

# CONTENTS

---

<b>サンプルの Experiment – sphericalHarmonics Demo .....</b>	<b>2</b>
クイックノート .....	2
手順.....	2
プロセスの内容.....	6
magsqr 関数のヘルプ.....	7
sphericalHarmonics 関数のヘルプ .....	8

# サンプルの Experiment – sphericalHarmonics Demo

## クイックノート

メニュー **File** → **Example Experiments** → **Visualization** → **sphericalHarmonicsDemo**

この Experiment には、パラメトリックサーフェスの例が含まれています。

サーフェスデータウェーブは、メインプロシージャウィンドウの関数 calcParametric() によって生成されます。この関数は、Spherical Harmonics Panel のコントロールの変更に応じて呼び出されます。

ここでは、デモの Experience と同じものを新しい Experiment から作成する方法を説明します。

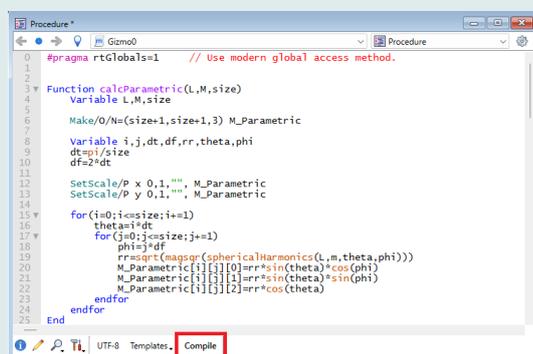
説明を短くするために、デモからデータをコピーするため、デモと新規の2つのインスタンスを起動しておきます。

## 手順

新しい Experiment を作成したところからの手順で確認します。

**1. デモの Example で、メニュー Windows → Procedure Windows → Procedure Window を選択し、プロシージャ全体をコピーし、新しい Experiment の Procedure Window にペーストします。**

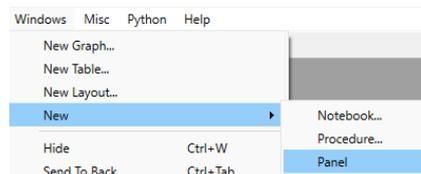
ウィンドウ下部の **Compile** ボタンをクリックします。



```
1 #pragma rtGlobals=1 // Use modern global access method.
2
3 Function calcParametric(L,M,size)
4 Variable L,M,size
5
6 Make/O/N=(size+1,size+1,3) M_Parametric
7
8 Variable i,j,dt,df,rr,theta,phi
9 dt=pi/size
10 df=2*dt
11
12 SetScale/P x 0.1,"", M_Parametric
13 SetScale/P y 0.1,"", M_Parametric
14
15 for (i=0;i<=size;i+=1)
16   theta=i*dt
17   phi=j*df
18   for (j=0;j<=size;j+=1)
19     rrsqrt(magsqr(sphericalHarmonics(L,m,theta,phi)))
20     M_Parametric[i][j][0]=rr*cos(theta)*cos(phi)
21     M_Parametric[i][j][1]=rr*cos(theta)*sin(phi)
22     M_Parametric[i][j][2]=rr*cos(theta)
23   endfor
24 endfor
25 End
```

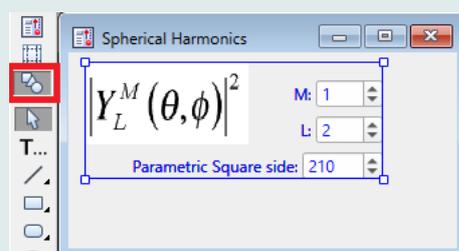
**2. 新しい Experiment で、メニュー Windows → New → Panel を選択します。**

