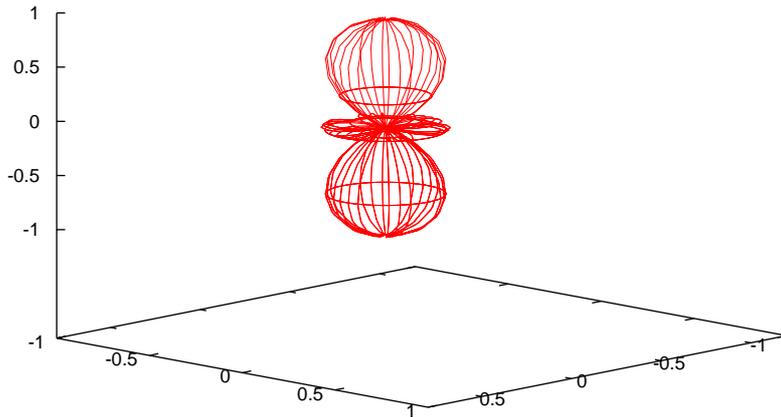
として

c の値を入力する

```
gnuplots>c = 0.2
gnuplots>replot
```

と入力してみると、やや小さく表示されました。そこで今度は $c=0.25$ とするとちょうど良い大きさが得られました。

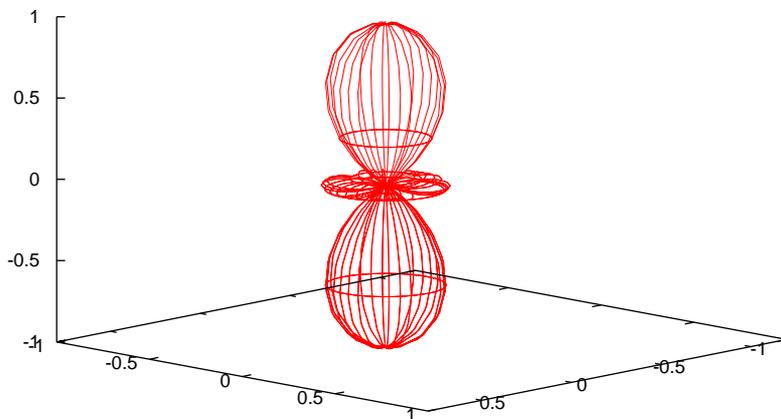
$x(u,v),y(u,v),z(u,v)$ ———



ところでこのグラフの下にスペースが出来てしまっています。この無駄が無ければ、このグラフをより大きく描くことが出来そうです。そこでこの下のスペースをなくするのが次のコマンドです:

```
r(u) を再定義する  
gnuplots>set ticslevel 0
```

$x(u,v),y(u,v),z(u,v)$ ———



実行すると上のグラフのように底面にそろえたグラフが描画されます。

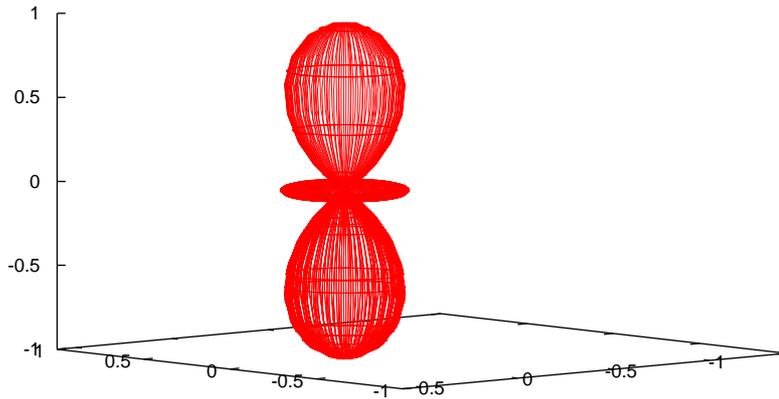
さて、この例の場合充分細かくグラフの線が選ばれているように見えますが、多くの場合グラフの線の間隔が広すぎて、雑なグラフになってしまう場合があります、こんなときに便利なのが次のコマンドです:

