

CONTENTS

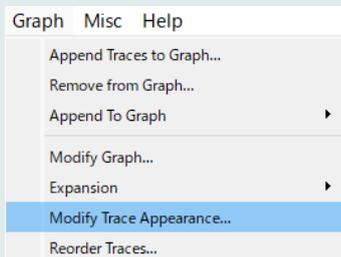
Volume II User's Guide: Part 1 (Graphs #2)	3
トレースの修正	3
修正するトレースの選択.....	3
トレースの表示モード.....	4
マーカー.....	5
マーカーの色.....	6
テキストマーカー.....	8
矢印マーカー.....	9
線のスタイルと幅.....	11
グラフの塗りつぶし.....	11
ウェーブフォームを持つ棒グラフ.....	12
XY ペアを持つ棒グラフ.....	12
グループ化、積み上げ、モードの追加.....	13
トレースの色.....	16
補助 (z) ウェーブからトレースのプロパティを設定.....	17
Set as f(z) ダイアログの Color as f(z) 設定.....	17
Set as f(z) ダイアログの Marker Size as f(z) 設定.....	19
Set as f(z) ダイアログの Marker Size as f(z) in Axis Units 設定.....	20
Set as f(z) ダイアログの Marker Number as f(z) 設定.....	21
Set as f(z) ダイアログの Pattern Number as f(z) 設定.....	21
Color as f(z) 凡例の例.....	22
トレースのオフセット.....	23
トレースの乗算.....	24
トレースを隠す.....	25
複素数表示モード.....	25
グラフのギャップ.....	26
エラーバー.....	27
エラー陰影.....	29
エラー楕円.....	30

Volume II User's Guide: Part 1 (Graphs #2)

トレースの修正

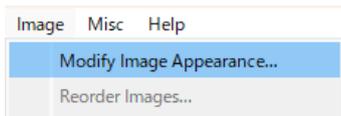
Igor Pro マニュアル : II-290 ページ以降をもとに編集

Graph メニューから **Modify Trace Appearance** を選択するか、グラフ内のトレースをダブルクリックすることで、グラフ内の各トレースの表示形式を指定することができます。



画像プロットの場合は、Image メニューから **Modify Image Appearance** を選択します。

Image メニューは画像プロットウィンドウを開いているときのみ表示されます。

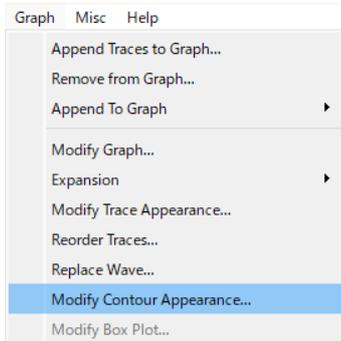


等高線プロットの場合は、**Modify Contour Appearance** を選択します。

この項目は等高線プロットウィンドウを開いているときのみ有効になります。

これを使うと、すべての等高線の表示形式を一度にコントロールできます。

ただし、等高線ラインを1つだけ目立たせたい場合は、Modify Trace Appearance ダイアログを使います。



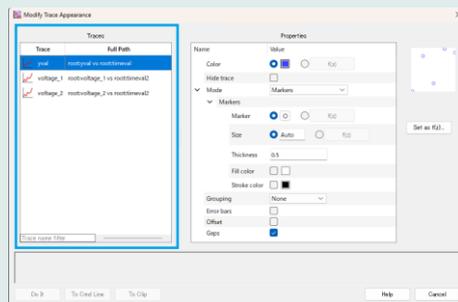
修正するトレースの選択

メニュー Graph → **Modify Trace Appearance** を選択します。

Modify Trace Appearance ダイアログの Traces リストから表示形式を変更したいトレースを選択します。

グラフ内のトレースをダブルクリックしてこのダイアログに移動した場合、そのトレースが自動的に選択されています。

複数のトレースを選択した場合、ダイアログ内の項目は、最初に選択したトレースの設定が表示されています。



選択を確定すると、選択したトレースの設定を変更することができます。

確定したら、別のトレースまたは複数のトレースを選択し、さらに変更することができます。

Igor は加えられた変更をすべて記憶しているのので、ダイアログを1回開いて設定するだけで全てを行うことができます。

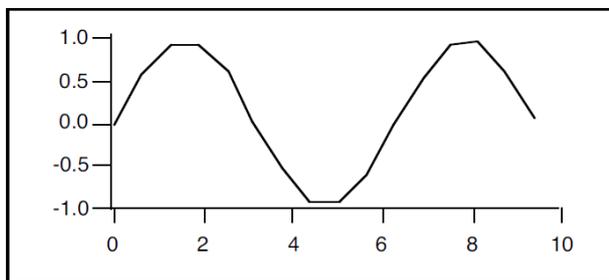
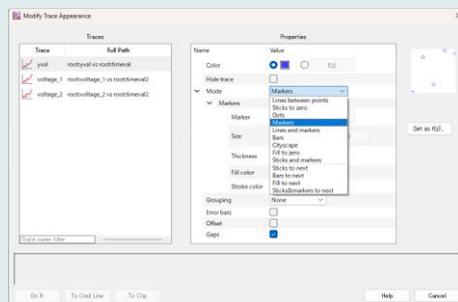
トレースの表示モード

メニュー **Graph** → **Modify Trace Appearance** を選択します。

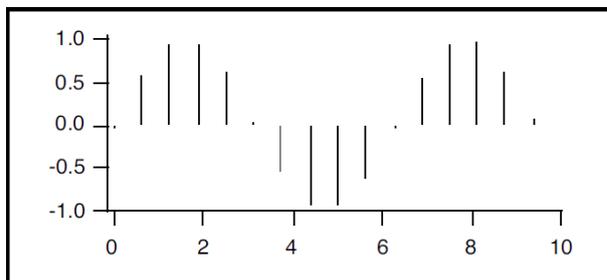
Mode ポップアップメニューから、選択したトレースの表示モードを選択します。

一般的に使われるモードは、点線、点、マーカー、線とマーカー、バーです。

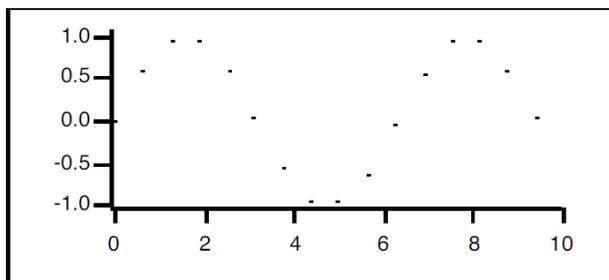
他のモードでは、 $Y=0$ または別のトレースに棒グラフ、塗りつぶしなどを描画することができます。



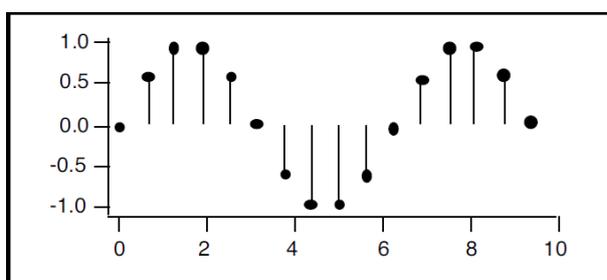
Lines between points mode



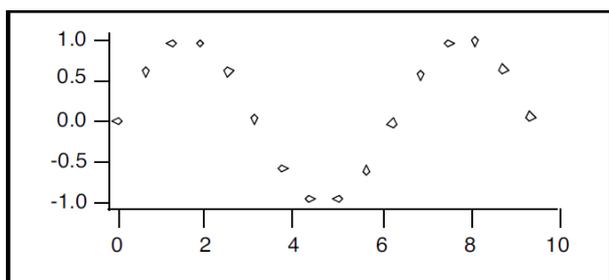
Sticks to zero mode



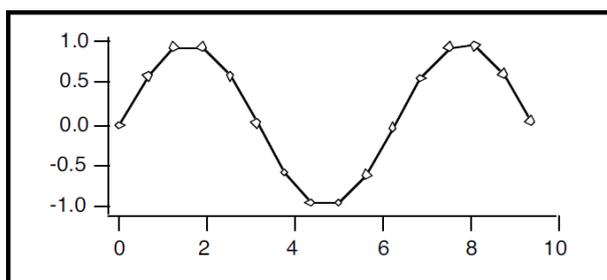
Dots mode



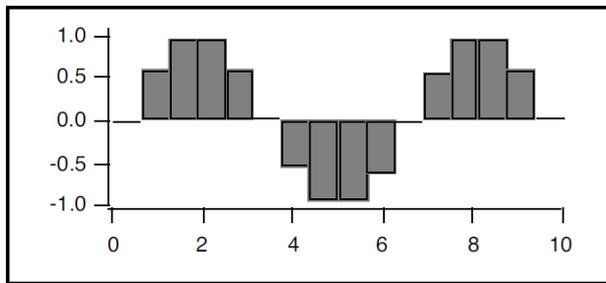
Sticks and markers mode



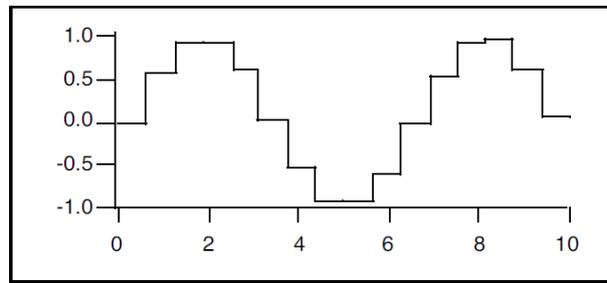
Markers mode



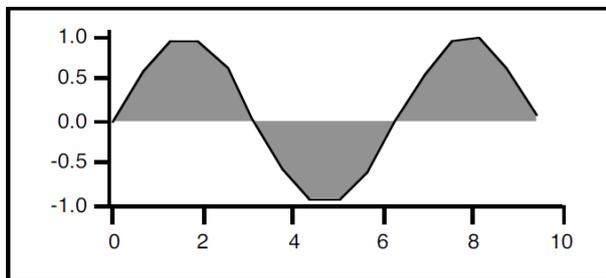
Lines and markers mode



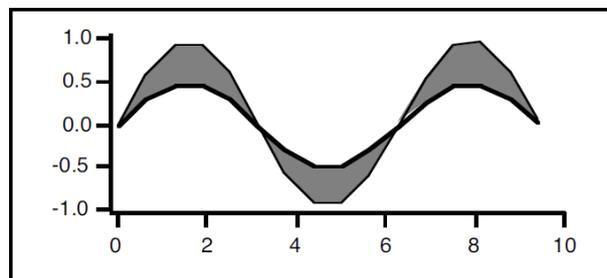
Bars mode



Cityscape mode



Fill to zero mode



Fill to next mode

マーカー

メニュー Graph → Modify Trace Appearance を選択します。

Marker または Lines and Markers モードを選択した場合、マーカー、マーカーのサイズ、マーカーの太さ、マーカーが透明か色付きかを選択することができます。