パネルの作成ウィンドウが開きます。



**3. デモの Experiment の Spherical Harmonic ウィンドウを編集モードにし、Ctrl+A キーで全ての要素を選択します。**

Ctrl+C キーでコピーし、新しい Experiment のパネルにペーストします。

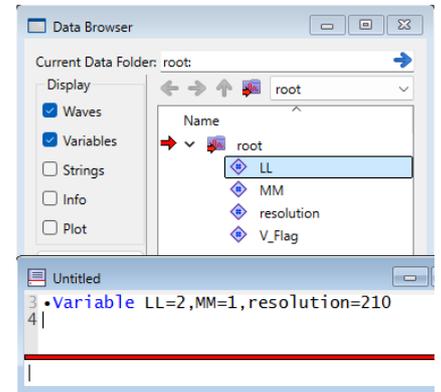


4. 次に、パネルの数値をプロシージャに渡すための変数を作ります。

ここではデフォルト値も設定します。

コマンドウィンドウで次を実行します。

```
Variable LL=2,MM=1,resolution=210
```

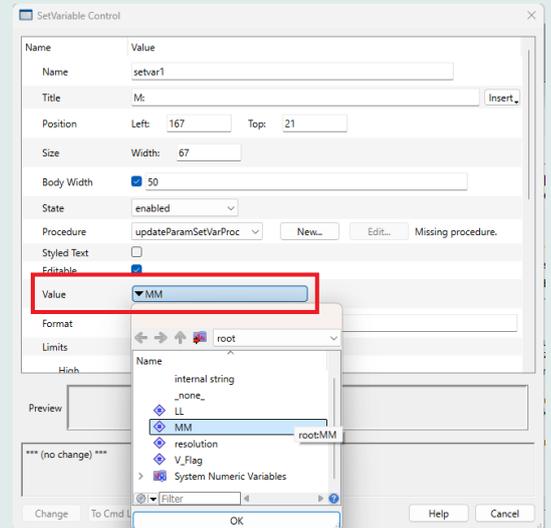


5. パネルの要素を変数と関連付けます。

パネルの「M:」の要素をダブルクリックします。

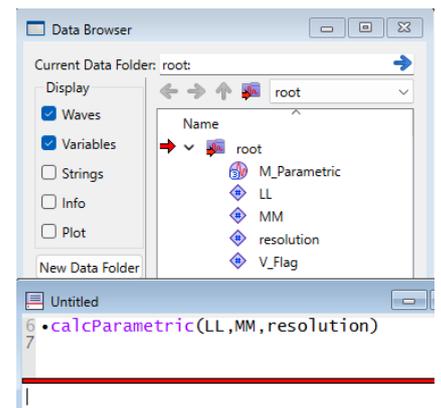
SetVariable Control ダイアログの Value に「MM」を設定します。

同様に、「L:」には「LL」、「Parametric Square side:」には「resolution」を設定します。



6. コマンドウィンドウで次を入力して初期状態のウェーブ (M\_Parametric) を作ります。

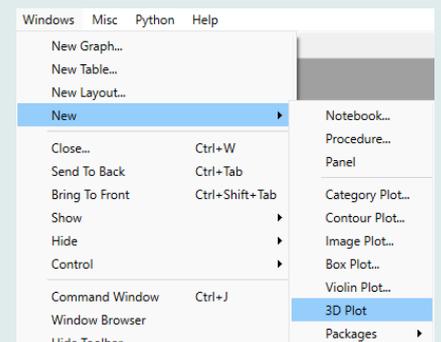
```
calcParametric(LL,MM,resolution)
```



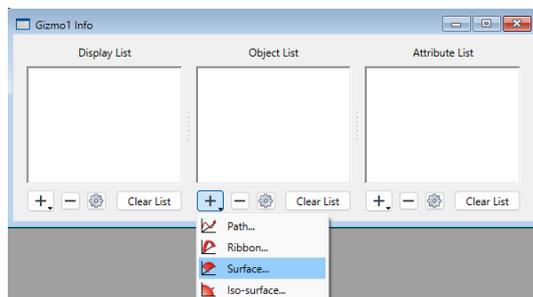
7. ここからは、Gizmo を使って、このウェーブの 3D グラフィックスを作成します。