グラフの描画の細かさを指定するコマンド

```
gnuplots>set isosample 50,50
```

最初の 50 が z 軸方向の細かさ、次が x 軸と y 軸方向の細かさを表します。

$x(u,v),y(u,v),z(u,v)$ ———

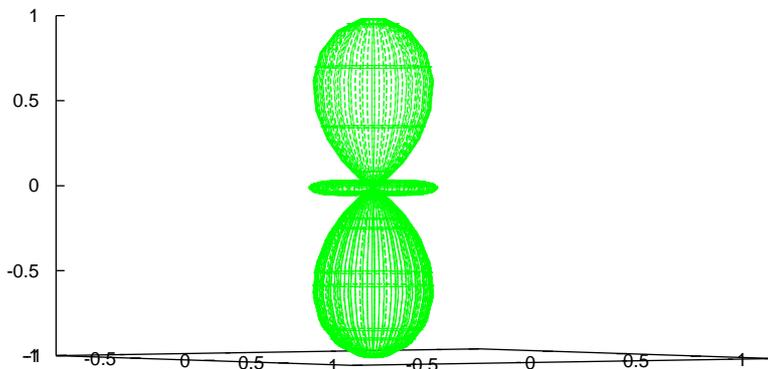


このグラフは赤い線で描かれていますが、これを変える方法があります。プロットするときに次のオプションを付けてください。

グラフの線の色を指定するオプション

```
gnuplots>plot x(u,v),y(u,v),z(u,v) with lines 2
```

$x(u,v),y(u,v),z(u,v)$ - - - - -

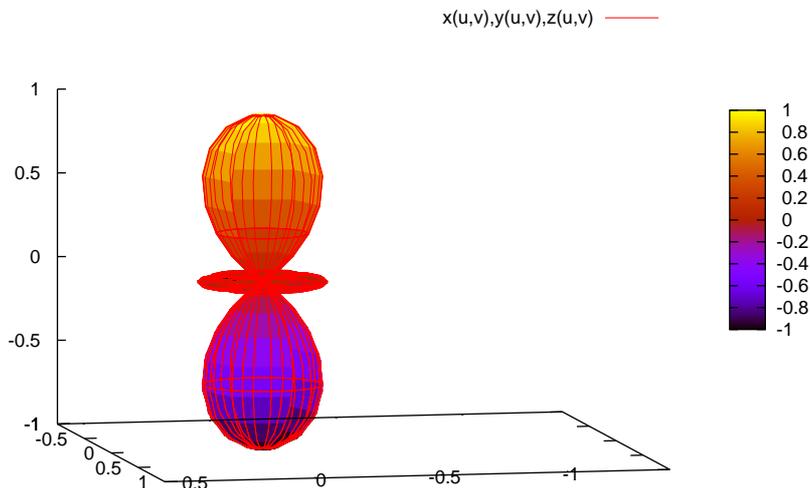


数字を変えると色が変わります。なお、上の図はよく見ると線ではなく”ドット”で描かれていますが、画面に出力する場合にはつながった線が表示されます。eps ファイルに出力する際の仕様のようです。さて、だいぶ3次元グラフらしくなってきましたが、出来れば面のポリゴンに色を塗りたいものです。そんなときには次のコマンドを使います:

グラフの表面の色を塗るコマンド

```
gnuplots>set pm3d
```

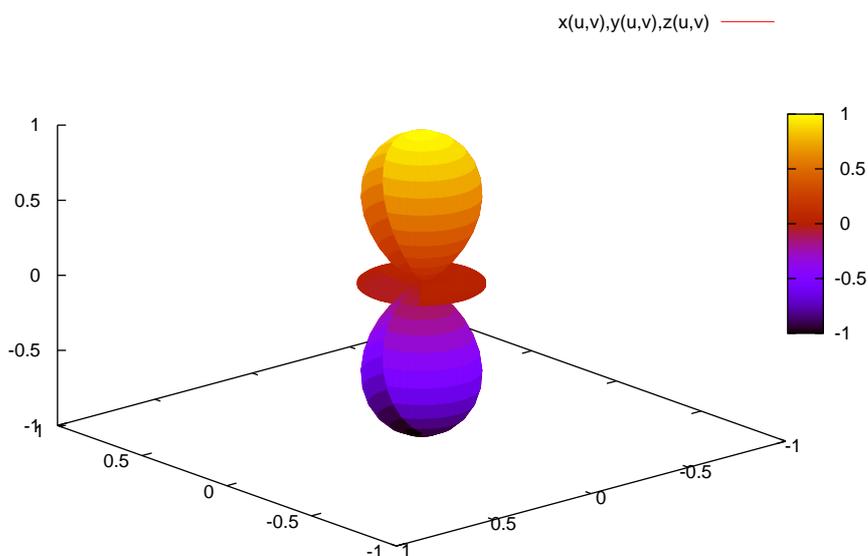
isosample 50,50 では細かすぎるので, isosample 10 にして描画したのが次のグラフです:



