

CONTENTS

ビジュアルヘルプ - 3D グラフィックス (1)	2
3D グラフィックス	2
1D ウェーブ	2
2D ウェーブ	3
トリプレットウェーブ	3
3D ウェーブ	4
Gizmo の概要	4
システム要件	4
ハードウェア互換性	5
Gizmo ガイドツアー	5
Gizmo 3D 散布図ツアー	5
Gizmo サーフェスプロットツアー	11
Gizmo 3D 散布図&カーブフィッティングサーフェスツアー	17
Voronoi 補間を使った Gizmo サーフェスツアー	19
Gizmo のウィンドウ	22
Gizmo 表示ウィンドウ	22
Gizmo 表示ウィンドウのツールパレット	23
Gizmo 表示ウィンドウのコンテキストメニュー	24
Gizmo 情報ウィンドウ	26
Gizmo Object List	27
Gizmo Display List	27
Gizmo Display List 内のアイテムの順序	27
Gizmo Attribute List	28
Gizmo オブジェクト	28
Gizmo の属性	29
内部属性	30
グローバル属性	30
埋め込み属性	31

ビジュアルヘルプ – 3D グラフィックス (1)

3D グラフィックス

Igor では、さまざまな 3D グラフィックスを作成できます。

- Surface Plots
- 3D Scatter Plots
- 3D Bar Plots
- Path Plots
- Ribbon Plots
- Isosurface Plots
- Voxelgram Plots

画像プロット、コンタープロット、ウォーターフォールプロットは 2D グラフィックスとみなされ、ヘルプやマニュアルの他のセクションで説明されています。

Igor の 3D グラフィックスツールは「Gizmo (ギズモ)」と呼ばれています。

Gizmo で作成する 3D グラフィックスのほとんどは、ウェーブに保存されたデータに基づいています。

どのようなタイプの 3D グラフィックスにどのようなタイプのウェーブデータが必要なのかを理解することが重要です。

以降のセクションで説明していきます。

1D ウェーブ

1D ウェーブは 3D プロットに使うことができません。

X、Y、Z 座標を表す 3つの 1D ウェーブを 3D プロットとして表示したい場合は、それらをトリプレットウェーブに変換する必要があります。

例えば、次のようになります (詳細は後で説明します)。

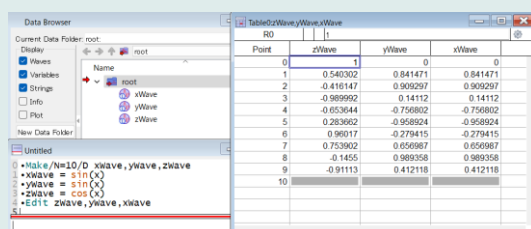
1. 新しいエクスペリメントを作成して、コマンドウィンドウで次を実行して、3つの 1D ウェーブを作成します。

```
Make/N=10/D xWave,yWave,zWave
```

```
xWave = sin(x)
```

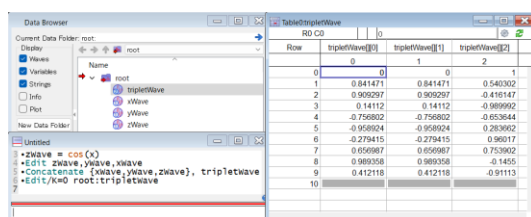
```
yWave = sin(x)
```

```
zWave = cos(x)
```

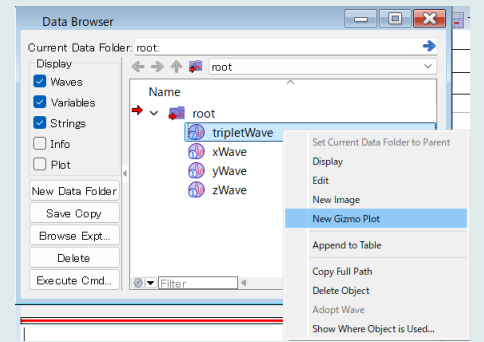


2. 次のコマンドを実行して、トリプレットウェーブに変換します。

```
Concatenate {xWave,yWave,zWave}, tripletWave
```

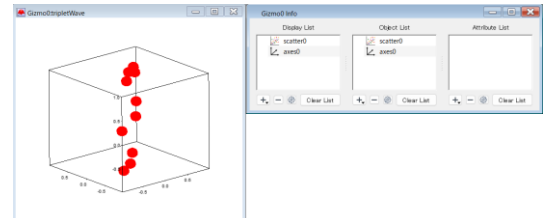


3. Data Browser で、作成したトリプレットウェーブを右クリックして、New Gizmo Plot を選択します。



4. 右のような 3D プロットが作成されます。

マウスでプロットを回転させてみてください。



3つの 1D ウェーブをトリプレットウェーブに変換することは、データが長方形のグリッドでサンプリングされていない場合に適しています。

データが長方形のグリッドでサンプリングされていることがわかっている場合は、Z データを含むウェーブを Redimension コマンドで 2D ウェーブを使ってサーフェスをプロットします。

この変換は、例えば、次のコマンドを使って実行できます。

```
Duplicate/O zWave, zMatrixWave  
Redimension/N=(numRows,numColumns) zMatrixWave
```

2D ウェーブ

2D ウェーブは、「Z 値の行列」とも呼ばれ、M 列 N 行のウェーブで、各要素がスカラー値 Z を表します。ウェーブのスケールリング（ヘルプ Waveform Model of Data を参照）を適用して、各行に X 値を、各列に Y 値を関連付けることができます。

2D ウェーブは、サーフェスプロットおよび 3D 棒グラフで 3D グラフィックスとして表示することができます。（2D ウェーブは、画像プロット、コンタープロット、ウォーターフォールプロットの 2D グラフィックスとしても表示できます）

トリプレットウェーブ

トリプレットウェーブは、各行に XYZ トリプレットを含む 3 列 M 行のデータです。X 値は最初の列に、Y 値は 2 番目の列に、Z 値は 3 番目の列に表示されます。トリプレットウェーブとは、X、Y、Z の座標を含む 2D ウェーブと解釈されます。

トリプレットウェーブは、3D 散布図、3D サーフェスプロット、3D パスプロット、3D リボンプロットで 3D グラフィックスとして表示することができます。

サーフェスプロットでは、トリプレットウェーブは表面上の三角形を定義します。

（トリプレットウェーブは、コンタープロットで 2D グラフィックスとして表示することもできます）

3D ウェーブ

3D ウェーブは、「ボリウム」と呼ばれることもあり、M 列× N 行× L 層のウェーブで、各要素がスカラー値 Z を表します。

3D グラフィックスで使うには、3D ウェーブには各次元で少なくとも 2 つの要素を含める必要があります。

3D ウェーブは、サーフェスプロット、等値面プロット、ボクセルグラムプロットで 3D グラフィックスとして表示できます。

(3D ウェーブは、3D ウェーブの 1 つのレイヤーが画像として表示される画像プロットで、2D グラフィックスとして表示することもできます)

Gizmo の概要

Igor の 3D プロットツールは「Gizmo (ギズモ)」と呼ばれています。

Gizmo は、3D グラフィックスの業界標準システムである OpenGL をベースとしています。

Gizmo は、Igor のデータとコマンドを OpenGL のデータと命令に変換し、その結果を「Gizmo ウィンドウ」と呼ばれる Igor のウィンドウに表示します。

Gizmo ウィンドウを作成するには、メニュー Windows→New→3D Plot を選択し、Gizmo メニューを使ってグラフィックオブジェクトを追加します。

OpenGL の知識がなくてもこれを行うことができます。

このレベルでは、サーフェスプロット、散布図、パスプロット、ボクセルグラム、等値面プロット、3D 棒グラフ、3D 円グラフを作成できます。

このようなオブジェクトは、次に説明する描画オブジェクトと区別するために、「ウェーブベースのオブジェクト」または「データオブジェクト」と呼びます。

基本的なプロットを作成した後、Gizmo の情報ウィンドウを使ってさまざまなプロパティを変更することができます。

上級ユーザーは、直線、三角形、四角形、立方体、球体、円柱、円盤、四面体、円錐などの 3D プリミティブのセットを使ってグラフィックスを構築することもできます。

このようなオブジェクトは、ウェーブベースのオブジェクトと区別するために、「ドローイングオブジェクト」と呼びます。

適切なスケーリングを適用すれば、描画オブジェクトとウェーブベースのオブジェクトを同じ Gizmo ウィンドウに組み合わせることができます。

Gizmo は、照明効果、光沢、テクスチャなどの高度な OpenGL 機能をサポートしています。

高度な 3D グラフィックスを作成したい上級ユーザーは、3D グラフィックス全般、特に OpenGL を知っていることが役立つでしょう。

システム要件

Gizmo の動作の多くは、使用しているコンピューターのグラフィックハードウェアとグラフィックドライバーソフトウェアに依存しています。

Gizmo を実行するには、VRAM 512MB 以上のグラフィックカードを搭載したハードウェアを推奨します。

Gizmo は、オンボードグラフィックスと共有メモリを搭載したコンピューターでも動作しますが、パフォーマンスが低下し、グラフィックスをエクスポートするときにエラーが発生する場合があります。

ハードウェア互換性

Gizmo グラフィックスは、実行中の OpenGL のバージョンに影響を受ける場合があります。

これは、グラフィックハードウェア、グラフィックドライバのバージョン、グラフィックアクセラレーションの設定によって異なります。

Gizmo ガイドツアー

以下のセクションのチュートリアルでは、Gizmo の基本的な機能、基本的な Gizmo プロットの作成方法、特定の種類のプロットのデータの形式について理解を深めることができます。

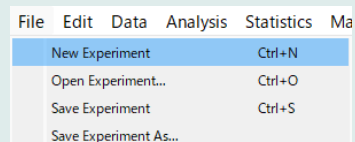
ツアーのさまざまなポイントで、Igor のコマンドの実行が指示されます。

ヘルプファイルでコマンドを選択し、Ctrl-Enter キーを押すことでそのコマンドを実行することができます。

Gizmo 3D 散布図ツアー

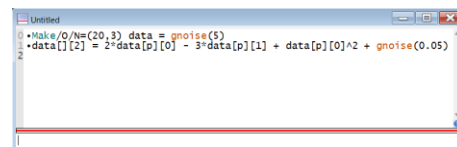
このツアーでは、XYZ データを含むトリプレットウェーブを作成し、それを 3D 散布図としてプロットします。

1. メニュー File→New Experiment を選択して、新しいエクスペリメントを作成します。



2. 次のコマンドを実行して、XYZ 散布データを持つトリプレットウェーブを作成します。

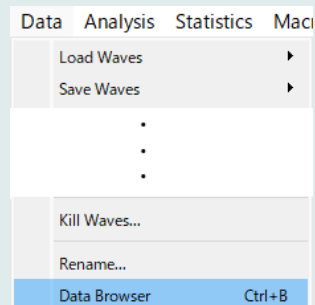
```
Make/O/N=(20,3) data = gnoise(5)
data[][2] = 2*data[p][0] - 3*data[p][1] + data[p][0]^2
+ gnoise(0.05)
```



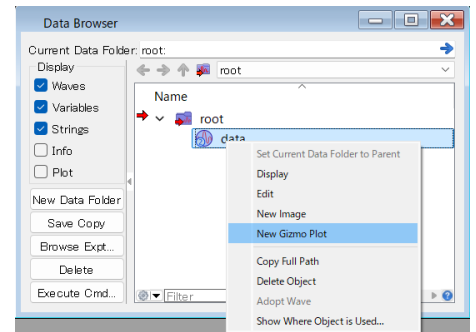
この散布データは、X と Y の多項式関数にほぼ等しい Z 値を持つ XY 平面内のランダムな位置を示しています。

3. このデータを Gizmo に表示するもっとも簡単な方法は、Data Browser でこのデータを右クリックして New Gizmo Plot を選択することです。