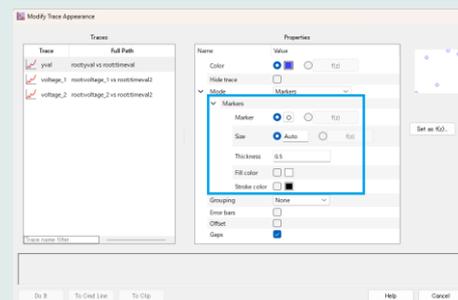
マーカーのサイズは、0 から 200 までの小数です。

0 は「自動」を意味し、グラフのサイズに適したマーカーサイズが選択されます。

マーカーの太さは、0 から 50 ポイントまでの小数で指定します。

小数点は、画面上では表示されない場合がありますが、エクスポートおよび印刷したグラフィックでは表示されます。

マーカーの太さを 0 にすると、マーカーが消えます。



マーカーのサイズと太さには相互作用があります。

マーカーを左右対称にする必要がある場合は、Igor がマーカーのサイズを調整します。

マーカーの未調整の幅と高さは、マーカーサイズの設定値を s とすると、 $2*s+1$ ポイントです。

マーカーと対応するマーカーコードの一覧表は次の通りです。



カスタムマーカーを作成することもできます。

setWindow コマンドの markerHook キーワードを参照してください。

マーカーの色

Igor では、グラフマーカーの色として、トレースの色、ストロークの色、塗りつぶしの色の3色をコントロールできます。

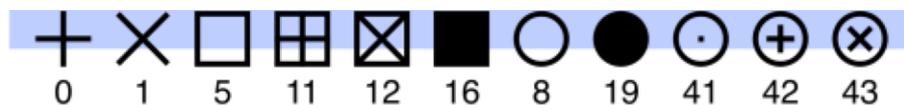
トレースの色は、トレース全体に対して選択された色であり、トレースモードに関係なく同じです。

塗りつぶしの色は、中が空のマーカーの背景領域を塗りつぶすために使われます。

デフォルトでは塗りつぶしの色がないため、マーカーの内側に後ろにあるオブジェクトが見えます。

以下はトレースの色を黒、マーカーサイズを 10、マーカーの太さを 2 ポイントに設定した Igor のマーカーのサンプルです。

青い長方形は、マーカーの後ろに描かれています。



青い長方形は、中が空のマーカー (5, 11, 12, 8, 41, 42, 43) の内部を通過しています。

トレースの色が黒なので、ソリッドのマーカー (16, 19) は、ソリッドの黒です。

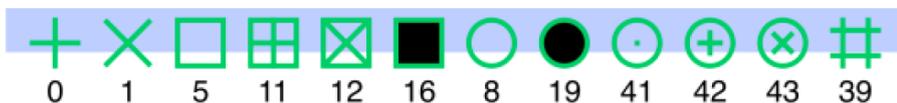
Modify Trace Appearance ダイアログを使って、ストローク（マーカーの線）の色を設定できます。

Stroke チェックボックスをオンにして、色を選択します。

また、ModifyGraph mrkStrokeRGB コマンドを使うこともできます。

次の図は、以下のコマンドを使ってストロークの色を緑にしたものです。

```
ModifyGraph mrkStrokeRGB=(1,52428,26586)
```



ストロークの色がトレースの色を上書きするので、すべてのマーカーの輪郭は緑になります。

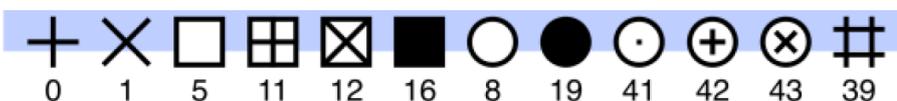
ソリッドのマーカーは黒で、緑の輪郭線が付いています。

中が空のマーカーの内側を不透明にすることができます。

Modify Trace Appearance ダイアログで、Fill チェックボックスをオンにします。

コマンドの場合は、次のようになります。

```
ModifyGraph opaque=1
```

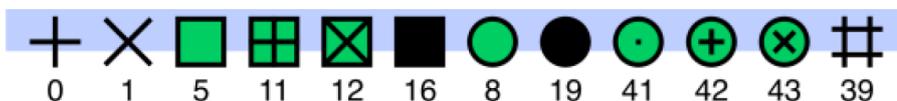


デフォルトでは、不透明にした中が空のマーカーの内側の色は白です。

Modify Trace Appearance ダイアログで塗りつぶしの色を選択するか、ModifyGraph mrkFillRGB コマンドを使って、内部の色を変更することができます。

次の図は、以下のコマンドを使って塗りつぶしの色を緑にしたものです。

```
ModifyGraph mrkFillRGB=(1,52428,26586)
```

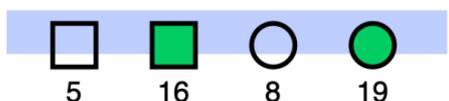


次のように、ソリッドなマーカーのストロークの色または中が空のマーカーの塗りつぶし色を設定することで、同一の結果を得ることができます。

トレースの色を緑に設定すると次のようになります。



次に、ストロークの色を黒に設定すると、黒の輪郭と緑の内部を持つソリッドなマーカーが作成されます。



最後に、塗りつぶし色を緑に設定し、中が空のマーカーが緑で塗りつぶされるようにしました。



これで、ソリッドなマーカーと中が空のマーカーが同じように見えます。

これは過去のバージョンでの面倒な操作です。

Igor Pro 9.0 より前のバージョンでは、塗りつぶしの色を白から変更できませんでした。

Igor Pro 9.0 より前では、11 や 42 などの内部にマーキングが付いた中が空のマーカーに色を塗ることができませんでした。

テキストマーカー

組み込みの描画マーカーに加えて、テキストマーカーとして次のどれかを使うように Igor に指示することもできます。

- フォントの1文字
- テキストウェーブの内容
- 数値ウェーブの内容

フォントに含まれる特殊記号で、Igor のビルトインマーカーに含まれていないものを使いたい場合、フォントの1文字が重要となります。

指定された文字はすべてのデータポイントで使われます。

他のオプションは、XY プロットに3番目の値を表示する方法を提供します。

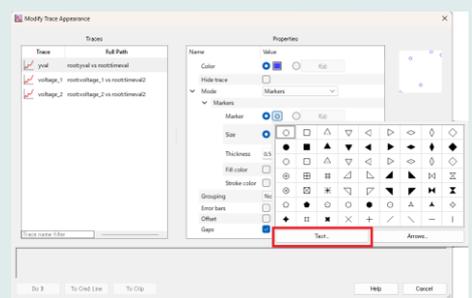
例えば、地震のマグニチュードと日付のプロットでは、テキストウェーブを使ってテキストマーカーを指定することで、地震の発生場所を示すこともできます。

あるいは、地震の発生場所と日付のプロットに数値ウェーブを使ってテキストマーカーを指定し、マグニチュードを表示することもできます。

XY プロットの各データポイントに対応するテキストまたは数値のウェーブのポイントが、マーカーのテキストを供給します。

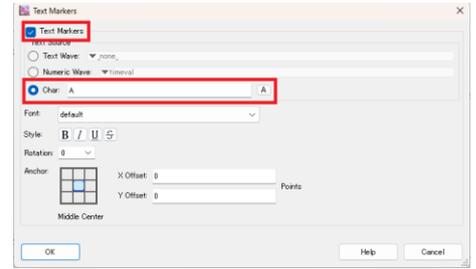
マーカーウェーブは、X ウェーブと Y ウェーブと同じ数のポイントを持つ必要があります。

1. **テキストマーカーを作成するには、Modify Trace Appearance ダイアログの Marker で Marker または Lines and Markers の表示モードを選択します。**
次に、Markers ポップアップメニューをクリックし、Text ボタンをクリックします。

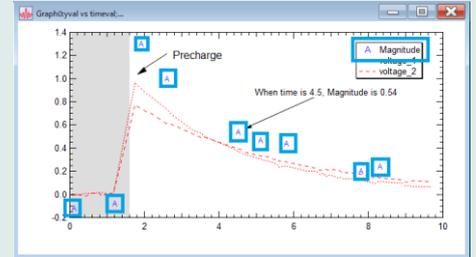


2. Text Markers ダイアログが表示され、テキストのソースを指定できるほか、マーカーのフォント、スタイル、回転、その他のプロパティを指定できます。

例として、Text Markers チェックボックスをオンにし、Char でマーカーに使う文字（ここでは「A」）を指定します。



3. 設定を完了すると、グラフに次のようにテキストマーカーが表示されます。



すべてのテキストマーカーを同じ量だけオフセットおよび回転することはできませんが、個々のデータポイントに対してオフセットおよび回転をすることはできません。

その場合は、タグを使ってください。

文字マーカーをデータポイントの正確な中心に配置するために、X と Y のオフセットを実験的に調整する必要があるかもしれません。

例えば、各データポイントのすぐ上にテキストを中央揃えにするには、Anchor ポップアップメニューから Middle Bottom を選択し、Y オフセットを 5~10 ポイントに設定します。

他の項目とは異なるオフセットが必要な場合は、タグを使う必要があります（マニュアル III-43 Tags を参照してください）。

Igor は、マーカーサイズからテキストマーカーに使うフォントサイズを決定します。これは、Modify Trace Appearance ダイアログで設定したものです。

使われるフォントサイズはマーカーサイズの3倍です。

テキストマーカーと通常の描画マーカーを表示したい場合があります。

これを行うには、グラフに同じウェーブを2回表示する必要があります。

グラフを作成し、描画マーカーを使うようにトレースを設定したら、メニュー Graph → Append Traces to Graph を選択して、2番番目のウェーブのコピーを追加します。