メニュー **Windows** → **New** → **3D Plot** を選択します。

Gizmo の表示ウィンドウと、情報パネルが開きます。

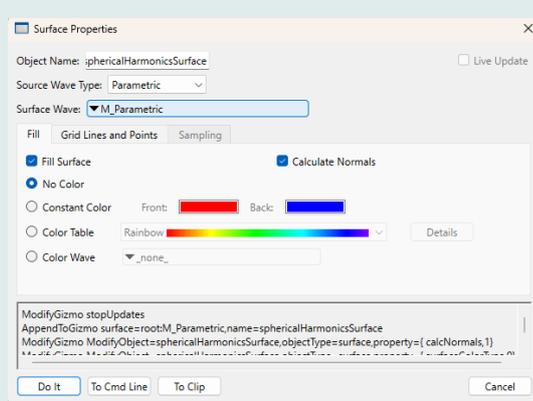


**8. Gizmo Info パネルの Object List の「+」をクリックして、Surface を選択します。**



**9. Surface Properties ダイアログで、次のように設定します。**

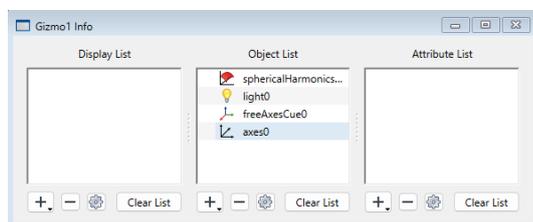
Object Name: sphericalHarmonicsSurface  
 Source Wave Type: Parametric  
 Surface Wave: M\_Parametric  
 Fill Surface オン  
 Calculate Normals オン  
 No Color ラジオボタンを選択



**設定したら、Do It をクリックします。**

**10. Gizmo Info パネルの Object List の「+」をクリックして、light (光源)、freeAxesCue (自由軸)、axes (軸) をリストに追加します。**

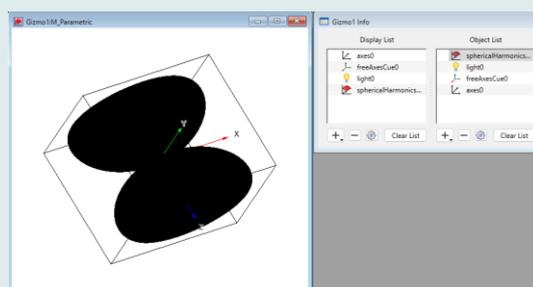
それぞれ設定画面が表示されますが、とりあえずデフォルトのまま追加します。



**11. Gizmo Info パネルの Object List の4つの項目を、Display List にドラッグします。**

上から、axes、freeAxesCue、light、sphericalHarmonicsSurface の順になるようにします。

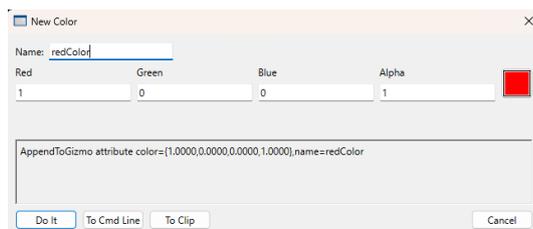
表示は上から順に処理されていくため、順番は重要です。



**12. Gizmo Info パネルの Attribute List の「+」をクリックして、Color を選択します。**

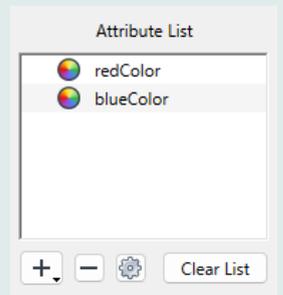
**New Color ダイアログで、次のように設定します。**

Name: redColor  
 Red: 1  
 Green: 0  
 Blue: 0  
 Alpha: 1



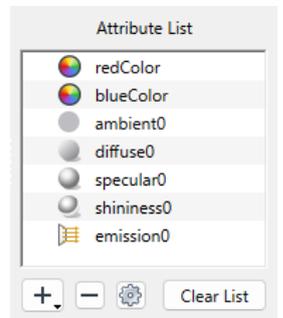
**13. 同じようにもう一つ Color を追加して、次のように設定します。**