メニュー Data→Data Browser を選択します。
(既に開いているかもしれません)



4. data ウェーブのアイコンを右クリックして、New Gizmo Plot を選択します。

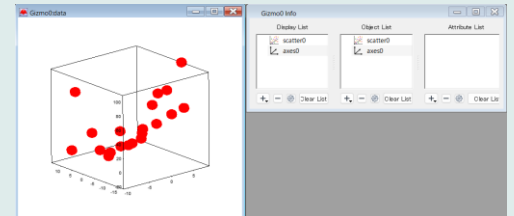


5. Gizmo0 という名前の新しいウィンドウで、data ウェーブから Gizmo 3D 散布図が作成されました。

また、「Gizmo0 Info」というタイトルで Gizmo 情報ウィンドウが作成されました。

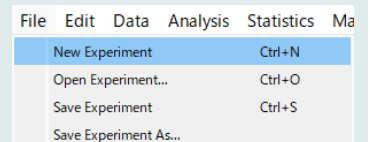
(Gizmo0 Info ウィンドウが表示されていない場合は、メニュー Gizmo→Show Info を選択してください)

Gizmo 散布図は、Gizmo0 ウィンドウ内のプロットをドラッグ、回転したり、矢印キーを使うことでも回転させることができます。自由に操作してみてください。



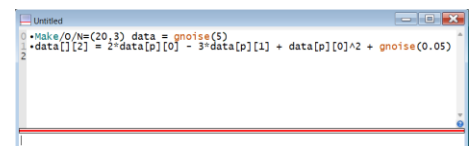
ここまではあまりにも簡単すぎて、参考にならないかもしれません。そこで、Data Browser のショートカットを使わずにやり直してみます。

1. File→New Experiment を選択して、新しいエクスペリメントを作成します。



2. 次のコマンドを実行して、XYZ 散布データを持つトリプレットウェーブを作成します。

```
Make/O/N=(20,3) data = gnoise(5)
data[][2] = 2*data[p][0] - 3*data[p][1] + data[p][0]^2
+ gnoise(0.05)
```



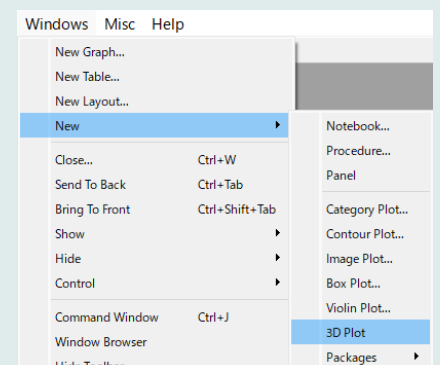
3. メニュー Windows→New→3D Plot を選択します。

コマンドウィンドウの履歴エリアに次のコマンドがあることに注目してください。

NewGizmo

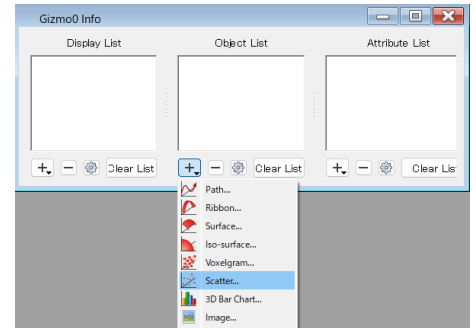
Igor は、空の Gizmo0 ウィンドウと Gizmo0 Info ウィンドウも作成しました。

(Gizmo0 Info ウィンドウが表示されていない場合は、メニュー Gizmo→Show Info を選択してください)



4. Gizmo0 Info ウィンドウの Object List (中央) の下にある「+」アイコンをクリックし、Scatter を選択します。

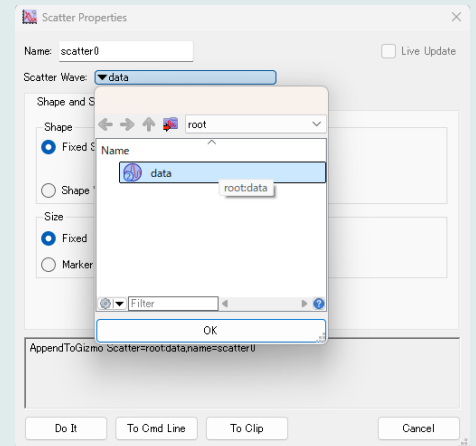
Scatter Properties ダイアログが開きます。



5. Scatter Wave メニューから data を選択します。

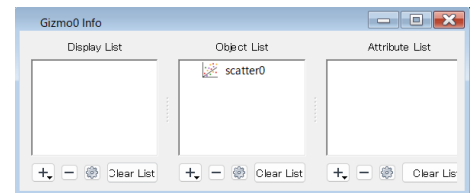
このメニューでは、トリプレットウェーブのみが表示されます。3D 散布図を作成するには、トリプレットウェーブが必要です。

他にもいくつかのオプションがありますが、今回はデフォルトのままにしておきます。



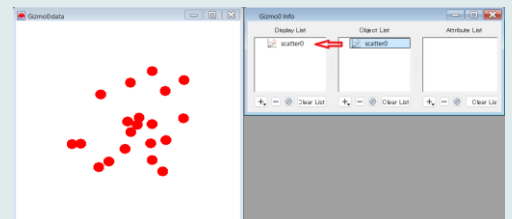
6. Do It をクリックします。

Igor は scatter0 という名前の 3D 散布オブジェクトを作成し、それを情報ウィンドウの Object List に追加しました。まだ Display List には追加していないため、Gizmo0 ウィンドウには表示されていません。



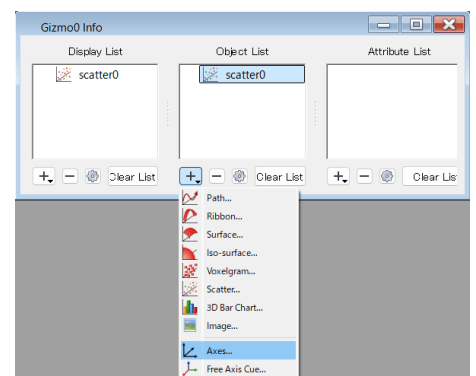
7. scatter0 オブジェクトを Object List から Display List にドラッグします。

XYZ データを表す球体が Gizmo0 ウィンドウに表示されます。プロットは完全ではありませんが、Gizmo0 ウィンドウをクリックしてドラッグすると、表示を回転させることができます。



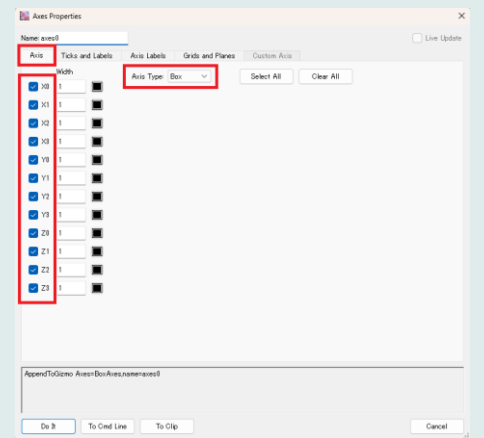
8. Gizmo0 Info ウィンドウで、Object List の下にある「+」アイコンをクリックし Axes を選択します。

Axes Properties ダイアログが表示されます。



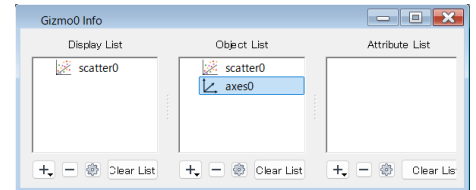
9. 選択されていない場合は、Axis タブを選択します。

Axis Type ポップアップメニューは Box に設定し、すべての軸のチェックボックス (X0、X1、…Z2、Z3) にチェックを入れます。



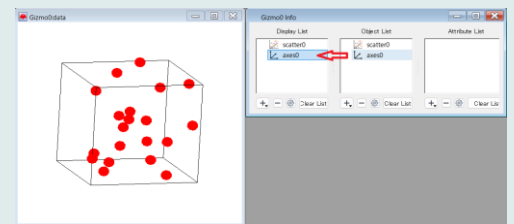
10. Do It をクリックします。

Igor は、axes0 という名前の axes オブジェクトを作成し、それを情報ウィンドウの Object List に追加しました。まだ Display List には追加していないため、Gizmo0 ウィンドウには表示されていません。



11. axes0 オブジェクトを Object List から Display List にドラッグします。

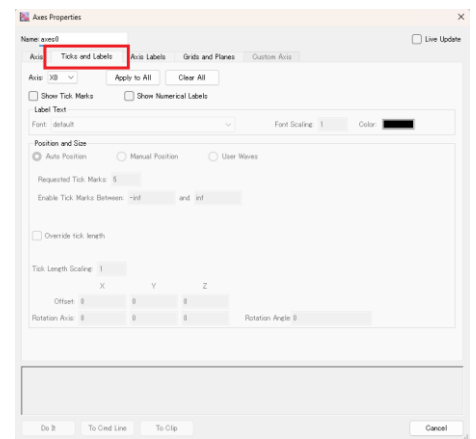
散布した球体の周りに箱型の軸が追加されます。



12. Object List または Display List のどちらかで、axes0 オブジェクトをダブルクリックします。

Axes Properties ダイアログが再び開きます。

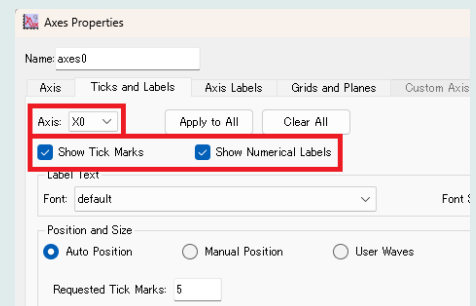
Ticks and Labels タブをクリックします。



13. Axis ポップアップメニューから X0 を選択し、Show Tick Marks と Show Numerical Labels のチェックボックスをオンにします。

Axis ポップアップメニューから Y0 を選択し、Show Tick Marks と Show Numerical Labels のチェックボックスをオンにします。

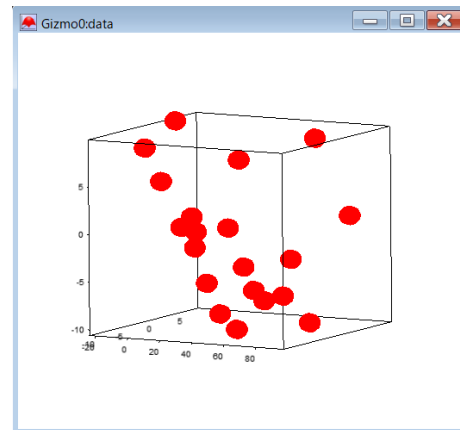
Axis ポップアップメニューから Z0 を選択し、Show Tick Marks と Show Numerical Labels のチェックボックスをオンにします。