このコピーをテキストマーカーを使うように設定します。

矢印マーカー

矢印マーカーは、例えば、流れや勾配の場を示すベクトルプロットを作るために使うことができます。

矢印マーカーは特殊な用途であり、少し事前準備が必要です。

次は非常に簡単な例です。コメント (//) の入力は自由です。

```
// XY データを作成
Make/O xData = {1, 2, 3}, yData = {1, 2, 3}
Display yData vs xData // グラフを作成
ModifyGraph mode(yData) = 3 // マーカーのモード
```

```
// 矢印データのウェーブを作って、各ポイントの長さや角度をコントロール
Make/O/N=(3,2) arrowData // 矢印の長さや角度をコントロール
Edit /W=(439,47,820,240) arrowData

// arrowData にサンプルのデータを投入
arrowData[0][0]= {20,25,30} // 列 0 : 各点の矢印の長さ
arrowData[0][1]= {0.523599,0.785398,1.0472} // 列 1 : ラジアンで矢印の角度を指定

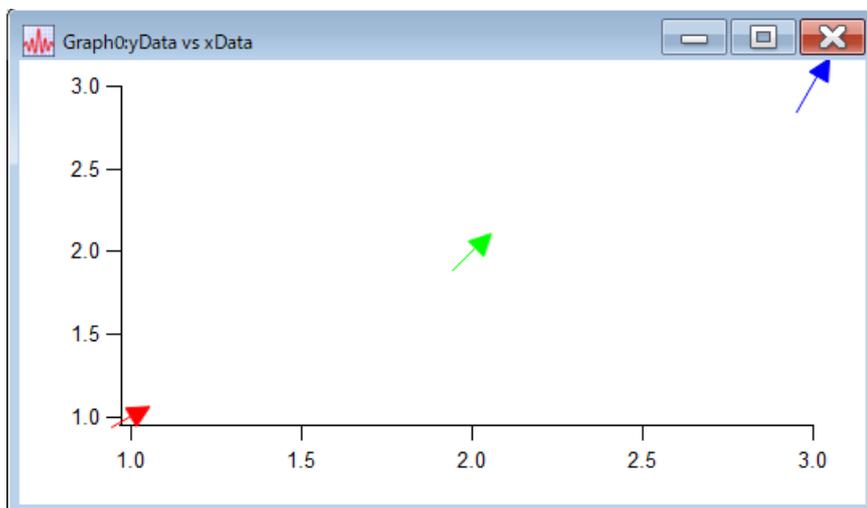
// 矢印を表示するためにトレースを矢印モードに設定
ModifyGraph arrowMarker(yData) = {arrowData, 1, 10, 1, 1}

// RGB カラーのウェーブを作成
Make/O/N=(3,3) arrowColor
Edit /W=(440,272,820,439) arrowColor

// 色のウェーブにいくつかの色を保存
arrowColor[0][0]= {65535,0,0} // Red
arrowColor[0][1]= {0,65535,0} // Green
arrowColor[0][2]= {0,0,65535} // Blue

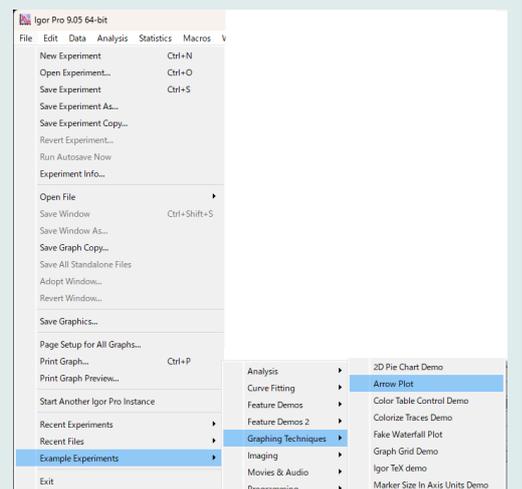
// f(z) モードとして色をオンにする
ModifyGraph zColor(yData)={arrowColor,*,*,directRGB,0}
```

上のプログラムをコマンドウィンドウで実行すると、次のようなグラフになります。



矢印マーカーのデモンストレーションは、メニュー **File** → **Example Experiments** → **Graphing Techniques** → **Arrow Plot** にあります。

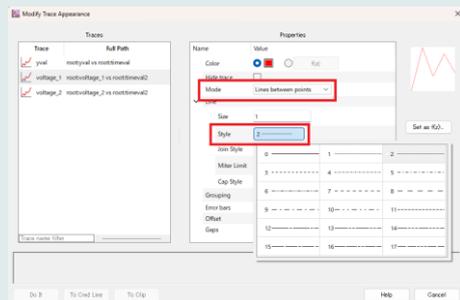
詳細は、マニュアル V-613 ModifyGraph (traces) の arrowMarker キーワードの説明を参照してください。



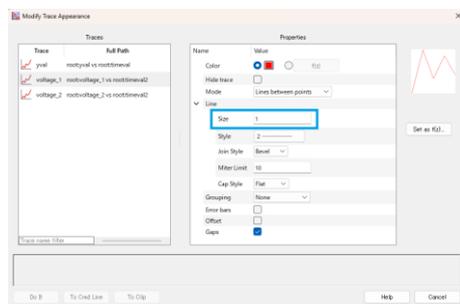
線のスタイルと幅

Modify Trace Appearance ダイアログで、Lines between points、Lines and markers、Cityscape モードを選択した場合、線のスタイルも選択できます。

破線のパターンは、Line セクションの Style を使って変更できます。



Markers モード以外のモードでは、線の太さを設定できます。線のサイズはポイントで表され、小数点以下も使うことができます。線幅がゼロの場合、線は消えます。



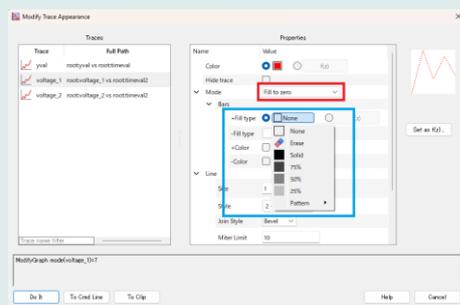
詳細は、マニュアル III-496 Dashed Lines を参照してください。

グラフの塗りつぶし

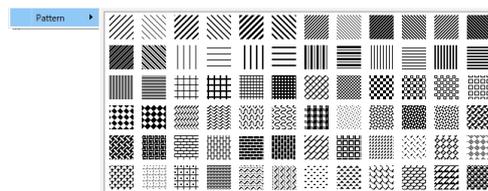
棒グラフと Fill to zero モードのトレースについては、Igor は塗りつぶしの形式を設定できるようになっています。

Modify Trace Appearance ダイアログで、Fill to zero モードを選択します。

塗りつぶしの種類は、塗りつぶしが透明であることを示す None、塗りつぶしが白で不透明であることを示す Erase、ソリッドな色を示す Solid、そして3つのグレイがあります。



パレットからパターンを選択することもでき、正方向領域 (+Color) と負方向領域 (-Color) の塗りつぶしのタイプと色を個別に選択することもできます。



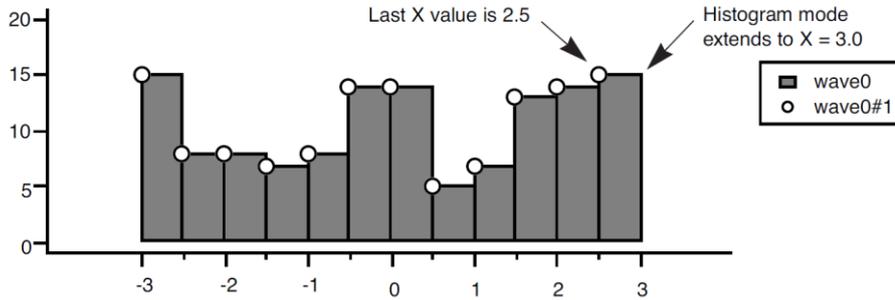
詳細は、マニュアル III-498 Fill Patterns と III-498 Gradient Fills を参照してください。

ウェーブフォームを持つ棒グラフ

(カテゴリ軸ではなく) 通常の連続 X 軸にプロットされたウェーブフォームに Bars モードを使う場合、X 値はウェーブの X スケーリングから計算されます。

バーは、指定された点の X 値から次の点の X 値まで (ただし、次の点の X 値は含まない) の範囲で描画されます。

このような棒グラフは、通常、2つの X 値間のヒストグラムのカウント数を表すために使われるため、Histogram Bars と呼ばれています。



棒グラフの中心を X 値に配置したい場合は、カテゴリプロットを作成すべきです。

カテゴリプロットは従来の棒グラフに適しています (マニュアル II-14 Category Plots を参照してください)。

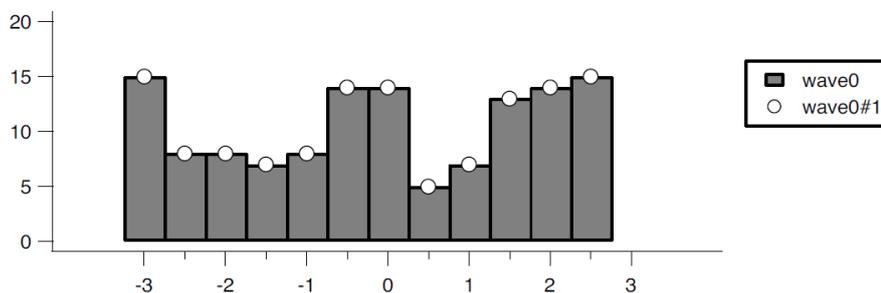
しかし、従来の棒グラフのように、フラットな部分が元の X 値を中心に表示されるように、ウェーブの X 値を調整することができます。

データを実際に修正することなくこれを行う方法の1つは、グラフのトレースをバー幅の半分だけオフセットすることです。

ドラッグするだけで移動させることもできますが、Modify Trace Appearance ダイアログを使って、より正確なオフセットコマンドを生成することもできます。

この例では、バーの幅は X 単位の 0.5 です。コマンドは次のように 0.25 だけずらします。

```
ModifyGraph offset (wave0)={-0.25,0}
```



XY ペアを持つ棒グラフ

X 軸が XY ペアでコントロールされている場合は、各バーの幅は2つの X 値によって決まります。

X 値の1つはバーの左端の位置を示し、次の X 値はバーの右端の位置を示します。

棒グラフを1本だけ想像すると、これは2つの点を持つ XY ペアを必要とします。

棒グラフの左端を設定する1つの X 値と、右端を設定する次の X 値です。

一般化すると、N 本の棒グラフでは、X と Y のウェーブに N+1 個の点が必要です。

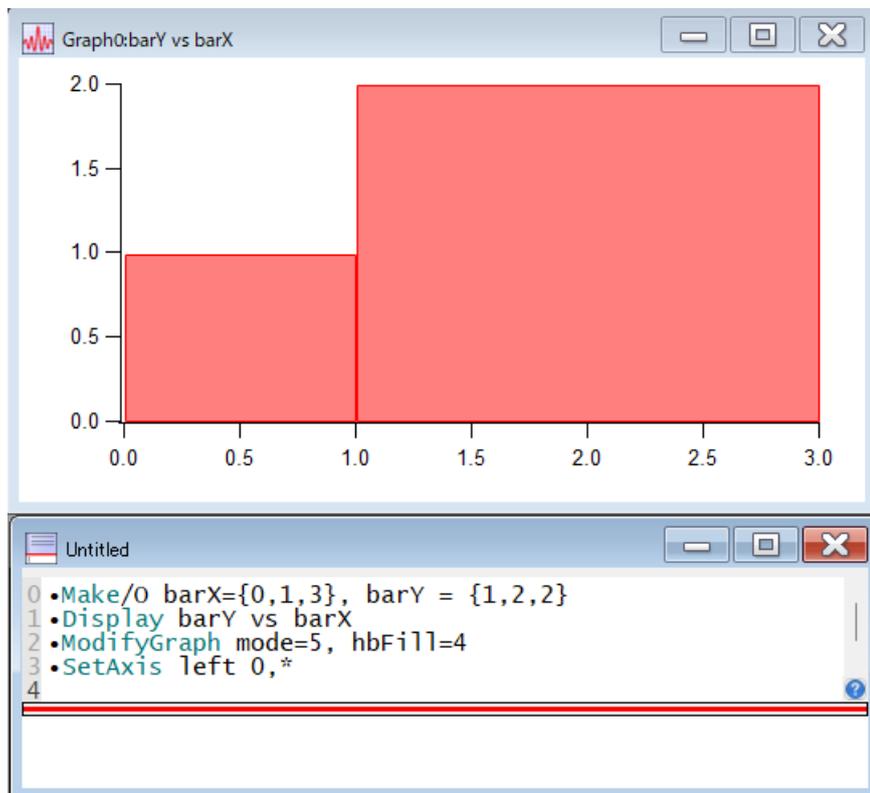
3つの XY 点を使って2つのバーを生成する例を次に示します。

```

Make/O barX={0,1,3}, barY = {1,2,2}
Display barY vs barX
ModifyGraph mode=5, hbFill=4
SetAxis left 0,*