Name: blueColor  
 Red: 0  
 Green: 0  
 Blue: 1  
 Alpha: 1



**14. 他の Attribute を追加します。**

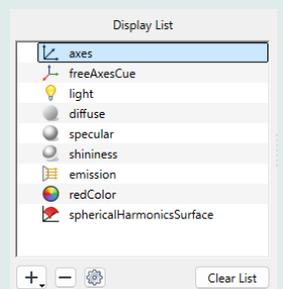
ambient GL\_FRONT\_AND\_BACK、色は黒  
 diffuse GL\_FRONT\_AND\_BACK、色は赤  
 specular GL\_FRONT\_AND\_BACK、色は黄  
 shininess Front Shininess=25, Back Shininess=25  
 emission GL\_FRONT\_AND\_BACK、色は黒



**15. Attribute List の項目を、Display List にドラッグします。**

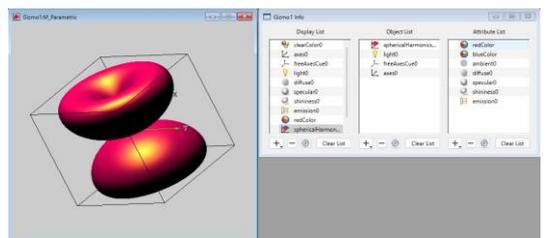
上から、axes、freeAxesCue、light、diffuse、specular、shininess、emission、redColor、sphericalHarmonicsSurface の順になるようにします。

blueColor は使いませんが、オブジェクトを青にしたい場合には、これを使います。

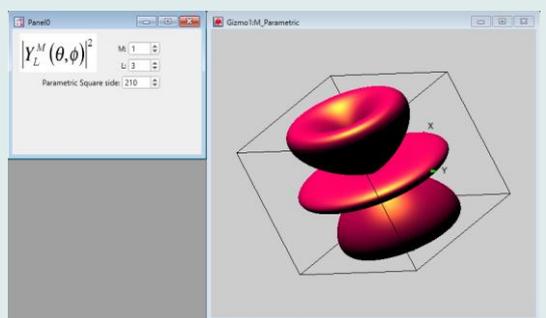


**16. Display List の下の「+」をクリックし、clearColor を選択して、リストの一番上に追加します。**

最終的に、右のような表示が得られます。



**17. 作成したパネルを操作モードにして、変数の数字を変えてみると、パネルの設定が 3D 表示に反映されることがわかります。**



## プロシージャの内容

// ウェーブ (M\_Parametric) を生成するための計算を行います。

```
Function calcParametric(L,M,size)
  Variable L,M,size

  Make/O/N=(size+1,size+1,3) M_Parametric

  Variable i,j,dt,df,rr,theta,phi
  dt=pi/size
  df=2*dt

  SetScale/P x 0,1,"", M_Parametric
  SetScale/P y 0,1,"", M_Parametric

  for(i=0;i<=size;i+=1)
    theta=i*dt
    for(j=0;j<=size;j+=1)
      phi=j*df
      rr=sqrt(magsqr(sphericalHarmonics(L,m,theta,phi)))
      M_Parametric[i][j][0]=rr*sin(theta)*cos(phi)
      M_Parametric[i][j][1]=rr*sin(theta)*sin(phi)
      M_Parametric[i][j][2]=rr*cos(theta)
    endfor
  endfor
End

Function updateParamSetVarProc(ctrlName,varNum,varStr,varName) : SetVariableControl
  String ctrlName
  Variable varNum
  String varStr
  String varName

  NVAR LL,MM,resolution

  if(abs(MM)>LL)
    if(MM<0)
      MM=-LL
    else
      MM=LL
    endif
  endif

  // Gizmo がウェーブを calcParametric() で更新しないようにするには、次が必要です。
  ModifyGizmo stopUpdates
  calcParametric(LL,MM,resolution)
  ModifyGizmo resumeUpdates
End
```