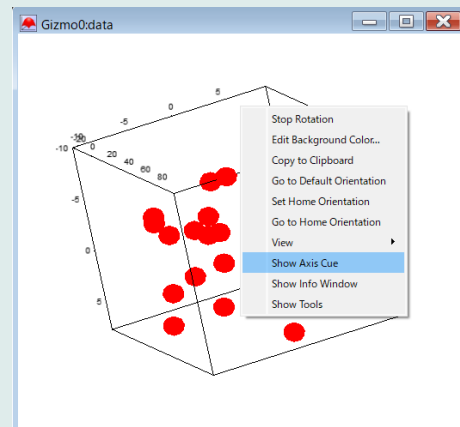
14. Do It をクリックします。

X0、Y0、Z0 軸に目盛ラベルが付けられました。

しかし、どの軸がどれなのでしょう？

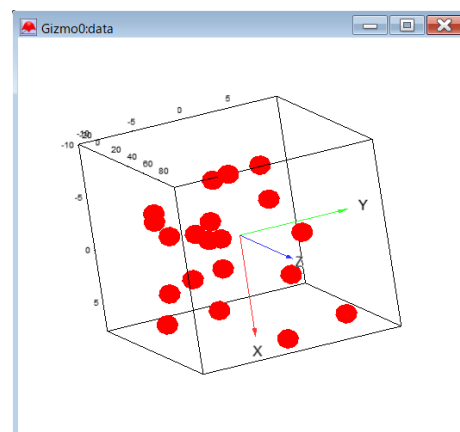


15. Gizmo0 ウィンドウのどこかを右クリックし、Show Axis Cue を選択します。

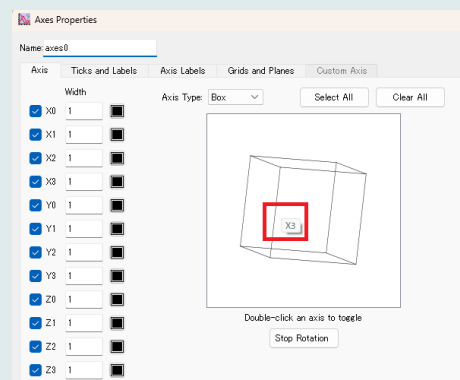


16. どの次元がどれであることを示す軸のヒントが追加されます。

表示を少し回転させて、軸のキューの感覚を把握することができます。



17. もう一度、情報ウィンドウで axes0 オブジェクトをダブルクリックして、Axis タブを選択します。 プロパティのダイアログにはボックスの軸が表示されます。 マウスカーソルを軸の上に移動させると、軸を識別するツールチップが表示されます。

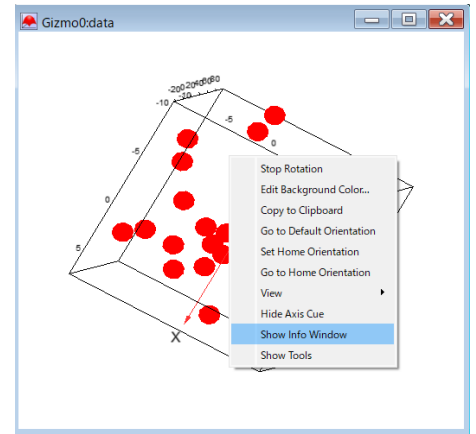


18. Gizmo0 Info ウィンドウのクローズボックスをクリックします。

Gizmo0 Info ウィンドウが隠れます。

通常、3D プロットの作成や調整を行っている間は表示しておき、プロットの表示のみを行いたい場合は非表示にすることができます。

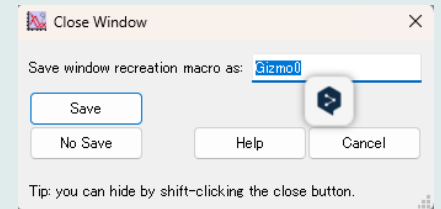
メニュー Gizmo→Show Info を選択するか、Gizmo0 ウィンドウを右クリックして Show Info Window を選択することで、いつでも表示することができます。



19. Gizmo0 ウィンドウのクローズボックスをクリックします。

Close Window ダイアログが表示され、ウィンドウを再作成するマクロとして保存するかを聞いてきます。

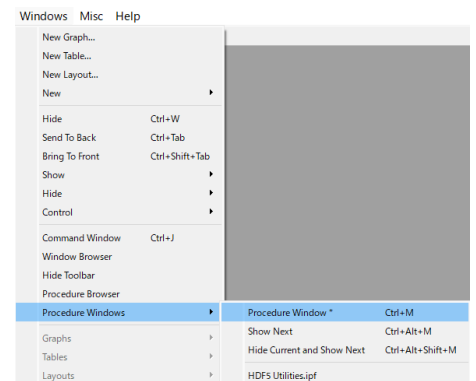
これは、グラフの再作成マクロと同じように動作します。



20. Save ボタンをクリックしてウィンドウ再作成マクロを保存します。

再作成マクロは、メインの Procedure Window に保存されます。

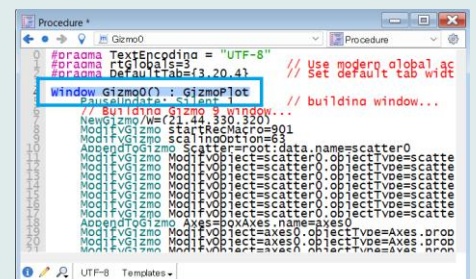
メニュー Windows→Procedure Windows→Procedure Window を選択します。



21. これは、Gizmo0 再作成マクロを含むメイン Procedure Window を表示します。冒頭部分を見るには、上までスクロールする必要があるかもしれませんが、冒頭は次のようになっています：

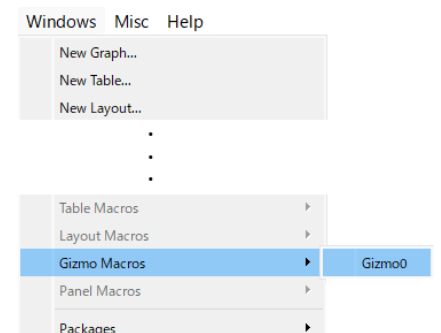
```
Window Gizmo0() : GizmoPlot
```

クローズボックスをクリックして Procedure Window を閉じます。



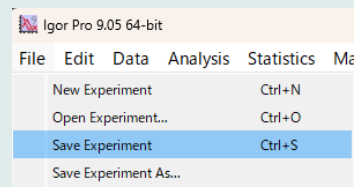
22. メニュー Windows→Gizmo Macros→ Gizmo0 を選択します。

Igor は、Gizmo0 再作成マクロを実行し、Gizmo0 ウィンドウを再表示します。



23. メニュー File→Save Experiment を選択して、「Gizmo 3D Scatter Plot Tour.pxp」として、エクスペリメントを保存します。

これは、後でツアーを再度確認したい場合に備えたもので、厳密には必要ありません。



これで、3D 散布図の作成は完了です。

気づいているかもしれませんが、まだ紹介していない散布図や軸のオプションが数多くあります。

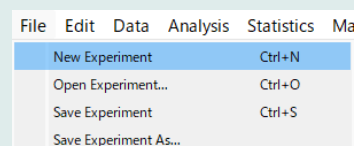
Gizmo Info ウィンドウ内の scatter0 と axes0 アイコンをダブルクリックすることで、この時点で実行することができます。

または、保存しておいて、ガイドの次のセクションに進むこともできます。

Gizmo サーフェスプロットツアー

このツアーでは、Z 値を含む 2D ウェーブを作成し、それをサーフェスプロットとしてプロットします。

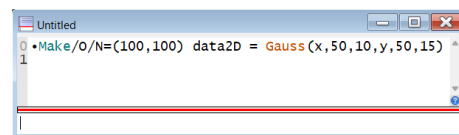
1. メニュー File→New Experiment を選択して、新しいエクスペリメントを作成します。



2. 次のコマンドを実行して、Z 値を持つ 2D ウェーブを作成します。

```
Make/O/N=(100,100) data2D = Gauss(x,50,10,y,50,15)
```

このマトリックスは、規則的なグリッド上の Z 値を表しています。



最初のツアーで見たように、このデータを Gizmo に表示するもっとも簡単な方法は、Data Browser で適切なウェーブを右クリックし、メニューから Gizmo Plot を選択することです。

同じショートカットは、Z 値の行列でも使うことができます。

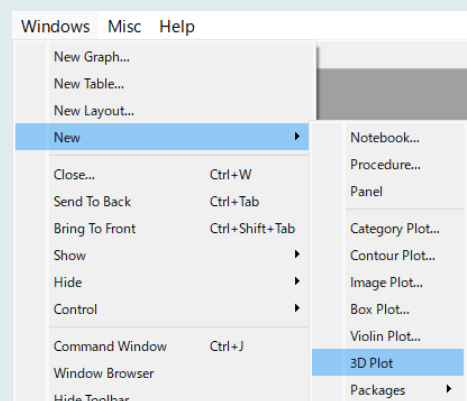
しかし、Gizmo についてより深く理解してもらうため、あえて難しい方法で説明します。

3. メニュー Windows→New→3D Plot を選択します。

コマンドウィンドウの履歴領域に次のコマンドがあることに注目してください

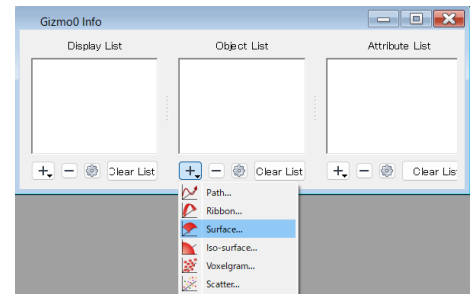
```
NewGizmo
```

Igor は、Gizmo0 の空のウィンドウと Gizmo0 Info ウィンドウも作成しました。



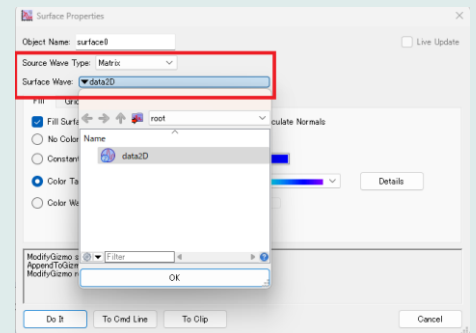
4. Gizmo0 Info ウィンドウの Object List の下にある「+」アイコンをクリックし、Surface を選択します。

Surface Properties ダイアログが表示されます。



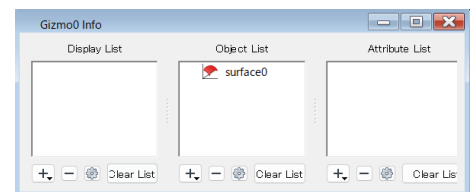
5. Source Wave Type ポップアップメニューから Matrix を選択し、Surface Wave ポップアップメニューから data2D を選択します。

他にもいくつかのオプションがありますが、今回はデフォルトのままにしておきます。



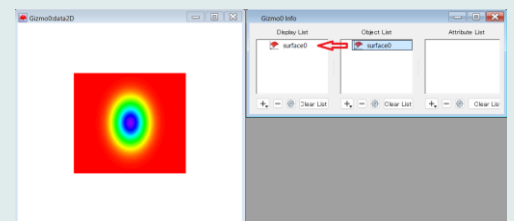
6. Do It をクリックします。

Surface0 という名前のサーフェスオブジェクトが作成され、Info ウィンドウの Object List に追加されました。
まだ Display List には追加していないため、Gizmo0 ウィンドウには表示されていません。

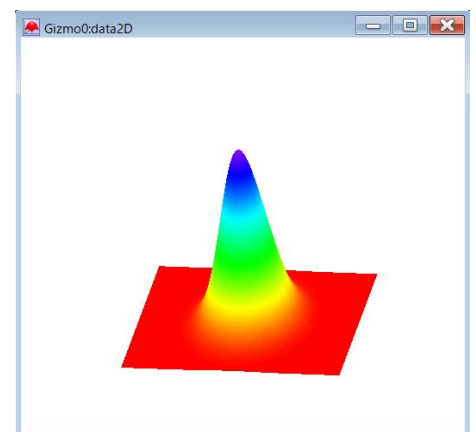


7. surface0 オブジェクトを Object List から Display List にドラッグします。

サーフェスが Gizmo0 ウィンドウに表示されます。
現在は上面から見ています。

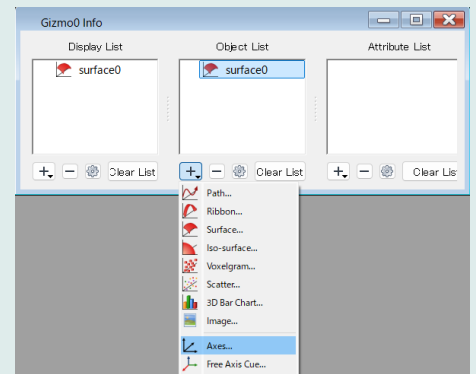


8. マウスを使って、Gizmo0 ウィンドウ内のサーフェスを回転させ、ガウスピークの側面が見えるように向きを変えます。



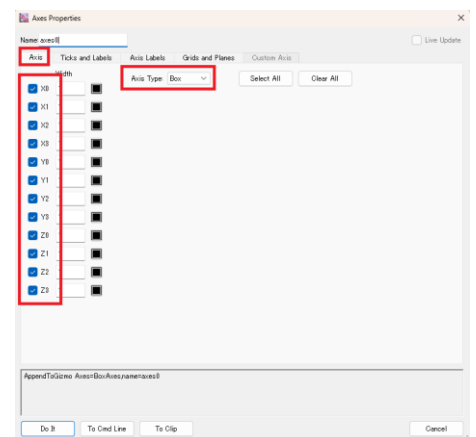
9. Gizmo0 Info ウィンドウで、Object List の下にある「+」アイコンをクリックし、Axes を選択します。