```

コマンドを実行すると、次のグラフが得られます。



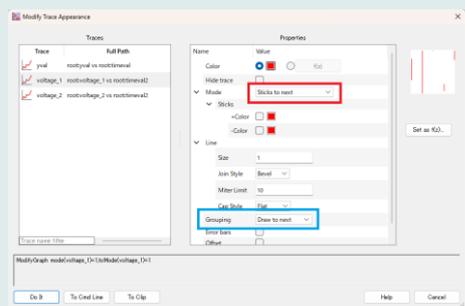
グループ化、積み上げ、モードの追加

通常、 $y=0$ に描画する4つのモード ("Sticks to zero", "Bars", "Fill to zero", "Sticks and markers") については、次のトレースの Y 値に描画するバリエーションを選択できます。

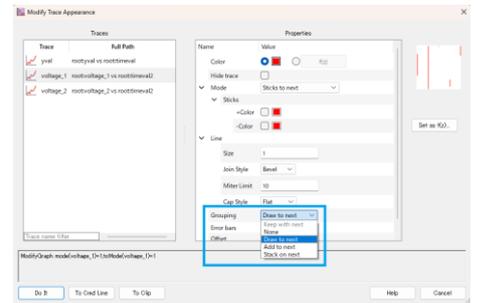
4つのバリエーションモードは、Stick to next, Bars to next, Fill to next, Sticks&markers to next です。

ここでいう Next とは、Modify Trace Appearance ダイアログおよび Reorder Traces ダイアログのトレースのリストで選択したトレースの下（後ろ）にリスト表示されるトレースを指します。

この4つのモードのいずれかを選択すると、Igor は Grouping ポップアップメニューから Draw to next を自動的に選択します。Add to next と Stack on next モードも選択できます。



Modify Trace Appearance ダイアログの Grouping ポップアップメニューは、レイヤーグラフや積み上げ棒グラフなどの特殊効果を作成するために使います。使うことができるモードは、Keep with next, None, Draw to next, Add to next, Stack on next です。



Keep with next は、カテゴリプロットでのみ使われ、詳細はマニュアル II-14 Category Plots を参照してください。

Draw to next は、通常は $Y=0$ に描画するモードの動作を変更し、現在のトレースと同じ軸のペアに対して描画される次のトレースの Y 値に描画するようにします。

次のトレースの X 値は、現在のトレースの X 値と同じでなければなりません。

同じでない場合、次のトレースは現在のトレースの下軸と一致しません。

Add to next は、プロットする前に、現在のトレースの Y 値を次のトレースの Y 値に追加します。

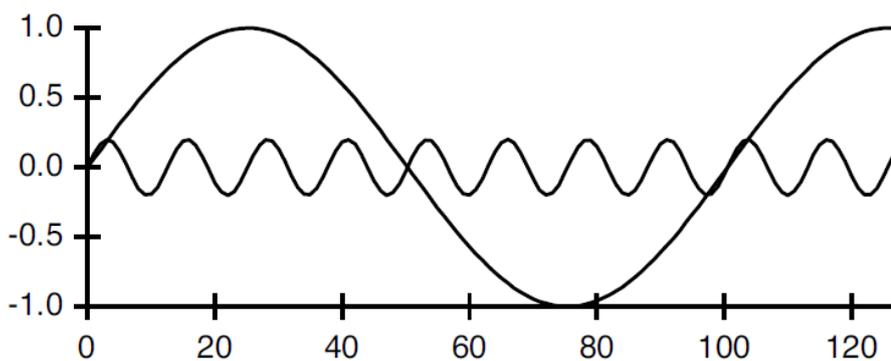
次のトレースも Add to next を使っている場合は、その追加が最初に実行され、以下同様になります。

通常、 $Y=0$ に描画する4つのモードのいずれかと併用すると、このモードも Draw to next のように動作します。

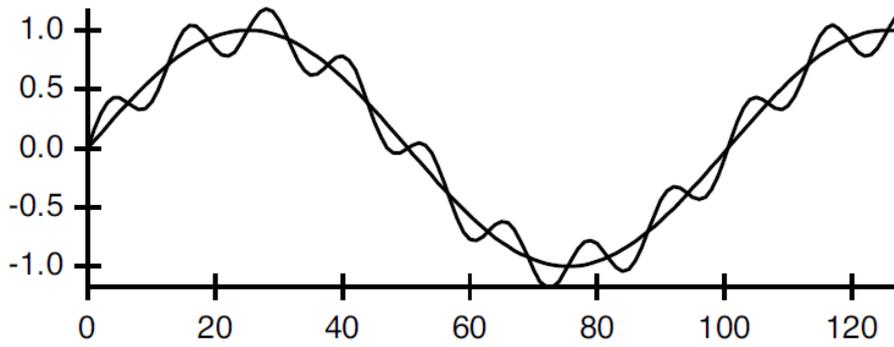
Stack on next は、Add to next と同じように動作しますが、 Y 値のバックトラックは許可されていません。

つまり、次のトレースの Y 値が正の場合、負の値はゼロのように作用し、次のトレースが負の場合、正の値はゼロのように作用します。

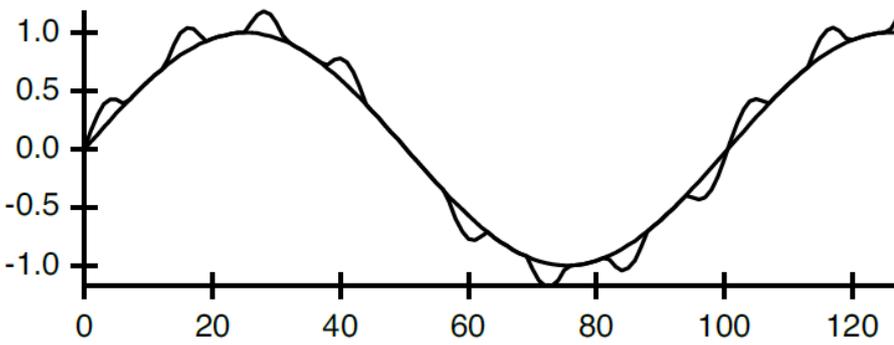
これは小さな正弦波と大きな正弦波の通常のプロットです。



このバージョンでは、小さな正弦波は Add to next モードに設定されています。



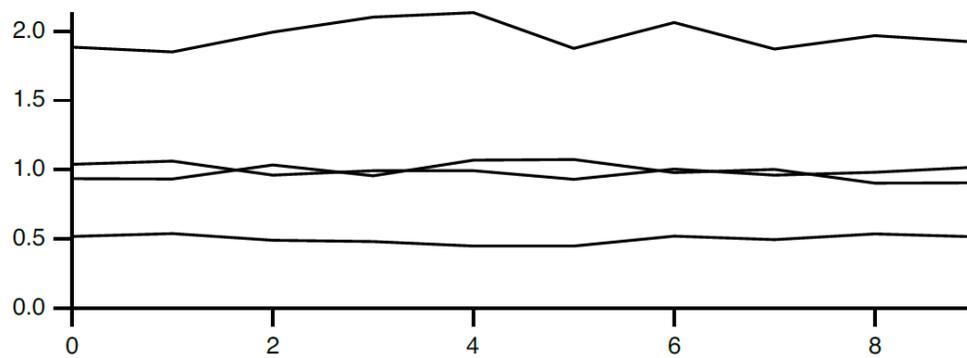
これは、Stack on next モードです。



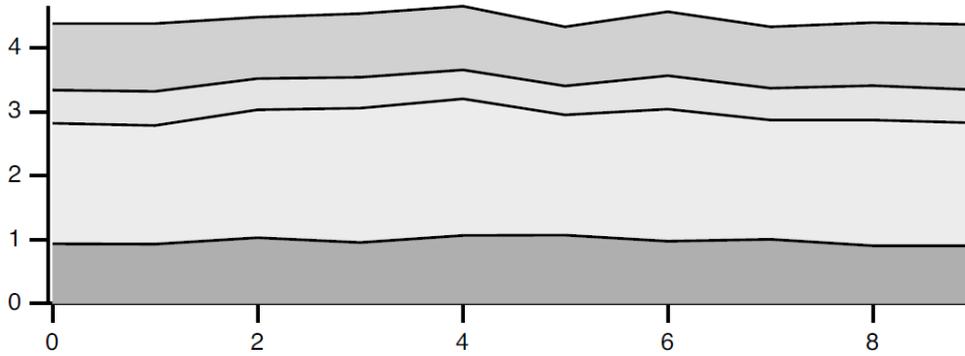
塗りつぶしを次のモードに使うと、グラフに複数のウェーブをプロットすることで、レイヤーグラフを作成できます。

データによっては、次のグループ化したモードに追加したいかもしれません。

例えば、次の通常のグラフでは、各トレースは地層の厚さを示しているかもしれません。



Fill to next モードと Add to next モードを使うことで、レイヤーをより分かりやすく表示することができます。



グループ化モードは、次のトレースのアイデンティティに依存するため、グラフ内のトレースの順序を調整する必要がある場合があります。

Reorder Traces ダイアログを使って、これを実行できます。メニュー Graph → Reorder Traces を選択してください。

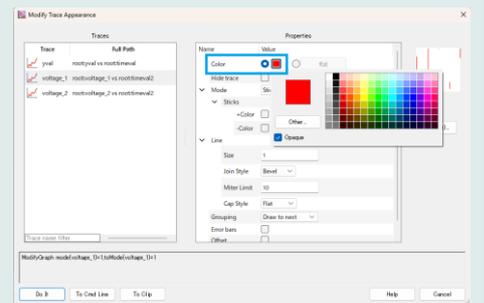
移動したいトレースを選択します。

選択したトレースをリスト内で上下にドラッグして適切な位置でドロップすることで、順序を調整します。

注記：さまざまなグループ化、追加、積み重ねモードで使うすべてのウェーブは、同じ数の点と X 軸のスケールを持ち、すべて同じ軸を使って表示される必要があります。

トレースの色

Modify Trace Appearance の色の選択のポップアップパレットから、選択したトレースの色を選択できます。

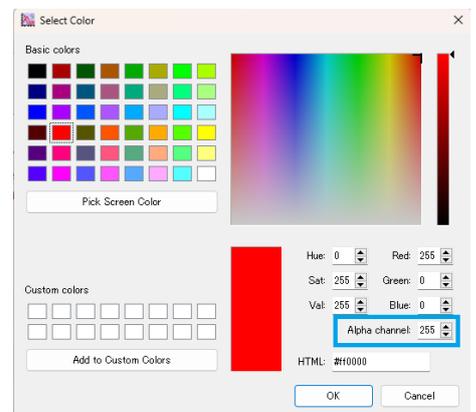


色に加えて、alpha プロパティを使って不透明度を指定することができます。

ポップアップパレットで Other ボタンをクリックします。

アルファ値が 255 の場合、トレースは完全に不透明になります。

アルファ値が 0 の場合、完全に透明になります。



補助 (z) ウェーブからトレースのプロパティを設定

補助ウェーブの値に基づいて、トレースの色、マーカー番号、マーカーサイズ、パターン番号をポイントごとに設定できます。

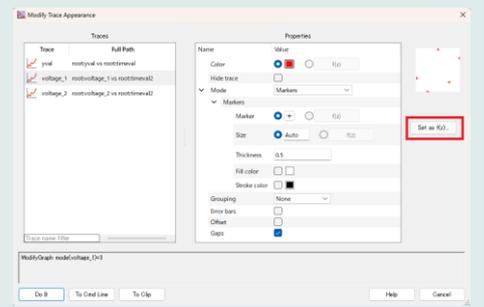
補助ウェーブは「Z ウェーブ」と呼ばれます。

他のウェーブがトレース上の各ポイントの X と Y の位置をコントロールするのに対し、Z ウェーブは3番目の特性をコントロールするからです。

例えば、地震が発生した場所にマーカーを配置し、それぞれの地震の規模を示すためにマーカーの大きさを変えることができます。

地震の深さをマーカーの色で表示し、異なる種類の地震を異なるマーカーの形状で表示することもできます。

トレースプロパティを Z ウェーブの関数として設定するには、Modify Trace Appearance ダイアログで Set as f(z) ボタンをクリックして、Set as f(z) ダイアログを表示します。