// この関数は、デモの手順の説明では使っていません。

// コマンドウィンドウで

// Gizmo0()

// を実行すると、軸、光源、サーフェスの設定をして、表示までを一括して行えます。

```

Window Gizmo0() : GizmoPlot
    PauseUpdate; Silent 1          // Gizmo 6 のウィンドウを構築します。

    // Gizmo XOP がない場合は、何もしません。
    if(exists("NewGizmo")!=4)
        DoAlert 0, "Gizmo XOP must be installed"
        return
    endif

    // Gizmo の要素を設定し、3D 表示までを行います。
    NewGizmo/N=Gizmo0/T="Gizmo0" /W=(14,44,687,614)
    ModifyGizmo startRecMacro
    AppendToGizmo Surface=root:M_Parametric,name=surface0
    ModifyGizmo ModifyObject=surface0 property={ srcMode,4}
    ModifyGizmo ModifyObject=surface0 property={ surfaceCTab,YellowHot256}
    ModifyGizmo ModifyObject=surface0 property={ SurfaceCTABScaling,16}
    ModifyGizmo modifyObject=surface0 property={calcNormals,1}
    AppendToGizmo light=Directional,name=light0
    ModifyGizmo light=light0 property={ position,-
0.832778,0.305999,0.461353,0.000000}
    ModifyGizmo light=light0 property={ direction,-0.832778,0.305999,0.461353}
    ModifyGizmo light=light0 property={ ambient,0.533333,0.533333,0.533333,1.000000}
    ModifyGizmo light=light0
property={ specular,1.000000,1.000000,1.000000,1.000000}
    AppendToGizmo freeAxesCue={0,0,0,1.3},name=freeAxesCue0
    AppendToGizmo Axes=boxAxes,name=axes0
    ModifyGizmo ModifyObject=axes0,property={-1,axisScalingMode,1}
    ModifyGizmo ModifyObject=axes0,property={-1,axisColor,0,0,0,1}
    ModifyGizmo setDisplayList=0, object=light0
    ModifyGizmo setDisplayList=1, opName=pushAttribute0, operation=pushAttribute,
data=1
    ModifyGizmo setDisplayList=2, object=surface0
    ModifyGizmo setDisplayList=3, opName=popAttribute0, operation=popAttribute
    ModifyGizmo setDisplayList=4, object=freeAxesCue0
    ModifyGizmo setDisplayList=5, object=axes0
    ModifyGizmo SETQUATERNION={0.046616,-0.515933,-0.845555,0.129079}
    ModifyGizmo autoscaling=1
    ModifyGizmo currentGroupObject=""
    ModifyGizmo compile

    ModifyGizmo showInfo
    ModifyGizmo infoWindow={693,217,1121,500}
    ModifyGizmo bringToFront
    ModifyGizmo endRecMacro
End

```

## magsqr 関数のヘルプ

### magsqr (z)

magsqr 関数は、複素数  $z$  の実部と虚部の二乗和、つまり大きさを二乗したものを返します。

## 例

waveCmplx は複素数、waveReal は実数と仮定します。

```
waveReal= sqrt(magsqr(waveCmplx))
```

は、waveReal の各ポイントを waveCmplx の複素ポイントの大きさに設定します。

waveCmplx のポイント数と waveReal のポイント数が異なる場合、予期せぬ結果が生じる可能性があります。

## sphericalHarmonics 関数のヘルプ

**sphericalHarmonics** ( $L, M, \theta, \varphi$ )

sphericalHarmonics 関数は、複素数型の球面調和関数を返します。

$$Y_L^M(\theta, \varphi) = (-1)^M \sqrt{\frac{2L+1}{4\pi} \frac{(L-M)!}{(L+M)!}} P_L^M(\cos(\theta)) e^{iM\varphi}$$

ここで、 $P_L^M(\cos(\theta))$  は関連するルジャンドル (Legendre) 関数です。

### 参照

Arfken, G., Mathematical Methods for Physicists, Academic Press, New York, 1985.