Axes Properties ダイアログが表示されます。



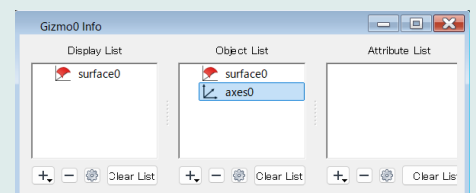
10. 選択されていない場合は、Axis タブを選択します。

Axis Type ポップアップメニューを Box に設定し、すべての軸のチェックボックス (X0、X1、…Z2、Z3) にチェックを入れます。



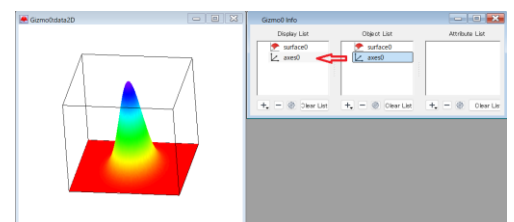
11. Do It をクリックします。

Igor は、axes0 という名前の axes オブジェクトを作成し、それを情報ウィンドウの Object List に追加しました。まだ Display List には追加していないため、Gizmo0 ウィンドウには表示されていません。



12. axes0 オブジェクトを Object List から Display List にドラッグします。

サーフェスプロットの周りに箱型の軸が追加されました。

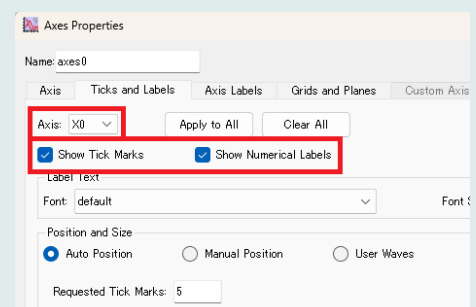


13. Object List または Display List のどちらかで、axes0 オブジェクトをダブルクリックします。

Axes Properties ダイアログが再び開きます。

Ticks and Labels タブをクリックします。
X0、Y0、Z0 軸の目盛と目盛ラベルをオンにします。

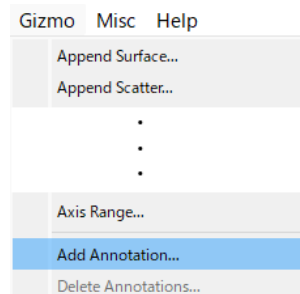
これは前のツアーで行ったことと同じです。



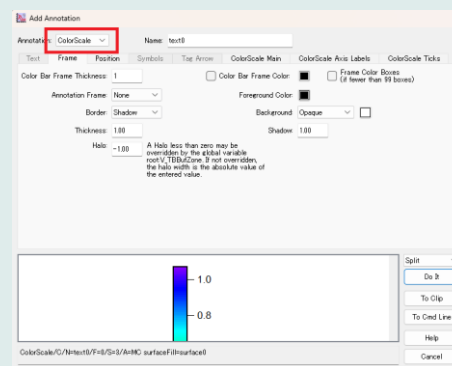
14. 次にカラースケールの注釈を追加します。

メニュー Gizmo→Add Annotation を選択します。

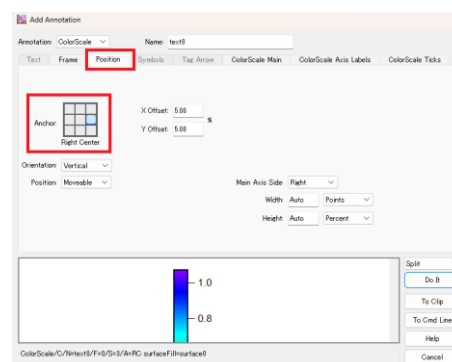
Add Annotation ダイアログが表示されます。



15. ダイアログの左上にある Annotation ポップアップメニューから、ColorScale を選択します。

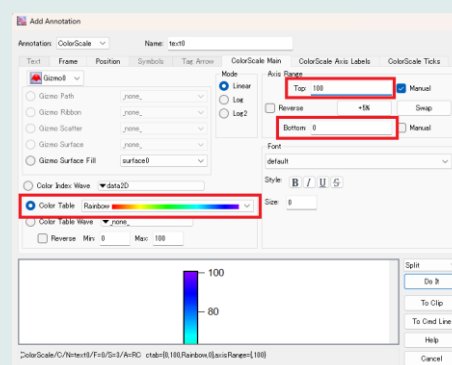


16. Position タブをクリックし、Anchor のセクションを Right Center に設定します。



17. ColorScale Main タブをクリックし、次のコントロールを設定します。

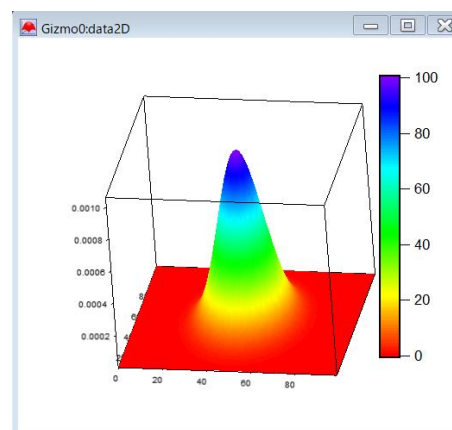
Color Table: Rainbow
Axis Range/Top: 100
Axis Range/Bottom: 0



18. Do It をクリックします。

カラースケールの注釈が作成されます。

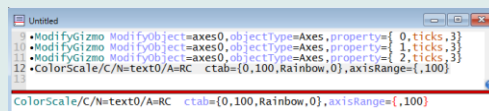
目盛ラベルが、サーフェスプロットの Z 軸目盛ラベルと一致しないことに注意してください。
データの範囲に合わせてカラースケールの範囲を設定する必要があります。
カラースケールを作成したコマンドを再実行することでこれを実行します。



19. コマンドウィンドウの履歴エリアにある最後のコマンドをクリックし、Enter キーを押してコマンドラインにコピーします。

コマンドラインには、次のように表示されているはずです（設定した項目により異なる場合があります）：

```
ColorScale/C/N=text0/A=RC
ctab={0,100,Rainbow,0},axisRange={0,100}
```



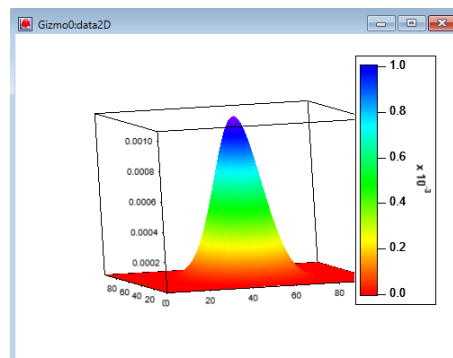
20. 次に示すように ctab を変更し、Enter キーを押して実行します。

```
ColorScale/C/N=text0/A=RC
ctab={WaveMin(data2D),WaveMax(data2D),Rainbow,0}
```

これで、カラースケールの目盛とサーフェスプロット内の z 軸目盛ラベルが一致しました。

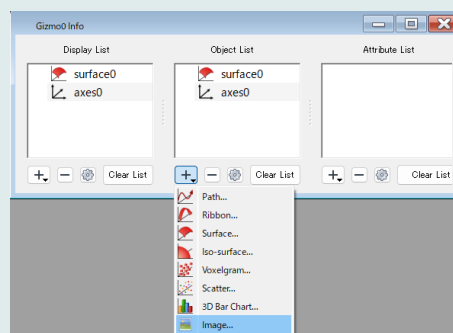
カラースケールの最大が 100 であり、対応が見にくいため、追加で次を実行することを勧めます。

```
ColorScale/C/N=text0 ctab={0,0.0011,Rainbow,0}
ColorScale/C/N=text0 axisRange={,0.001}
```



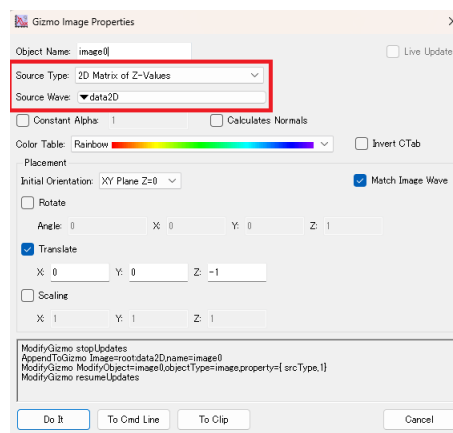
21. Gizmo0 Info ウィンドウの Object List の下にある「+」アイコンをクリックし、Image を選択します。

Gizmo Image Properties ダイアログが表示されます。

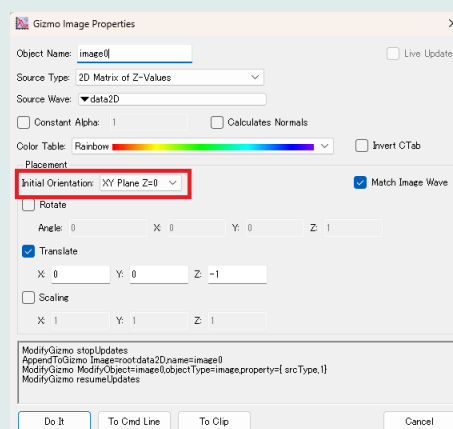


22. Source Type を 2D Matrix of Z-Values に設定し、Source Wave ポップアップメニューから data2D を選択します。

サーフェスプロットと画像の両方で、同じウェーブをソースとして使います。

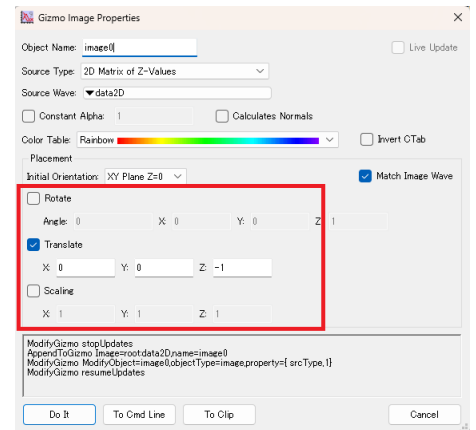


23. Initial Orientation ポップアップメニューから、XY Plane Z=0 を選択します。



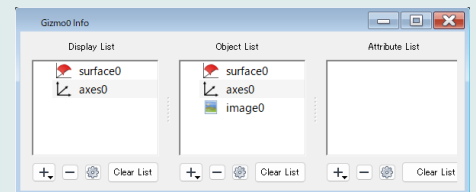
24. Translate 以外のチェックボックスをすべてオフにし、Translate の X、Y、Z 成分をそれぞれ 0、0、-1 に設定します。