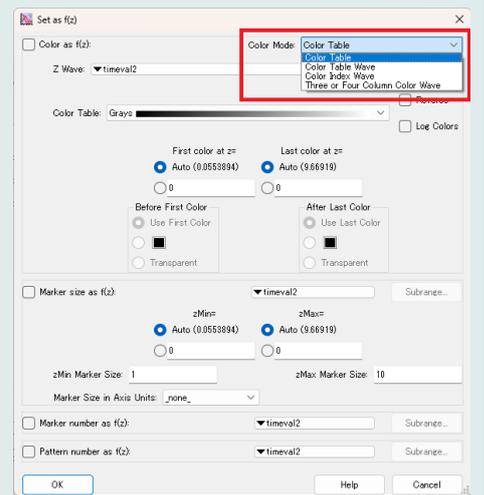


Set as f(z) ダイアログの Color as f(z) 設定

1. Modify Trace Appearance ダイアログで Set as f(z) ボタンを押すと、Set as f(z) ダイアログが表示されます。

Color as f(z) は、Color Table, Color Table Wave, Color Index Wave, Three or Four Column Color Wave の4つのモードがあります。

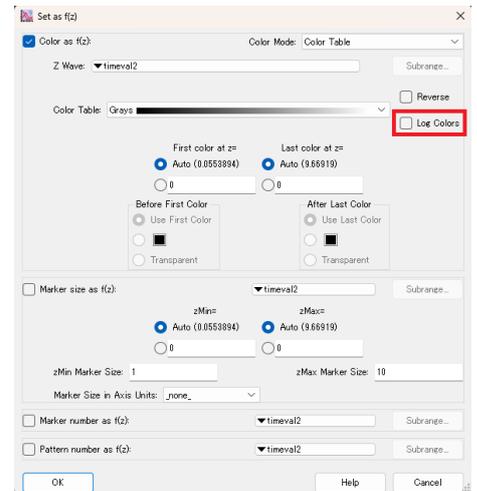
モードは Color Mode ポップアップメニューから選択します。



2. Color Table モードでは、トレース上の各データポイントの色は、対応する Z ウェーブの値をビルトインカラーテーブルにマッピングすることで決定されます。

Log Colors チェックボックスがチェックされている場合は対数、チェックされていない場合は線形となります。

Log Colors オプションは、zWave が多くの桁に広がり、より小さな値のより詳細な変化を表示したい場合に便利です。

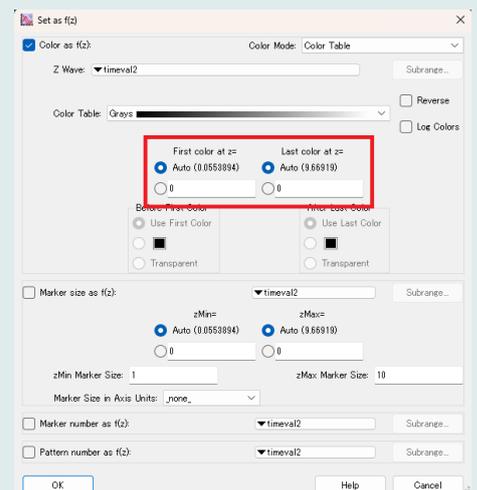


3. First color at z= と Last color at z= の設定は、Z ウェーブの値の範囲を定義し、カラーテーブルにマッピングします。範囲外の値は、範囲の最後の色になります。

First color at z= または Last color at z= に Auto を選択すると、Igor は Z ウェーブで検出された最小値または最大値を使います。

Z 値のいずれかが NaN の場合、Igor は X または Y データが NaN の場合と同じ方法でそれらのデータ点を処理します。

これは Modify Trace Appearance ダイアログの Gaps 設定に依存します。



Color Table Wave モードは、Color Table モードと同じですが、ビルトインのカラーテーブルではなく、ユーザーが指定したカラーテーブルのウェーブによって色が決まる点が異なります。詳細はマニュアル II-399 Color Table Waves を参照してください。

Color Index Wave モードでは、トレース上のデータポイントの色は、選択した Z ウェーブの値を、選択した 3 列カラーインデックスウェーブの X スケーリングにマッピングすることで決定されます。

これは、ModifyImage cindex が (Z ウェーブ値の代わりに) 3 列のカラーインデックスマトリックス内の色に画像の値をマッピングする方法と似ています。

詳細はマニュアル II-400 Indexed Color Details を参照してください。

Three or Four Column Color Wave モードでは、選択したウェーブの最初の 3 列の赤、緑、青の値に従ってデータポイントが色付けされます。

3 列のウェーブの各行は、トレース上のデータポイントの色を示します。

このモードでは、トレース上の各データ点の色を完全にコントロールすることができます。

ウェーブに 4 番目の列がある場合、それは不透明度をコントロールします。

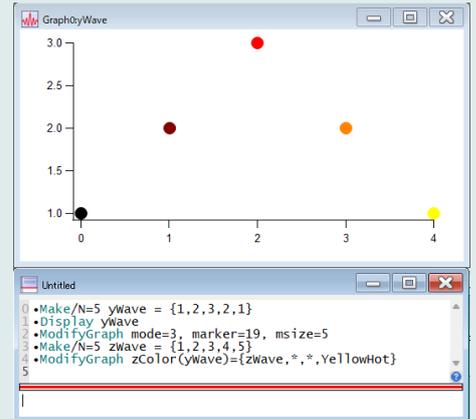
詳細はマニュアル II-401 Direct Color Details を参照してください。

Color as f(z) の例

1. 新しい Experiment を作成し、コマンドウィンドウで次を実行します。

```
Make/N=5 yWave = {1,2,3,2,1}
Display yWave
ModifyGraph mode=3, marker=19, msize=5
Make/N=5 zWave = {1,2,3,4,5}
ModifyGraph zColor(yWave)={zWave,*,*,YellowHot}
```

ポイントごとに色が違うグラフが生成されます。



2. 直接色を指定するための3列のウェーブを作成し、それに RGB 値を手動で入力するようにしてみます。

コマンドウィンドウで次を実行します。

```
Make/N=(5,3) directColorWave
```

The figure shows a window titled 'TableOdirectColorWave' with a table containing 6 rows and 4 columns. The columns are labeled 'directColorWave[0]', 'directColorWave[1]', 'directColorWave[2]', and 'directColorWave[3]'. The rows are numbered 0 to 5.

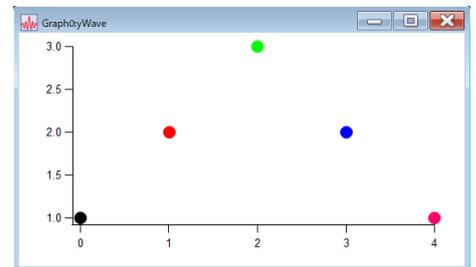
Row	directColorWave[0]	directColorWave[1]	directColorWave[2]	directColorWave[3]
0	0	0	0	0
1	65535	0	0	0
2	0	65535	0	0
3	0	0	0	65535
4	65535	0	26214	0
5				

3. このウェーブを使ってマーカーの色を直接コントロールすることができます。

コマンドウィンドウで次を実行します。

```
ModifyGraph
zColor(yWave)={directColorWave,*,*,directRGB}
```

グラフのドットの色が変わります。



Set as f(z) ダイアログの Marker Size as f(z) 設定

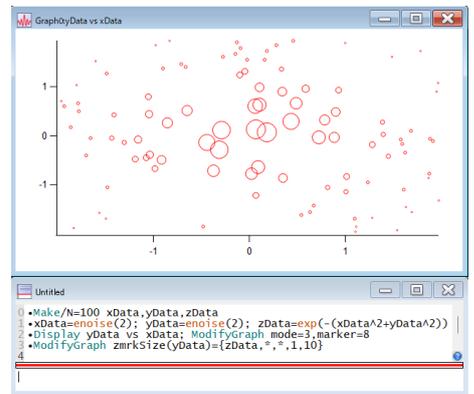
Modify Trace Appearance ダイアログで Set as f(z) ボタンを押すと、Set as f(z) ダイアログが表示されます。Marker size as f(z) は Color Table モードの Color as f(z) と同じように機能しますが、Z 値は最小 (zMin Marker Size) および最大 (zMax Marker Size) マーカ設定を使って定義したマーカーサイズの範囲にマッピングされます。

この例では、マーカーのサイズの機能として3番目の値を使います。

新しい Experiment を作成し、コマンドウィンドウで次を実行します。

```
Make/N=100 xData,yData,zData
xData=enoise(2); yData=enoise(2); zData=exp(-(
(xData^2+yData^2))
Display yData vs xData; ModifyGraph mode=3,marker=8
ModifyGraph zmrkSize(yData)={zData,*,*,1,10}
```

3番目の値(zData) 値に応じてポイントの大きさが決まるグラフが生成されます。



Set as f(z) ダイアログの Marker Size as f(z) in Axis Units 設定

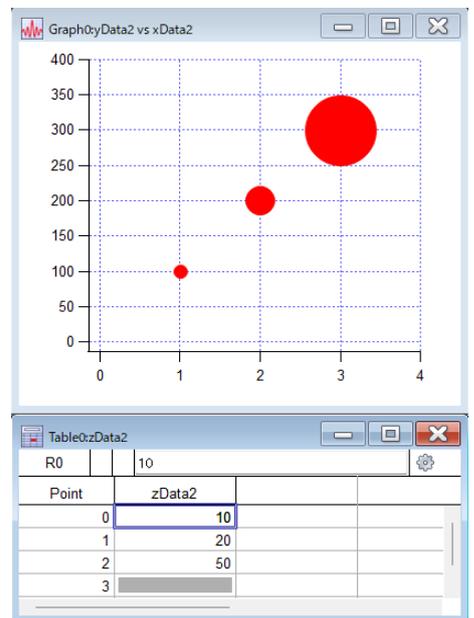
Modify Trace Appearance ダイアログで Set as f(z) ボタンを押すと、Set as f(z) ダイアログが表示されます。Igor Pro 9.0 以降では、軸の単位でマーカーの大きさを指定することができます。

新しい Experiment を作成し、コマンドウィンドウで次を実行します。

```
Make/O xData2 = {1,2,3}
Make/O yData2 = {100,200,300}
Make/O zData2 = {10, 20, 50}
Display/W=(400,50,700,300) yData2 vs xData2
// マーカーのモード。塗りつぶされた円のマーカー
ModifyGraph mode=3, marker=19, msize=6
ModifyGraph grid=1
ModifyGraph nticks(left)=10
SetAxis left,0,400
SetAxis bottom,0,4
ModifyGraph zmrkSize(yData2)={zData2,*,*,6,10,left}
```