さて、実はこのグラフは半透明で描画されており背面が透けてしまっています。これを防止するのが次のコマンドです:

グラフを不透明にするコマンド

```
gnuplots>set hidden3d
```



このコマンドを使用するときは注意が必要です。充分細かいメッシュにしておかないととてつもなく形がいびつになってしまいます。下の例は `isosample = 50` で `hidden3d` にしたものです。

ここまででだいぶきれいなグラフになってきたと思いますが、色に不満がある方もいらっしゃるかもしれません。面の色のグラデーションを変えるには次のコマンドがあります:

グラフの色を変えるコマンド

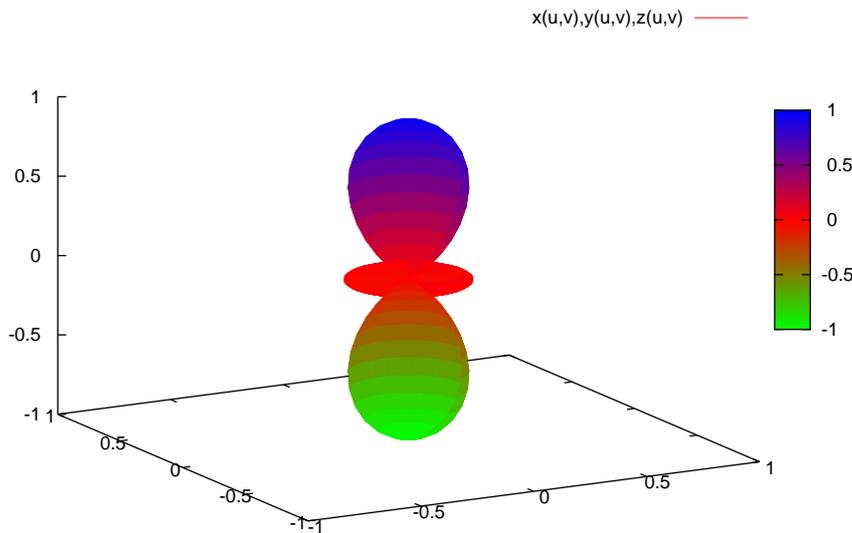
```
gnuplots>set palette defined (-1 "green", 1 "blue")
```

これは z 軸の -1 の地点を green にし 1 の地点を blue になるようにグラデーションさせるコマンドです。このコマンドは始点-終点だけでなく、次のように始点-中間-終点のように指定することも出来ます:

グラフの色を変える指定

```
gnuplots>set palette defined (-1 "#00FF00", 0 "#FF0000", 1 "#00FF00")
```

この指定をすると次のようなグラフになります:



このように色の指定はカラーネームだけでなく RGB それぞれ 256 ビットの指定が出来ます。実は色の指定は別の方法もあって、

グラフの色を変える指定その 2

```
gnuplots>set palette rgbformulae 22, 13, -31
```

なども使えます。しかしこの方法について筆者はあまり分かっていませのでコマンドの紹介だけにとどめます。

0.4 軸やラベルの消去について

これは筆者の個人的な好みの問題ですが、グラフ以外に余計な目盛りやカラーバーなどをなくしたいと思うことがしばしばありましたので、それについて書きます。まず、面に色を塗ったときに現れるカラーバーは、