Z 方向への変換は、画像をディスプレイボリューム（表示の 3 次元空間）の中心からディスプレイボリュームの面に移動させ、サーフェスプロットに配置します。



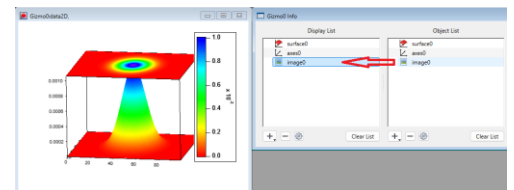
25. Do It をクリックします。

image0 という名前の画像オブジェクトが作成され、情報ウィンドウの Object List に追加されます。まだ Display List には追加していないため、Gizmo0 ウィンドウには表示されていません。

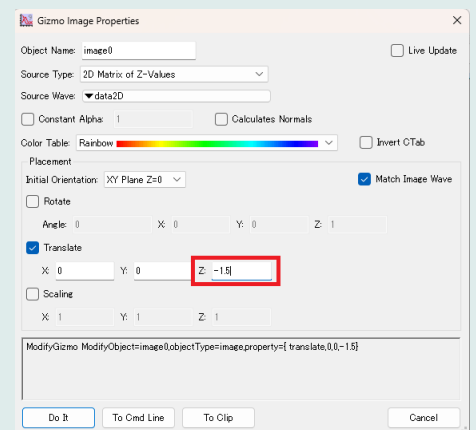


26. image0 オブジェクトを Object List から Display List にドラッグします。

サーフェスプロットに対応した画像プロットが表示されました。表示を確認するには、回転させる必要があるかもしれません。

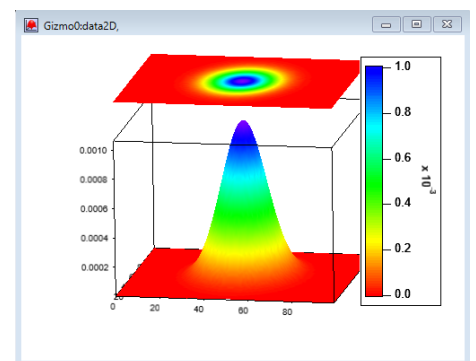


27. Object List の image0 オブジェクトをダブルクリックし、Translate Z を -1.5 に設定し、Do It をクリックします。



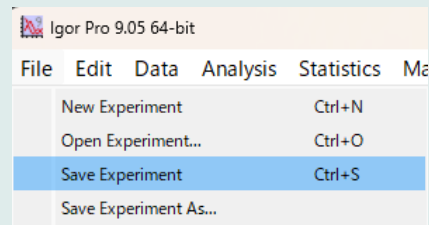
28. これにより、画像プロットとサーフェスプロットが分離され見やすくなります。

Translate パラメーターは、軸の単位ではなく、+/-1 のディスプレイボリュームの単位で示します。ディスプレイボリュームの単位は、特に球やボックスなどの描画扱うときに、多くの場面で使われます。



29. メニュー File→Save Experiment を選択して、エクスペリメントを「Gizmo Surface Plot Tour.pxp」として保存します。

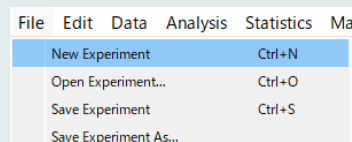
これは、後でツアーを再度確認したい場合に備えたもので、厳密には必要ありません。



Gizmo 3D 散布図&カーブフィッティングサーフェスツアー

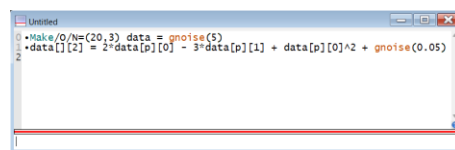
このツアーでは、トリプレットウェーブから 3D 散布図を作成し、カーブフィッティングを実行し、カーブフィット出力が元の散布データとどのように関連しているかを表すサーフェスを追加します。

1. メニュー File→New Experiment を選択して、新しいエクスペリメントを作成します。



2. 次のコマンドを実行して、XYZ 散布データを持つトリプレットウェーブを作成します。

```
Make/O/N=(20,3) data = gnoise(5)
data[][2] = 2*data[p][0] - 3*data[p][1] + data[p][0]^2
+ gnoise(0.05)
```

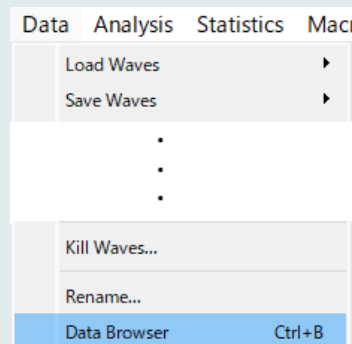


この散布データは、X と Y の多項式関数にほぼ等しい Z 値を持つ XY 平面内のランダムな位置を示しています。

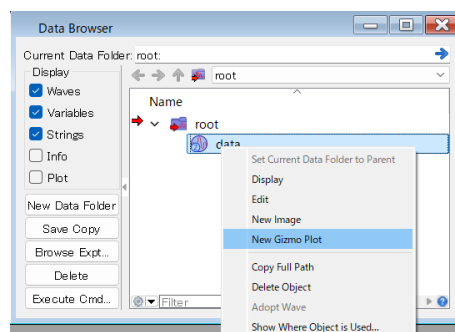
3. メニュー Data→Data Browser を選択します。

(既に開いているかもしれません)

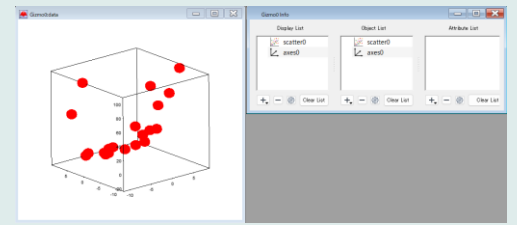
このデータを Gizmo に表示するもっとも簡単な方法は、Data Browser でこのデータを右クリックして Gizmo Plot を選択することです。



4. data ウェーブのアイコンを右クリックして、New Gizmo Plot を選択します。



5. Gizmo0 という名前の新しいウィンドウで、data ウェーブから Gizmo 3D 散布図が作成されます。

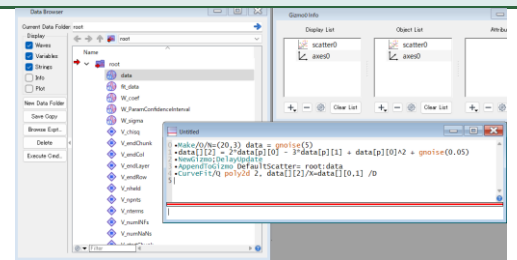


6. コマンドラインで次を実行します。

```
CurveFit/Q poly2d 2, data[][2]/X=data[][0,1] /D
```

Igor は 2D 多項式カーブフィッティングを実行し、出力ウェーブと変数を生成します。

主な出力は、fit_data ウェーブです。



7. Data Browser ウィンドウを閉じます。

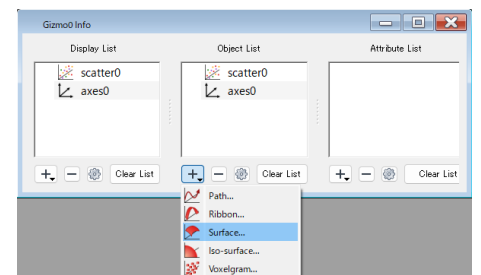
これは単に画面をすっきりさせるためです。

閉じなくても構いません。

次に、Gizmo プロットにサーフェスを追加します。

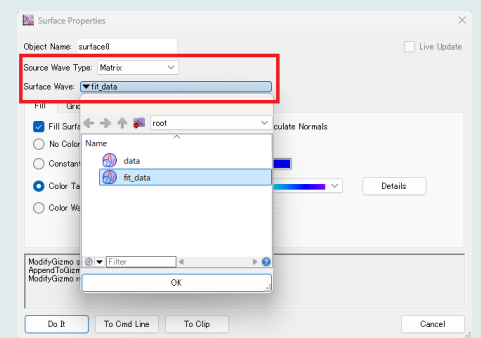
8. Gizmo0 Info ウィンドウの Object List の下の「+」アイコンをクリックし、Surface を選択します。

Surface Properties ダイアログが表示されます。



9. Source Wave Type ポップアップメニューから Matrix を選択し、Surface Wave ポップアップメニューから fit_data を選択します。

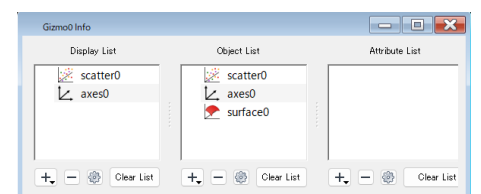
他にもいくつかのオプションがありますが、今回はデフォルトのままにしておきます。



10. Do It をクリックします。

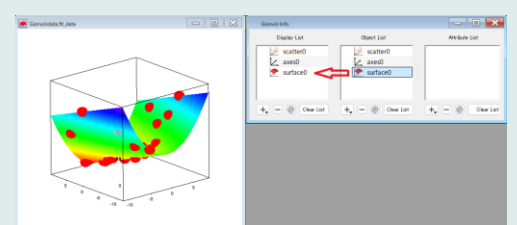
surface0 という名前の画像オブジェクトが作成され、情報ウィンドウの Object List に追加されます。

まだ Display List には追加していないため、Gizmo0 ウィンドウには表示されていません。

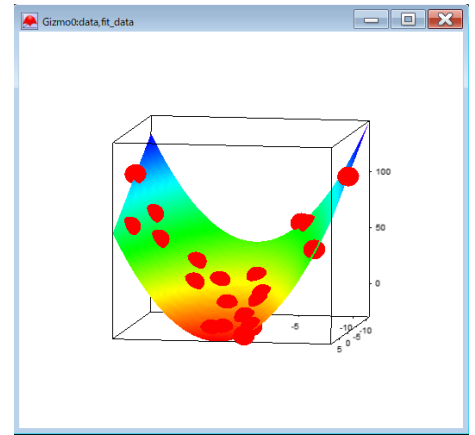


11. surface0 オブジェクトを Object List から Display List にドラッグします。

サーフェスが Gizmo0 ウィンドウに表示され、散布オブジェクトにうまくフィットしているように見えます。

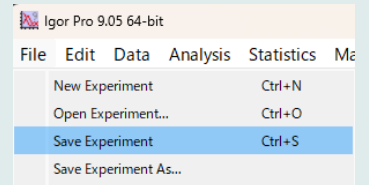


12. マウスを使って、Gizmo プロットの内容を回転させて、さまざまな角度からフィットぐあいを確認します。



13. メニュー File→Save Experiment を選択して、エクスペリメントを「Gizmo 3D Scatter Plot and Fitted Surface Tour.pxp」として保存します。

これは、後でツアーを再度確認したい場合に備えたもので、厳密には必要ありません。



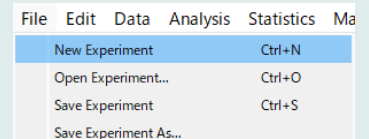
Voronoi 補間を使った Gizmo サーフェスツアー

Z 値の 2D 行列からサーフェスプロットを作る方法は、すでに説明しました。

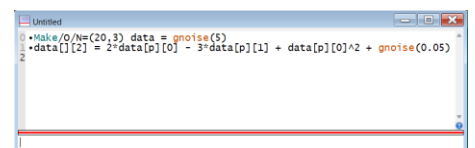
このツアーでは、XYZ 散布データの 3D サーフェス表示をプロットする方法を説明します。

このプロセスでは、Voronoi 補間を使って XYZ データの三角形分割を行います。

1. メニュー File→New Experiment を選択して、新しいエクスペリメントを作成します。



2. 次のコマンドを実行して、XYZ 散布データを持つトリプレットウェーブを作成します。



```
Make/O/N=(20,3) data = gnoise(5)
```

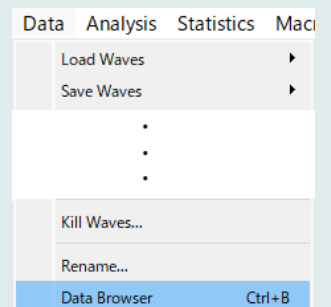
```
data[[2]] = 2*data[p][0] - 3*data[p][1] + data[p][0]^2  
+ gnoise(0.05)
```