最後のコマンドは、マーカーのサイズを、左軸にスケーリングされた値を持つウェーブ zData2 によってコントロールするように設定しています。

例えば、zData2[2] は 50 であり、yData2 のポイント 2 のマーカーの半径を左軸の 50 単位に設定します。



グラフ上のどの軸でも指定することができます。

その軸が後で削除された場合、マーカーのサイズは軸を指定しなかったかのように no axis の状態に戻ります。

マーカーのサイズを軸の単位で拡大縮小する場合、zmrkSize キーワードの zMin, zMax, mrkMin, mrkMax パラメーターに指定した値は、指定した軸を削除しない限り、マーカーサイズには影響しません。

その場合、これらのパラメーターはトレースのマーカーサイズをコントロールします。

マーカーのサイズスケールが、トレースがプロットされている軸のスケールとは独立している場合、スケールを示すために自由軸を使うことができます。

希望する範囲で自由軸を作り、それを `zmrkSize` キーワードの軸として設定することで、これを行うことができます。

自由軸の説明は、マニュアル II-279 Types of Axes を参照してください。

指定した軸が対数軸の場合、Z 値の解釈が変更されます。

対数軸と同じ長さの概念的な軸を（単位）ポイントで考えてみます。

概念的な軸の最小値と最大値を、対数軸の最小値の対数と最大値の対数に設定します。

与えられた Z 値に対応するマーカーサイズは、概念的な軸の Z 単位の長さになります。

これを視覚化するために、この概念的な軸を表す自由軸を作ることができます。

これに関するデモンストレーションは、次の場所にあります。

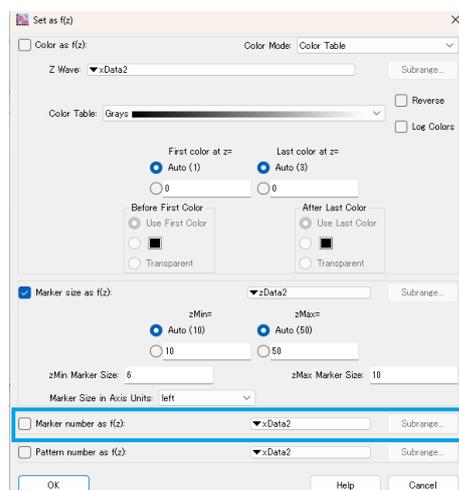
メニュー File → Example Experiments → Graphing Techniques → Marker Size in Axis Units

Set as f(z) ダイアログの Marker Number as f(z) 設定

Modify Trace Appearance ダイアログで Set as f(z) ボタンを押すと、Set as f(z) ダイアログが表示されます。

Marker Number as f(z) モードでは、各データポイントの実際のマーカー番号を含む Z ウェーブを作る必要があります。

詳細は、マニュアル II-291 Markers を参照してください。

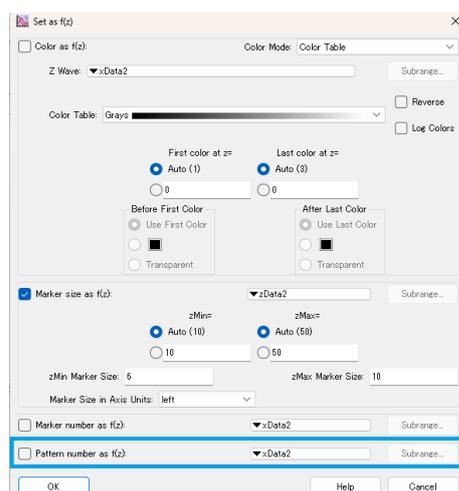


Set as f(z) ダイアログの Pattern Number as f(z) 設定

Modify Trace Appearance ダイアログで Set as f(z) ボタンを押すと、Set as f(z) ダイアログが表示されます。

Pattern Number as f(z) モードでは、各データポイントの実際のパターン番号を含む Z ウェーブを作る必要があります。

パターン番号の一覧については、マニュアル III-498 Fill Patterns を参照してください。



Color as f(z) 凡例の例

色を f(z) モードとして使うグラフがあるとき、色が何を意味するのかを示す凡例を作成したいと思うかもしれません。

このセクションでは、これを行うための Legend コマンドの機能を使う方法を説明します。

新しい Experiment を作成して、以下のコマンドを1つずつ実行してください。

1つずつ実行することで各コマンドの役割がよくわかります。

```
// テストデータの生成
Make /O testData = {1, 2, 3}

// マーカーモードでグラフを表示
Display testData
ModifyGraph mode=3,marker=8,msize=5

// トレースからシンボルが生成される通常の凡例を生成
Legend/C/N=legend0/J/A=LT "%s(testData) First\r%s(testData) Second\r%s(testData) Third"

// マーカーの色をコントロールするカラーインデックスウェーブを作成
Make /O testColorIndex = {0, 127, 225}

// Color as f(z) モードを使うためにグラフトレースを変更。
// Rainbow256 はビルトインのカラーテーブルの名前。
// 0 と 255 という数値は、カラーテーブルの最初のエン트리と最後のエントリに対応する
// カラーインデックスの値を設定する。
ModifyGraph zColor(testData)={testColorIndex,0,255,Rainbow256,0}

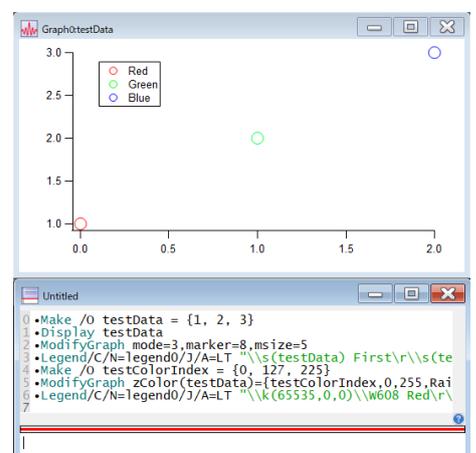
// 色を表示するように凡例を変更
Legend/C/N=legend0/J/A=LT "%k(65535,0,0)%W608 Red\r%k(0,65535,0)%W608
Green\r%k(0,0,65535)%W608 Blue"
```

結果は次のようなグラフになります。

最後のコマンドでは、凡例で使うマーカー（この場合は円形のマーカー 08）とマーカーの太さ（6 は 1.0 ポイントを指す）を指定するために、エスケープシーケンス「%W」を使用しました。

エスケープシーケンス「%k」は、エスケープシーケンス「%W」で指定されたマーカーを描くときに使う色を指定します。

マーカーの塗りつぶし色を指定するには「%K」を使います。色は RGB 形式で指定し、各要素は 0 から 65535 の範囲の値となります。



この例では、2つのバックスラッシュを使っています。これは、Igor のリテラル文字列では、1つのバックスラッシュがエスケープ文字として扱われるためです。

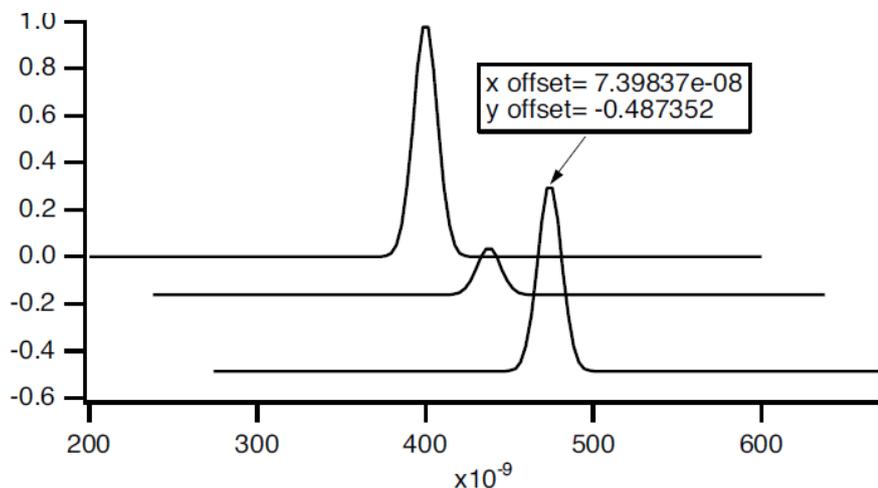
最終的なテキストにバックスラッシュを残す必要があるため、Igor が「k」と「W」で必要とするバックスラッシュを残すには、リテラル文字列で2つのバックスラッシュを使う必要があります。

Add Annotation ダイアログで凡例のテキストを入力する場合、バックスラッシュは1つだけ使い、ダイアログは2つのバックスラッシュを含む requires コマンドを生成します。

トレースのオフセット

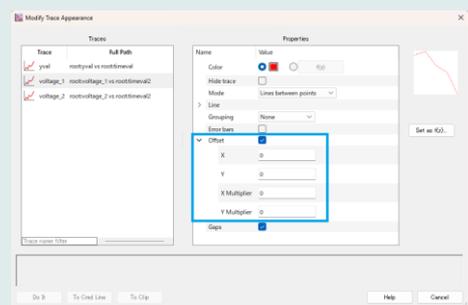
グラフのトレースを、関連するウェーブのデータを変更することなく、水平方向または垂直方向にオフセットすることができます。

これは主に、トレースの形状を比較したり、重なり合ったトレースを広げて見やすくしたりするのに役立ちます。



各トレースには X と Y のオフセットがあり、両方とも初期値はゼロです。

Modify Trace Appearance ダイアログでオフセットのチェックボックスをオンにすると、トレースの X および Y オフセットを入力することができます。

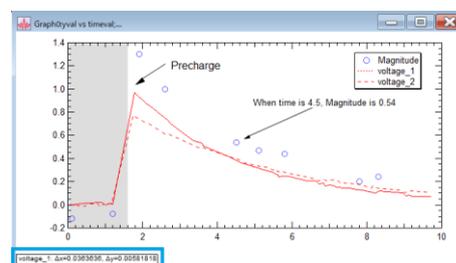


また、グラフ上でクリック&ドラッグしてオフセットを設定することもできます。

これを行うには、**オフセットしたいトレースをクリックし、マウスを約1秒間押し続けます。**

グラフの左下に読み取りボックスが表示されます。

トレースをドラッグすると、読み取りボックスに X と Y のオフセットが表示されます。



指定したトレースを表示するのにそれほど時間がかからない場合、画面上でトレースをドラッグすると、そのトレースを表示することができます。

ウェーブのオフセット中に Shift キーを押すと、Igor はオフセットを水平または垂直の大きさに制限します。

Caps Lock キーを押すことで、トレースのドラッグを無効にすることができます。これはトラックボールユーザーにとって便利な機能です。

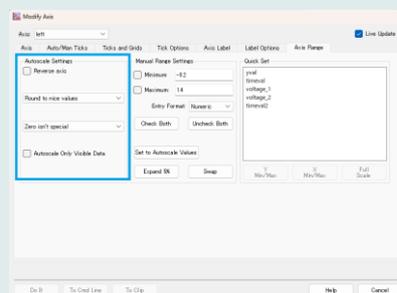
オフセットは元に戻すことができないため、誤って不要なトレースをドラッグしてしまった場合は、メニュー Edit → Undo を選択します。