カラーバーを非表示にするコマンド

```
gnuplots>unset colorbox
```

でなくすことが出来ます。また、関数名と線の色 of 凡例の表示は、

関数名と線の色 of 凡例を非表示にするコマンド

```
gnuplots>unset key
```

でなくすことが出来ます。数値の表示は、

x 軸の数値を非表示にする設定

```
gnuplots>set format x ""
```

で非表示に出来ます。なお""の部分に入れた文字やフォーマットオプションが各数値の表示部分に表示されます。また x 軸 y 軸 z 軸の刻み表示は

x 軸の刻みを非表示にするコマンド

```
gnuplots>unset xtics
```

などで行えます。このようにしても、最終的に枠が残ってしまうのですが、これも

枠を非表示にするコマンド

```
gnuplots>unset border
```

によってなくすことが出来ます。

タイトルをなくすときには単に、

タイトルを非表示にするコマンド

```
gnuplots>set title
```

と打ちますが、逆に表示するときには、

タイトルを表示するコマンド

```
gnuplots>set title "test"
```

とします。残念ながら日本語のタイトル表示には対応していないようです。

0.5 グラフのファイル出力と画面出力

グラフをカラーの eps ファイルに出力するのは意外と簡単です。次の3行を入力してください:

関数名と線の色を非表示にするコマンド

```
gnuplots>set term postscript eps enhanced color
gnuplots>set output "test.eps"
gnuplots>replot
```

最初の行はグラフの出力先をカラー eps ファイルにする指定です。次の行はそのためのファイル名を指定する行です。そして必ず忘れてはいけないのが、そのグラフを再度プロットすることです。出力先がファイルなのでこのコマンドを打ち忘れると、開けないサイズが0の eps ファイルだけが出来てしまって、困ってしまいます。

あえてモノクロにしたいときには、

関数名と線の色を非表示にするコマンド

```
gnuplots>set palette gray
gnuplots>set term postscript eps enhanced monochrome
gnuplots>set output "test.eps"
gnuplots>replot
```

などと入力すると白黒の表示になります。ポイントはカラーパレットもモノクロにしておくことです。



hidden3dOFF



hidden3dON

再度グラフをモニタに出力したい場合は、

グラフをモニタに出力するコマンド

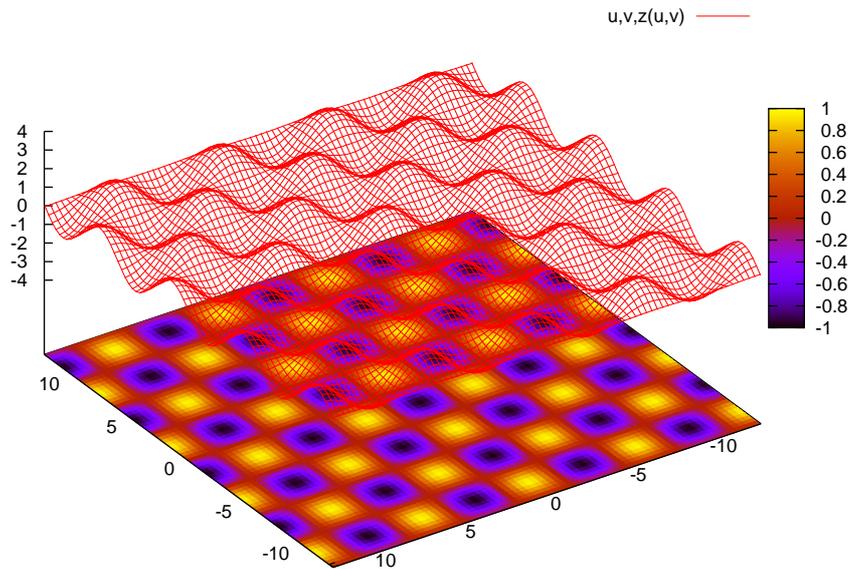
```
gnuplots>set term windows
gnuplots>replot
```

と入力します。

0.6 その他のコマンド

グラフの底面に等高線を表示するオプション

```
gnuplots>set pm3d at b
```



グラフを非表示にするオプション

```
gnuplots>set pm3d at b  
gnuplots>unset surface
```

この設定にすると底面の等高線だけが表示されます。等高線だけ表示させるには次の方法もあります:

等高線のみ表示するオプション

```
gnuplots>set pm3d map
```

それぞれの表示の違いに注意してください。