この散布データは、X と Y の多項式関数にほぼ等しい Z 値を持つ XY 平面内のランダムな位置を示しています。

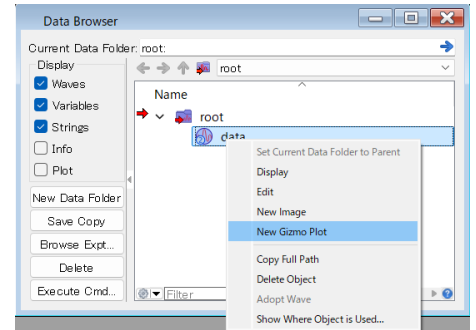
3. メニュー Data→Data Browser を選択します。

(既に開いているかもしれません)

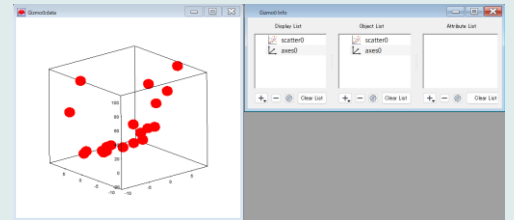
このデータを Gizmo に表示するもっとも簡単な方法は、Data Browser でこのデータを右クリックして Gizmo Plot を選択することです。



4. data ウェーブのアイコンを右クリックして、New Gizmo Plot を選択します。



5. Gizmo0 という名前の新しいウィンドウで、データウェーブから Gizmo 3D 散布図が作成されます。

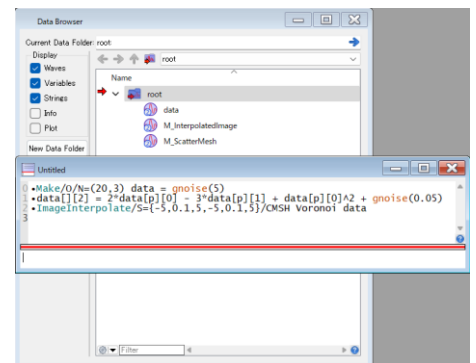


6. コマンドラインで次を実行します。

```
ImageInterpolate/S={-5,0.1,5,-5,0.1,5}/CM5H  
Voronoi data
```

Voronoi 補間は2つのウェーブ (M_ScatterMesh と M_InterpolatedImage) を作成します。

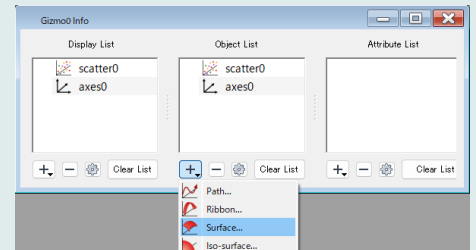
M_ScatterMesh は、散布データをフィッティングする 3D 空間内のポリゴンを一連の XYZ 座標で構成されています。



7. M_ScatterMesh を使って、3D プロットにサーフェスを追加します。

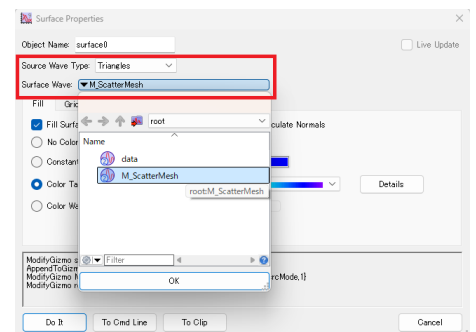
Gizmo0 Info ウィンドウの Object List の下にある「+」アイコンをクリックし、Surface を選択します。

Surface Properties ダイアログが表示されます。



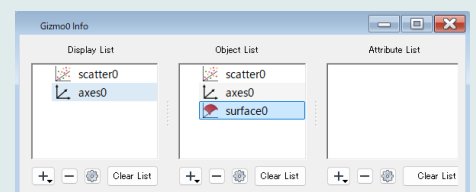
8. Source Wave Type ポップアップメニューから Triangles を選択し、Surface Wave ポップアップメニューから M_ScatterMesh を選択します。

他にもいくつかのオプションがありますが、今回はデフォルトのままにしておきます。



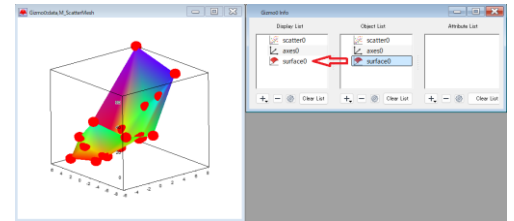
9. Do It をクリックします。

surface0 という名前の画像オブジェクトが作成され、情報ウィンドウの Object List に追加されます。まだ Display List には追加していないため、Gizmo0 ウィンドウには表示されていません。

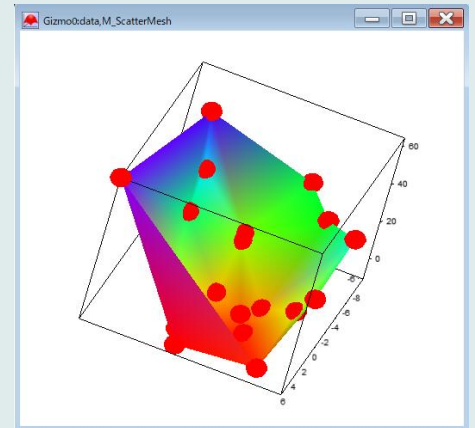


10. surface0 オブジェクトを Object List から Display List にドラッグします。

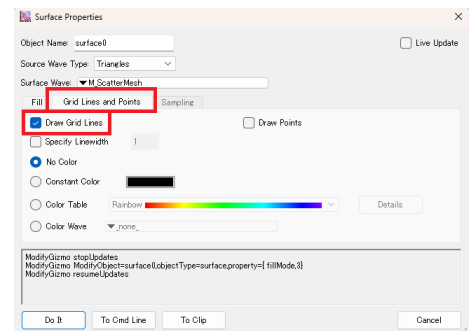
サーフェスが Gizmo0 ウィンドウに表示されます。



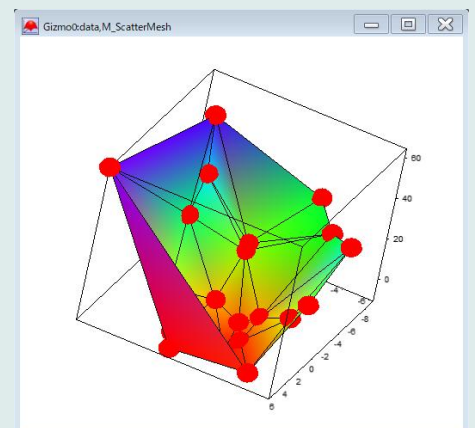
11. マウスを使って、Gizmo プロットの内容を回転させて、さまざまな角度からフィットぐあいを確認します。



12. Display List の surface0 オブジェクトをダブルクリックし、Grid Lines and Points タブをクリックし、Draw Grid Lines チェックボックスにチェックを入れ、Do It をクリックします。



13. これは Voronoi 補間によって作成されたポリゴンを M_ScatterMesh ウェーブで表示したものです。

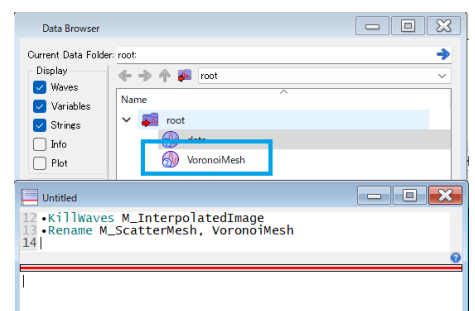


14. コマンドラインで次を実行して、クリーンアップします。

```
KillWaves M_InterpolatedImage  
Rename M_ScatterMesh, VoronoiMesh
```

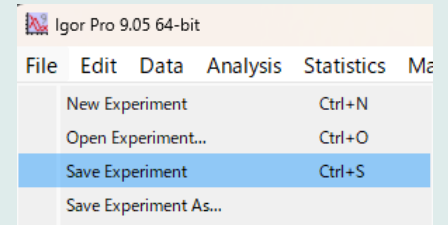
Igor が作成したデフォルトのウェーブ名を残しておく、後で同じデフォルトのウェーブ名を使う別のコマンドを実行しても、誤ってデータを上書きしてしまうことがありません。

また、M_InterpolatedImage ウェーブはもう不要です。



13. メニュー File→Save Experiment を選択して、エクスペリメントを「Gizmo Surface Using Voronoi Interpolation Tour.pxp」として保存します。

これは、後でツアーを再度確認したい場合に備えたもので、厳密には必要ありません。

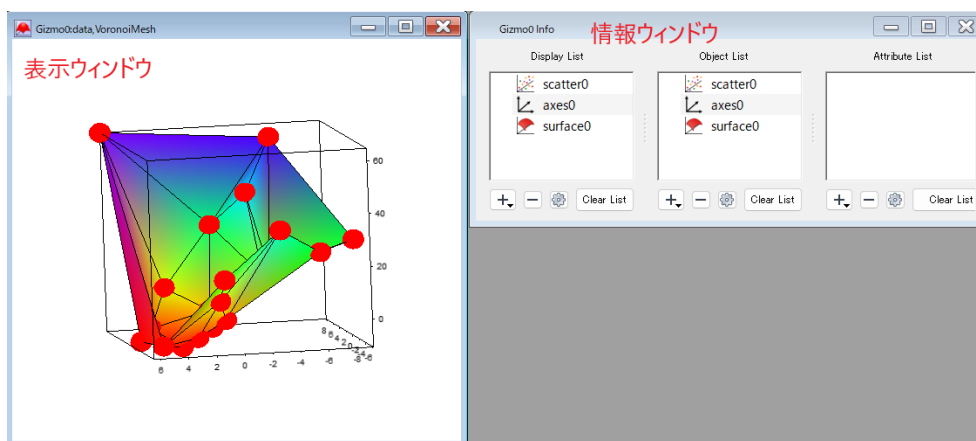


これで Gizmo のガイドツアーは終了です。

サンプルのエクスペリメントについては、File→Example Experiments→Visualization を選択してください。

Gizmo のウィンドウ

各 3D プロットに対して、Gizmo は表示ウィンドウと関連する情報ウィンドウを作成します。



表示ウィンドウには、回転可能な 3D オブジェクトが表示されます。

情報ウィンドウを使って、表示するオブジェクト、描画の順序、それらのプロパティをコントロールします。

必要のないときは、両方のウィンドウを隠して、画面をすっきりさせることができます。

また、Gizmo ウィンドウを通常のグラフと同様に終了して再作成することもできます。

Gizmo の表示ウィンドウはいくつでも作成できます。

複数の Gizmo の表示ウィンドウを開いたままにしておくと、いくつかの問題があります。

非アクティブな Gizmo 表示ウィンドウや非表示の Gizmo 表示ウィンドウもグラフィックリソースを消費するため、アクティブな Gizmo 表示ウィンドウに使われる可能性のあるリソースを消費してしまいます。

一部のノートパソコンでは、回転するオブジェクトを含む Gizmo の表示ウィンドウを閉じることによって、消費電力を削減できる場合があります。

使っているハードウェアによっては、使っていない Gizmo の表示ウィンドウを再作成マクロとして保存すると便利かもしれません。

文章を簡潔にするために、Gizmo 表示ウィンドウを「Gizmo ウィンドウ」と表記することがあります。

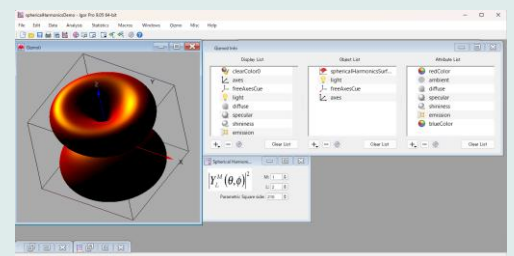
Gizmo 情報ウィンドウを「Gizmo 情報ウィンドウ」または「情報ウィンドウ」と表記します。