トレースにタグを付けて、現在のオフセット値を表示させることができます。

詳細は、マニュアル III-38 Dynamic Escape Codes for Tags を参照してください。

グラフに自動スケーリングが有効になっている場合、Igor はトレースオフセットを考慮しようとしません。

軸のスケーリングに影響を与えずにトレースのオフセットを設定したい場合、Graphs メニューの Set Axis Range を使って、自動スケーリングを無効にします。



対数軸を使うトレースをオフセットする場合、軸が対数ではない場合と同じ距離だけトレースがオフセットされます。

トレースの形は変わりません。単に移動します。

もし、ウェーブのデータに定数を追加してトレースをオフセットしようとする、トレースが歪んでしまいます。

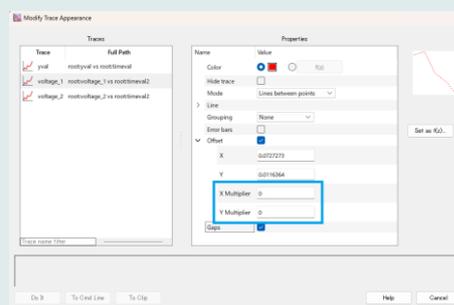
トレースの乗算

トレースをオフセットするだけでなく、トレースを拡大する倍率を指定することもできます。

プロットに使われる有効な値は、**乗数*データ+オフセット**です。

Modify Trace Appearance ダイアログの **Offset** で設定します。

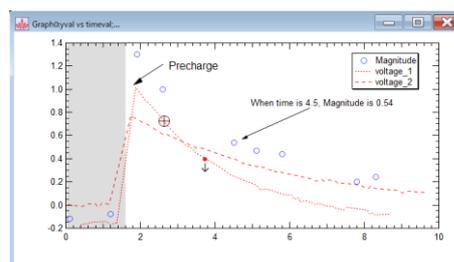
デフォルト値のゼロは、乗数が指定されていないことを意味し、データがゼロで乗算されるべきではないことを意味するものではありません。



(対数ではない) 通常の軸を使っている場合、トレースのオフセットについて説明されているのと同じクリック&ホールドの方法で、トレースをインタラクティブに拡大縮小することができます。

まず、トレースのどこかにカーソル A を配置し (メニュー Graph → Show Info からカーソル A を配置)、参照ポイントとします。

次に、クリックしたままにしてオフセットモードに入ったら、Alt キーを押して、オフセットではなく乗数を調整します。



キーを押したり、離したりすることで、スケーリングとオフセットを切り替えることができます。

対数軸では、トレースの乗算は対数軸上のトレースをオフセットする大体的方法を提供します（次を思い出してください： $\log(a*b)=\log(a)+\log(b)$ ）。

互換性の理由と、トレースのオフセット方法が対数軸と線形軸のモード間の切り替えをより適切に処理するため、Igor では、トレースをドラッグするときに、オフセットがゼロでない（デフォルトでは not set）場合にのみ、乗法を使います。

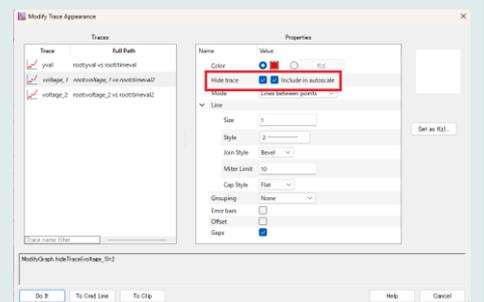
したがって、乗法を使うには、コマンドラインを使うか、Modify Trace Appearance ダイアログの Offset コントロールを使って、ゼロ以外の乗数を設定する必要があります。

この目的のためには、1 が適当です。

トレースを隠す

グラフのトレースを非表示にするには、Modify Trace Appearance ダイアログの Hide Trace チェックボックスをチェックします。

トレースを非表示にした場合、自動スケーリング時に非表示のトレースのデータを使うかどうかをコントロールするには、Include in Autoscale チェックボックスを使います。



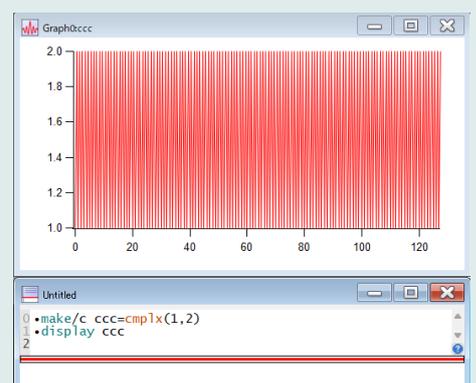
複素数表示モード

複素データのトレースを表示するときには、Modify Trace Appearance ダイアログの Complex Mode ポップアップメニューを使って、データの表示方法をコントロールすることができます。

複素数と実数の成分を一緒に、または個別に表示でき、また、振幅または位相を表示することもできます。

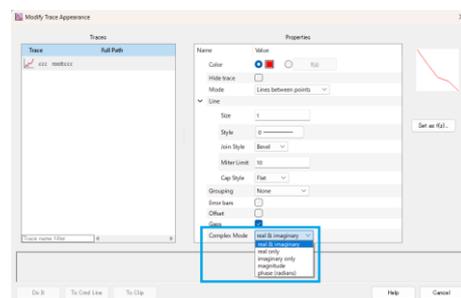
このポップアップメニューは複素数をプロットしているときにのみ表示されます。

例えば、コマンドウィンドウで次を実行して複素数のウェーブを作成して、グラフ表示します。



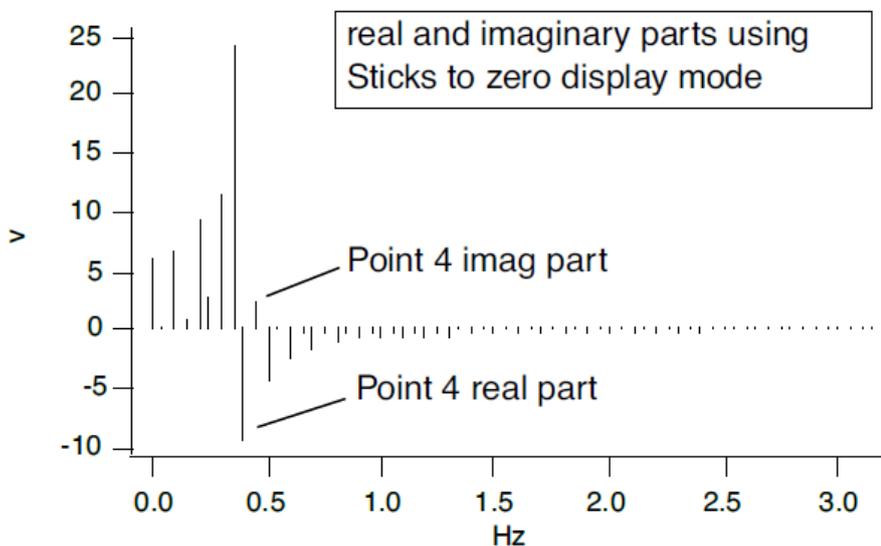
```
make/c ccc=cplx(1,2)
display ccc
```

トレースをダブルクリックして **Modify Trace Appearance** ダイアログを表示すると **Complex Mode** の項目があることがわかります。



デフォルトの表示モードは、**Lines between points** です。

ウェーブの実数部と虚数部を点ごとに並べて表示するには **Sticks to zero** モードを使います。



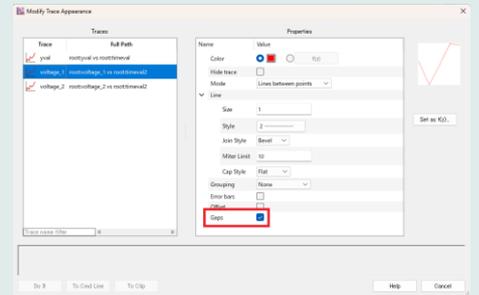
グラフのギャップ

Igor では、欠落または未定義の値は浮動小数点値 NaN (Not a number) として格納されます。

通常、Igor はグラフ上で NaN をギャップとして表示し、そのポイントにはデータがないことを示します。

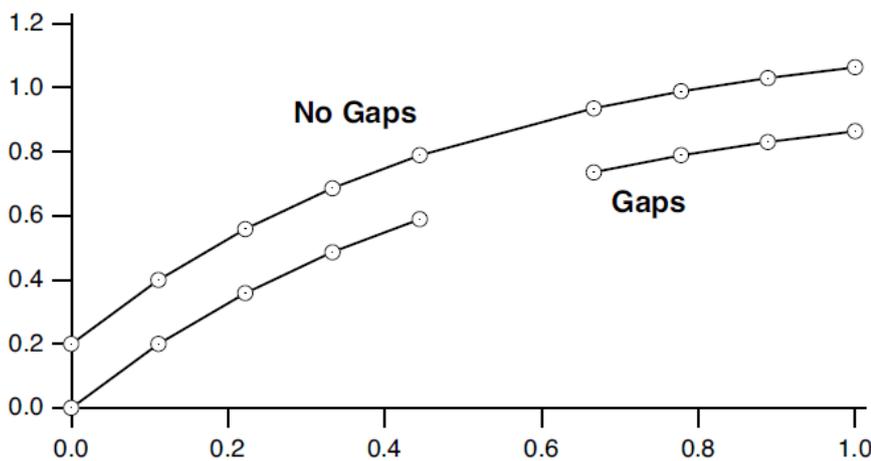
状況によっては、欠損値の前後の値を接続して欠損値を処理することが望ましい場合があります。

Modify Trace Appearance ダイアログの Gaps チェックボックスを使って、これをコントロールできます。



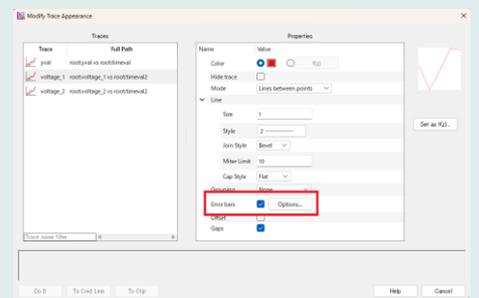
このチェックボックスがチェックされている場合（デフォルト）、Igor は欠損値をデータのギャップとして表示します。

チェックボックスのチェックを外すと、Igor は欠損値を無視し、欠損値の前後にある利用可能なデータ点を接続します。



エラーバー

1. Modify Trace Appearance ダイアログの Error Bars チェックボックスは、選択したトレースにエラーバーを追加します。
このチェックボックスを選択するか、Options ボタンをクリックすると、Igor は Error Bars ダイアログを表示します。