Gizmo 表示ウィンドウ

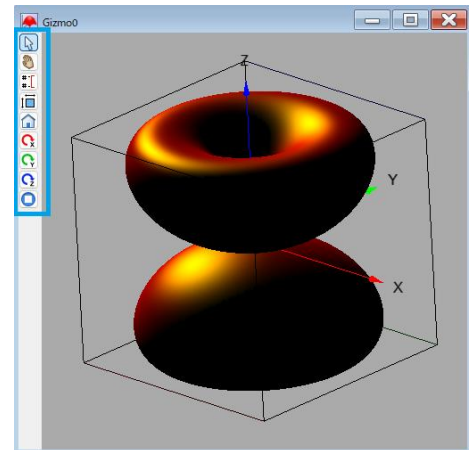
Gizmo 表示ウィンドウは、関連する Gizmo 情報ウィンドウの表示リストで指定されたとおりに、ウェーブと 3D 描画プリミティブを表すオブジェクトを回転可能な 3D 表示で表示します。

サンプルのエクスペリメントを使って説明します。

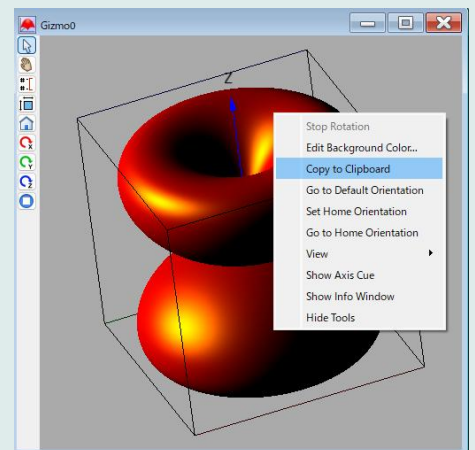
1. メニュー File→Example Experiments→Visualization
→sphericalHarmonicsDemo を選択します。



2. メニュー Gizmo→Show Tools を選択すると、ツールパレット（表示ウィンドウの左側）を表示できます。



3. Gizmo 表示ウィンドウの外観を変更するには、表示ウィンドウ内で右クリックしてポップアップメニューにアクセスします。



ウィンドウ内のシーンを回転させるには、ウィンドウの中心に仮想のトラックボールを配置し、それを回転させるように、マウスをクリックしたままドラッグします。

また、ツールパレット、マウスホイール、カーソルキー、キーボードの x, y, z キーを使ってシーンを回転させることもできます。

Gizmo 表示ウィンドウのツールパレット

Gizmo ツールパレットを表示するには、メニュー Gizmo→Show Tools を選択するか、表示ウィンドウを右クリックして、Show Tools を選択します。

上から順に次のアイコンがあります。



矢印ツール

矢印ツールが選択されている場合、表示ウィンドウの本体をドラッグすると、3D シーンが回転します。

矢印ツールとハンドツールは同時に使うことはできません。

ハンドツール

ハンドツールが選択されている場合、表示ウィンドウの本体をドラッグすると、3D シーンがパニングします。

矢印ツールとハンドツールは同時に使うことはできません。

軸ツール

軸ツールをクリックすると、Axis Range ダイアログが表示されます。

これは、メニュー Gizmo→Axis Range を選択するのと同じです。

アスペクト比ツール

アスペクト比のモードを切り替えます。

アスペクト比モードがオフの場合、各軸の長さは同じです。

アスペクト比モードがオンの場合、各軸の長さは、その軸に対して表示されるデータの範囲に比例します。

ホームツール

ホームツールをクリックすると、X、Y、Z の回転角度が 0 に設定されるか、または「ホーム」として指定した他の方向に向きます。

X 軸を中心に回転

X 軸を中心に回転ツールをクリックすると、X 軸を中心に 3D シーンが回転し始めます。

停止するには、停止ツールをクリックするか、表示ウィンドウの本体を 1 回クリックします。

Y 軸を中心に回転

Y 軸を中心に回転ツールをクリックすると、Y 軸を中心に 3D シーンが回転し始めます。

停止するには、停止ツールをクリックするか、表示ウィンドウの本体を 1 回クリックします。

Z 軸を中心に回転

Z 軸を中心に回転ツールをクリックすると、Z 軸を中心に 3D シーンが回転し始めます。

停止するには、停止ツールをクリックするか、表示ウィンドウの本体を 1 回クリックします。

停止ツール

停止ツールをクリックすると、すべての回転が停止します。

Gizmo 表示ウィンドウのコンテキストメニュー

Gizmo 表示ウィンドウのコンテキストメニューには、一般的なタスクのショートカットが表示されます。

上から下に向かって、次の項目が含まれます。

Stop Rotation	
Edit Background Color...	
Copy to Clipboard	
Go to Default Orientation	
Set Home Orientation	
Go to Home Orientation	
View	▶
Show Axis Cue	
Show Info Window	
Hide Tools	
Match Window Size	▶
Rotate to Match	▶
Sync to Gizmo	▶

Stop Rotation

3D シーンの回転を停止します。

Edit Background Color

Gizmo ウィンドウの背景色を設定します。

Copy to Clipboard

Edit メニューの Export Graphics ダイアログで設定したフォーマットを使って、Gizmo プロットをクリップボードにコピーします。

Go to Default Orientation

新しい Gizmo ウィンドウが最初に作成されるときに有効になる 3D 空間の方向に戻ります。

Set Home Orientation

Go to Home Orientation が選択されたとき、またはツールパレットのホームアイコンがクリックされたときに戻る先として、現在の方向をホームとして保存します。

Go to Home Orientation

3D シーンを Home Orientation に戻します。

View

3D シーンをいくつかのプリセット方向の 1 つに回転します。

Show Axis Cue

X、Y、Z 方向を示す矢印を表示します。

Show Info Window

アクティブな Gizmo 表示ウィンドウに関連した Gizmo 情報ウィンドウを表示します。

Show/Hide Tools

Gizmo ツールパレットの表示／非表示を切り替えます。

Match Window Size

別の Gizmo 表示ウィンドウを、アクティブな Gizmo 表示ウィンドウと同じサイズに設定します。

この項目は、複数の Gizmo 表示ウィンドウがある場合にのみ表示されます。

Rotate to Match

別の Gizmo 表示ウィンドウを、アクティブな Gizmo 表示ウィンドウと同じ方向に回転させます。

この項目は、複数の Gizmo 表示ウィンドウがある場合にのみ表示されます。

Sync to Gizmo

現在の Gizmo プロットの回転を、別の Gizmo プロットと同じ向きに固定します。

別のプロットを回転させると、両方のプロットが同じ向きに回転します。

相互同期を行うには、各ウィンドウを他のウィンドウと同期するように設定する必要があります。

この項目は、複数の Gizmo 表示ウィンドウがある場合にのみ表示されます。

Gizmo 情報ウィンドウ

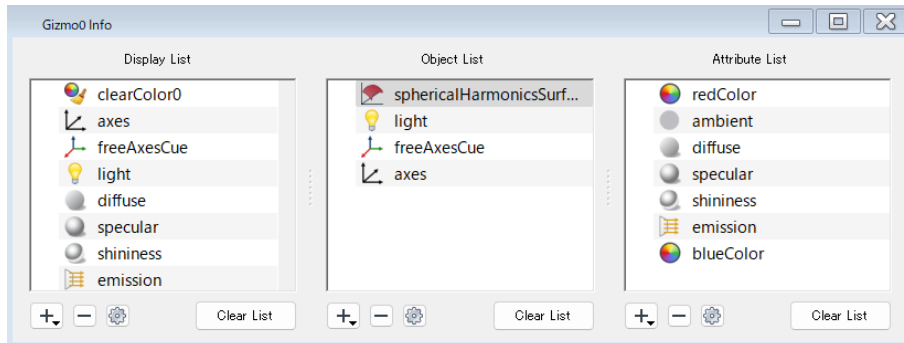
Gizmo 情報ウィンドウは、Gizmo 表示ウィンドウ内のオブジェクトの表示をコントロールするためのメインのユーザーインターフェイスです。

各情報ウィンドウには、結果のグラフを表示する Gizmo 表示ウィンドウが関連付けられています。

情報ウィンドウには3つのリストが含まれます：Display List、Object List、Attribute List です。

これらのリストを使って、オブジェクトを追加したり、Gizmo 表示ウィンドウでのオブジェクト外観を変更したりします。

実際に、Gizmo の表示ウィンドウに描画されるのは、Display List に表示されている項目のみです。

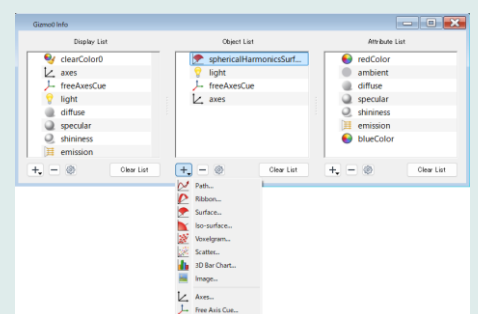


Object List の下にある「+」アイコンをクリックして、オブジェクトを作ることができます。

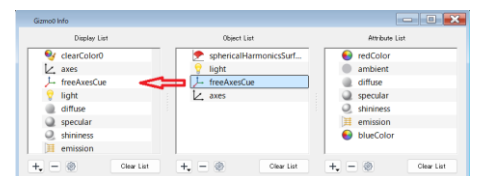
「+」アイコンをクリックすると、追加するオブジェクトの種類を選択するメニューが表示されます。

Attribute List の下にある「+」アイコンは属性を追加し、Display List の下にある「+」アイコンは処理を追加します。

Object List にはオブジェクトのみが含まれ、Attribute List には属性のみが含まれます。

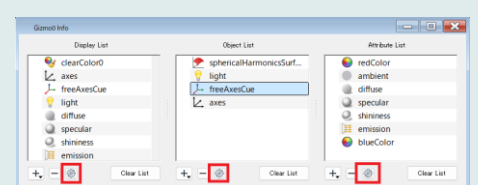


Display List には、Object List からドラッグしたオブジェクト、Attribute List からドラッグした属性、そしてコマンドを含めることができます。



リスト内のアイテムをダブルクリックするか、アイテムを選択してリストの下にある歯車アイコンをクリックすると、アイテムのプロパティを編集できます。

リストからアイテムを削除するには、アイテムを選択して Delete キーを押すか、リストの下での「-」アイコンをクリックします。



1つのエクスペリメントで2つ以上の Gizmo ウィンドウを開いている場合、2つ以上の情報ウィンドウを開くことができます。

1つの情報ウィンドウの Attribute List から属性をドラッグし、それを別の情報ウィンドウの Attribute List にドロップすることができます。

ある情報ウィンドウの Object List からオブジェクトをドラッグし、別の情報ウィンドウの Object List にドロップすることができます。

ある情報ウィンドウの Object List からオブジェクトをドラッグして、別の情報ウィンドウの Display List にドロップすると、受け取り側のウィンドウの Object List にも対応するエントリを追加します。

Igor の2つのインスタンス（別々に起動した Igor）に属する情報ウィンドウ間でオブジェクトをドラッグ&ドロップすることはできません。

Gizmo Object List

情報ウィンドウの中段のリストは Object List です。

このリストには、作成したオブジェクトがすべて表示され、Display List で使えるようになります。

Gizmo は、サーフェスプロットなどのウェーブベースのオブジェクトや、球体などの描画プリミティブを含む、多くの種類のオブジェクトをサポートしています。

Object List の下にある「+」アイコンをクリックすると、使うことができるオブジェクトタイプのメニューが表示されます。

詳細は「Gizmo オブジェクト」もセクションを参照してください。

オブジェクトを Gizmo プロットに表示するには、Display List にドラッグして追加する必要があります。

Gizmo Display List

Display List は、Gizmo の表示ウィンドウに表示される内容をコントロールします。

Gizmo は、Display List 内の項目を上から順番に処理します。

Object List からドラッグしたオブジェクトと、Attribute List からドラッグした属性に加えて、Display List に次のコマンドを追加することができます：

ClearColor、ColorMaterial、Translate、Rotate、Scale、Main Transformation、Enable、Disable、Ortho。

ColorMaterial の有効化と無効化の処理には、OpenGL に関するある程度の知識が必要です。

Main Transformation アイテムは、照明と組み合わせて使います。

これは「Gizmo 照明の位置」のセクションで説明しています。

Ortho 処理は、2D 空間を 2D スクリーンに投影するコントロールを行います。

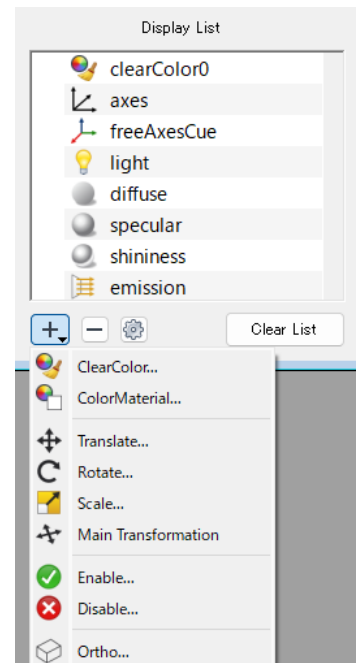
これは「Gizmo の投影」のセクションで説明しています。

OpenGL に精通している方は、Gizmo が自動的に生成するいくつかの OpenGL 命令（表示変換、デフォルトの照明など）がリストに表示されないことに注意してください。

ユーザー独自の設定を指定すると、さまざまなデフォルトの設定は実行されません。

例えば、デフォルトでは、Gizmo はシーンを照らすためにニュートラルな環境光を使います。

しかし、Display List に 1 つ以上の照明を追加すると、デフォルトの環境光は省略されます。



Gizmo Display List 内のアイテムの順序

Display List 内のアイテムをドラッグ&ドロップして希望の位置に移動させることで、表示順を変更することができます。

プロットの表示を決定する OpenGL 描画命令の順序を決定するため、Display List 内の項目の順序は重要です。

これは Translate、Rotate、Scale などの処理を行うと明らかになります。

リスト内の位置が問題とならないアイテムもいくつかありますが、大半のケースでは、アイテムの順序を変更すると明らかな表示の変化が見られます。

例えば、Rotate と Translate の処理の順序を入れ替えると、全く異なる結果となります。

例は「Gizmo オブジェクトの回転」のセクションを参照してください。

Gizmo Attribute List

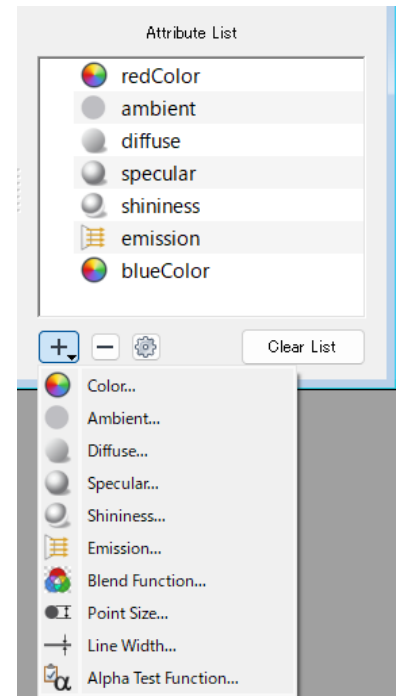
Attribute List は情報ウィンドウの右側に表示されます。

「+」アイコンをクリックして、必要な属性の種類を選択することで、属性を作成します。

その属性をグローバル属性として Display List にドラッグするか、埋め込み属性として Object List のアイテムのアイテムにドラッグします。

Attribute List 内の項目の順序は重要ではありません。

属性の詳細は、「Gizmo の属性」のセクションで説明しています。



Gizmo オブジェクト

Gizmo オブジェクトには、主に 5つのカテゴリがあります：ウェーブベースのオブジェクト、軸オブジェクト、描画プリミティブオブジェクト、照明、その他のオブジェクトです。

「データオブジェクト」とも呼ばれるウェーブベースの Gizmo オブジェクトは、ウェーブからデータを取得し、以下のタイプを含みます：

- Path Plot (パス [経路] プロット)
- Ribbon Plot (リボンプロット)
- Surface Plot (サーフェスプロット)
- Isosurface Plot (アイソサーフェスプロット)
- Voxelgram Plot (ボクセルグラムプロット)
- 3D Scatter Plot (3D 散布図)
- 3D Bar Plot (3D 棒グラフ)
- Gizmo Image Plot (Gizmo 画像プロット)

軸オブジェクトは以下のタイプを含みます：

- Axis Objects (軸オブジェクト)
- Axis Cue Objects (軸方向矢印オブジェクト)

描画プリミティブオブジェクトは以下のタイプを含みます：

- Line Objects (線オブジェクト)
- Triangle Objects (三角形オブジェクト)
- Quad Objects (四角形オブジェクト)
- Box Objects (箱オブジェクト)
- Sphere Objects (球オブジェクト)
- Cylinder Objects (円柱オブジェクト)
- Disk Objects (円盤オブジェクト)
- Tetrahedron Objects (四面体オブジェクト)
- Pie Wedge Objects (パイの切れ端 [楔形] オブジェクト)

照明オブジェクトタイプは1つだけです：

- Light Objects (「Gizmo の色、素材、照明」のセクションを参照)

その他のオブジェクトタイプには以下のものがあります：

- Group Objects
- Texture Objects
- Matrix4x4 Objects

指定された Gizmo 表示ウィンドウの情報ウィンドウの Object List の下にある「+」アイコンをクリックしてオブジェクトを作ります。

特定のタイプのオブジェクトには、オブジェクトを最初に作る時に編集可能な内部プロパティがあります。Object List または Display List でオブジェクトをダブルクリックすると、後でも編集することができます。

Gizmo オブジェクトを作成すると、Object List に追加されます。
Display List にドラッグするまでは表示ウィンドウには表示されません。
指定したオブジェクトを Display List に複数回ドラッグすることができます。
これにより、毎回新しい表示オブジェクトが作成されます。

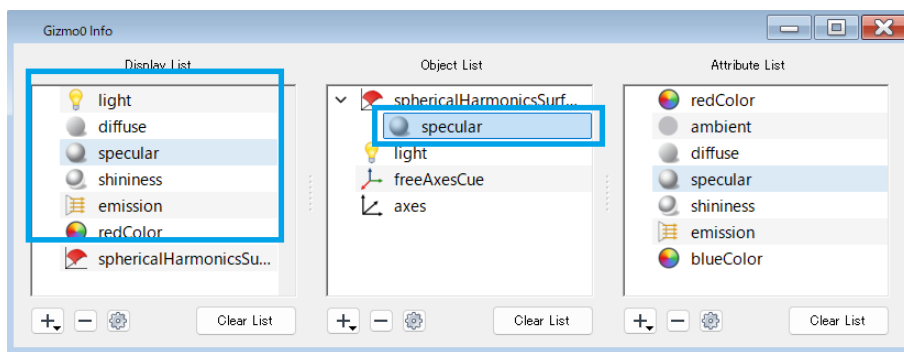
Gizmo の属性

Gizmo の属性は、グローバル属性として Gizmo Display List に適用したり、埋め込み属性として特定の Gizmo オブジェクトに適用したりできる設定をカプセル化します。

Gizmo は以下のタイプの属性をサポートしています：

- Color (色)
- Ambient (環境)
- Diffuse (拡散)
- Specular (鏡面)
- Shininess (光沢)
- Emission (放射)
- Blending (ブレンド)
- Point Size (ポイントサイズ)
- Line Width (線幅)
- Alpha test function (アルファテスト機能)

情報ウィンドウ内の Attribute List を使って属性を作成します。
次に、その属性をグローバル属性として Display List にドラッグするか、または埋め込み属性として Object List 内のオブジェクトにドラッグします。



グローバル属性と埋め込み属性に加えて、（Igor 7 以降では）次のセクションで説明する内部属性があります。

内部属性

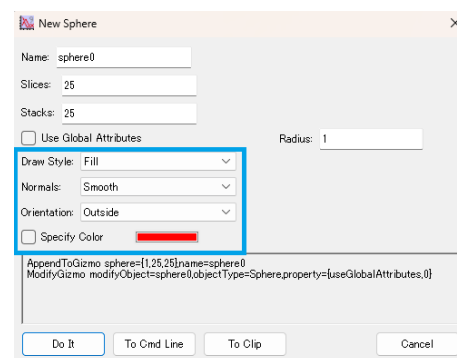
内部属性はオブジェクトに組み込まれています。

例えば、New Sphere ダイアログは右のようになります。

Draw Style、Normals、Orientation、色の設定は、Sphere オブジェクトの内部属性です。

Use Global Attributes チェックボックスは、その下のコントロールを無効にし、該当するオブジェクトのそれぞれのグローバル属性を使います。

他のグローバル属性の使用には影響しません。



Specify Color チェックボックスは、色についても同じことを行います。

チェックされていない場合、オブジェクトに固有の色はありません。

この場合、Display List 内でこのオブジェクトの前に Color Material 処理と Color 属性を追加する必要があります。

Specify Color にチェックを入れると、Gizmo はオブジェクトのデフォルトの Color Material を作成し、指定された内部 Color 属性を使います。

（Igor 7 以前の Gizmo では内部属性をサポートしていなかったため、グローバル属性または埋め込み属性を使う必要がありました。グローバル属性や埋め込み属性よりも、プレファレンスで設定できる場合は内部属性を使うことを推奨します。）

グローバル属性

属性を Display List にドラッグすると、グローバル属性として機能し、Display List のそれより後（リストの下）のすべてのオブジェクトに影響します。

Display List に Color などの属性を配置すると、OpenGL は内部に色指定のないオブジェクトをすべてこのグローバルな色指定を使って描画します。

また、適用した色を確認するには、Color Material も必要です。

内部属性と埋め込み属性はグローバル属性を上書きします。

埋め込み属性

埋め込み属性は推奨ではなく、主に後方互換性を維持するためにサポートされています。

新しいプロジェクトでの使用は推奨しません。

プリミティブなオブジェクトには独自の内部属性が用意されているため、そちらを使うべきです。

Object List でオブジェクトの上に属性をドラッグすると、そのオブジェクトに埋め込まれます。

任意の数のオブジェクトに任意の属性を埋め込むことができ、また任意の数の属性を任意のオブジェクトに埋め込むことができます。

例えば、Sphere オブジェクトを作成し、それを青く表示させたい場合、Attribute List で青の属性を作成し、それを Object List の Sphere オブジェクトの上にドロップします。

Sphere オブジェクトの内部の Color 属性を直接設定するのではなく、そうすることのメリットは、複数のオブジェクトで同じ Color 属性を再利用できることです。

このアプローチでは、1つの属性を変更するだけで、関連するすべてのオブジェクトの色を変更することができます。

内部属性は、埋め込み属性とグローバル属性を上書きします。

属性が埋め込まれたオブジェクトが描画されると、Gizmo は最初に描画環境の状態を保存します。

次に、オブジェクトが描画される直前に、埋め込み属性を実行し、最後に描画環境の状態を復元します。

その結果、埋め込み属性は、埋め込まれたオブジェクトのみに影響します。

あるオブジェクトに矛盾する属性を適用した場合、オブジェクトの表示に影響を与えるのは、埋め込まれたリストの最後の属性のみです。

例えば、赤色、青色、緑色の属性が埋め込まれた Sphere オブジェクトの場合、球は緑色で描画されます。