エラーバーは、グラフのトレースに追加できるスタイルです。

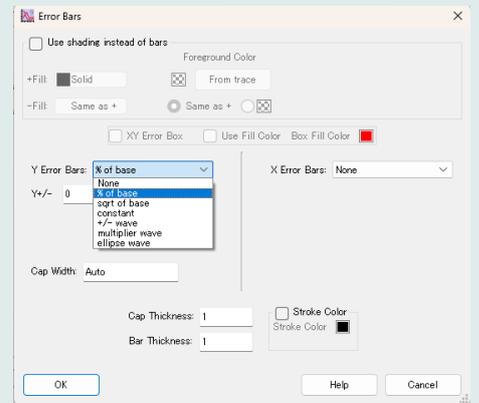
エラー値は、

- 一定の数値、指定したポイントにおけるウェーブの値の一定の割合
- 指定したポイントにおけるウェーブの値の平方根
- または他のウェーブから取得した任意の値

とすることができます。

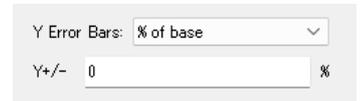
この最後のケースでは、エラー値は上下左右方向に対して個別に設定できます。

2. Y Error Bars と X Error bars のポップアップメニューから、希望するモードを選択してください。



3. 選択したモードによってダイアログが変化します。

% of base モードでは、ベースのウェーブのパーセンテージを入力します。

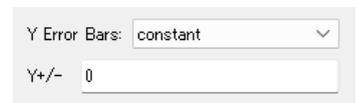


sqrt of base モードでは、これ以上の値を入力する必要はありません。

このモードは、データがカウント値の場合にのみ意味があります。



constant モードでは、X または Y 方向の定数のエラー値を入力します。



4. +/- wave を選択すると、上下または左右のエラー値を指定するウェーブを選択するポップアップメニューが表示されます。

これらのウェーブはエラーウェーブと呼ばれます。

エラーバーの値は、通常、すべて正の値になります。

これは、各点からエラーバーまでの線の長さを指定しているためです。

これは片側エラーバーのみをサポートする唯一のモードです。

エラーウェーブは、ベースウェーブと同じ数値形式と長さである必要はありません。

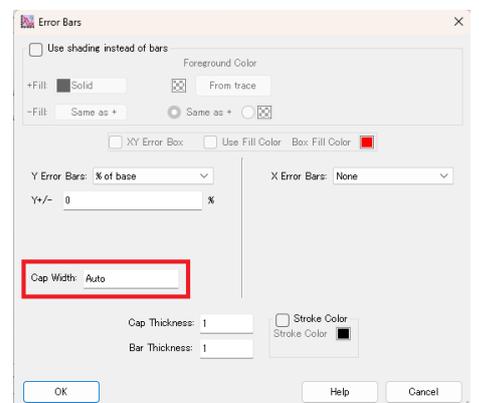
エラーウェーブの点の値が NaN の場合、その点に対応するエラーバーは描画されません。

5. Cap Width の設定では、エラーバーの端のキャップの幅をポイントの整数として指定します。

キャップ幅を Auto (またはゼロ) に設定することもできます。

その場合、Igor がグラフのサイズに適したキャップ幅を選択します。

この場合、キャップ幅はマーカーのサイズの2倍に1ポイント加えたサイズに設定されます。最良の結果を得るには、キャップの幅は奇数であるべきです。



6. キャップの太さとエラーバーの太さを設定できます。

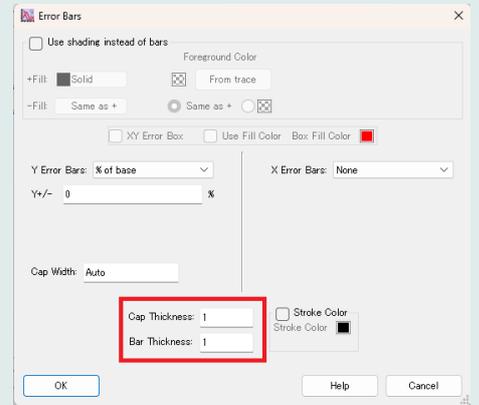
これらの設定の単位はポイントです。

小数も使用できます。

エクスポートまたは印刷時には、整数倍ではない太さが適切に生成されます。

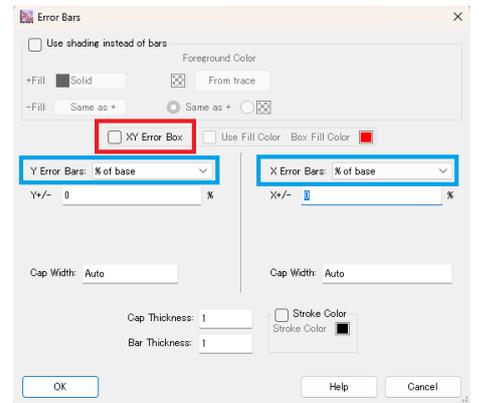
Cap Thickness をゼロにすると、キャップは描画されません。

Bar Thickness をゼロにすると、エラーバーは描画されません。



7. XY Error box チェックボックスをオンにすると、不確実な領域を示すためにエラーバーではなく、ボックスが表示されます。XY Error box は Y Error Bars と X Error Bars の両方でモードを選択すると有効になります。

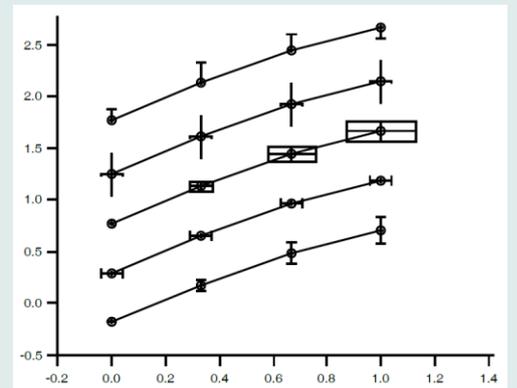
エラー値の1つ以上が NaN の点については、ボックスは描画されません。



8. エラーバー付きのグラフの簡単な例を示します。

一番上のトレースは +/- Wave モードで +wave のみを使っています。

エラーバーの最後の値を負の値にすることで、エラーバーの方向を反転させています。

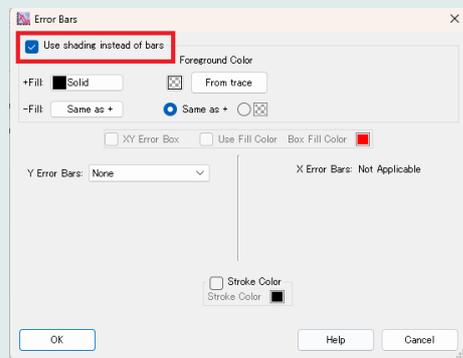


エラー陰影

Igor Pro 7 以降では、エラーバーの代わりに陰影を使うことができます。

1. Modify Trace Appearance ダイアログから Error Bars ダイアログを開き、Use shading instead of bars チェックボックスをオンにして、陰影を有効にします。

陰影モードでは、shade キーワードで使われるパラメーターに応じて +error から -error レベルの間、または +error からデータ、データから -error の間を塗りつぶします。



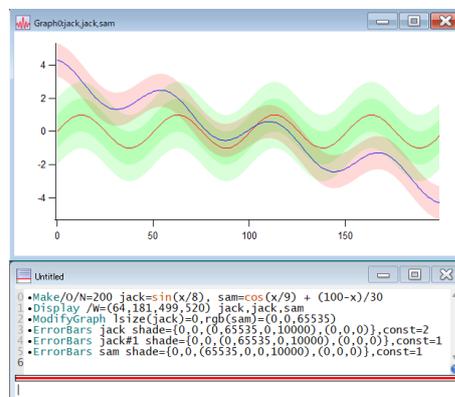
2. 次のコマンドは、複数の重なり合うエラーの透明度の陰影を示しています。

新しい Experiment を作成して、コマンドウィンドウで次のコマンドを実行します。

```
Make/O/N=200 jack=sin(x/8), sam=cos(x/9) + (100-x)/30
Display /W=(64,181,499,520) jack,jack,sam
ModifyGraph lsize(jack)=0,rgb(sam)=(0,0,65535)
ErrorBars jack shade={0,0,(0,65535,0,10000),(0,0,0)},const=2
ErrorBars jack#1 shade={0,0,(0,65535,0,10000),(0,0,0)},const=1
ErrorBars sam shade={0,0,(65535,0,0,10000),(0,0,0)},const=1
```

Windows においては、古い GDI グラフィック技術では陰影は機能しません。

詳細は、マニュアル III-506 Graphics Technology を参照してください。

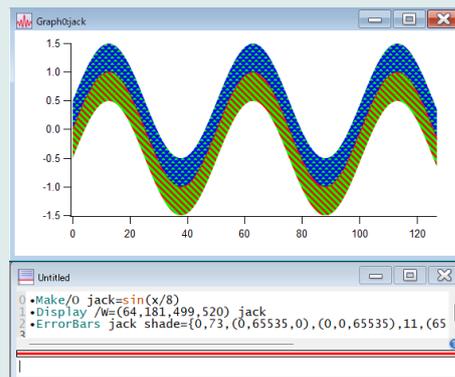


3. 次のコマンドは、異なる +error と -error の陰影と、パターンを使った例を示しています。

新しい Experiment を作成して、コマンドウィンドウで次のコマンドを実行します。

```
Make/O jack=sin(x/8)
Display /W=(64,181,499,520) jack
ErrorBars jack
shade={0,73,(0,65535,0),(0,0,65535),11,(65535,0,0),(0,65535,0)},const=0.5
```

詳細は、ErrorBars shade キーワードの説明を参照してください。



エラー楕円

Igor Pro 9.0 以降では、棒や箱の代わりにエラー楕円を指定することができます。エラー楕円とは、X と Y のエラーと、X と Y の相関関係の両方を示しています。

各データポイントに 1 行ずつ、3 列のウェーブを使ってエラー楕円用のデータを提供する必要があります。このウェーブは、Error Bars ダイアログで Ellipse Wave とラベル付けされています。これは、ErrorBars コマンドで ewave=ew と指定されます。

ew の各行には、1 つのデータポイントのエラー楕円に関する情報が含まれています。ew 列の解釈は、Error Bars ダイアログ (Modify Trace Appearance ダイアログから開く) のモードに依存し、ErrorBars コマンドの ELLIPSE キーワードの mode パラメーターによって指定されるモードに依存します。

mode=0: ew は、X の標準偏差、Y の標準偏差、X と Y の相関関係を含んでいます。

mode=1: ew は、X の分散、Y の分散、X と Y の共分散を含んでいます。

ErrorBars コマンドで ew を指定する場合、各トレースのデータポイントに 3 列 1 行の 2D ウェーブとなるのであれば、サブレンジの指定を含めることができます。

マニュアル II-321 Subrange Display Syntax を参照してください。

エラー楕円の色

各エラー楕円の色は、楕円が付けられているデータポイントの色から取得されます。

通常、それは単にトレースの色です。

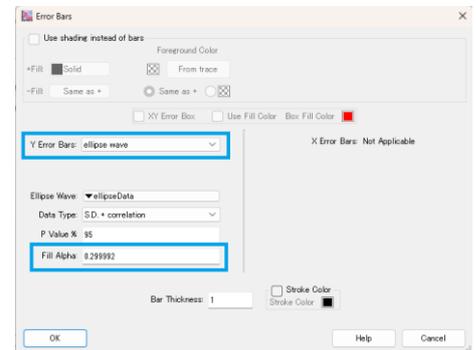
トレースの色を $f(z)$ として使うか、ポイントのカスタマイズによって、個々のデータポイントとそれに対応する楕円の色を変更することができます。

マニュアル II-306 Customize at Point を参照してください。

Error Bars ダイアログの Fill Alpha 設定を使って、指定したトレースのすべてのエラー楕円の不透明度を指定できます。

これは、すべてのエラー楕円の塗りつぶし色のアルファを指定し、データポイントのアルファを上書きします。

ErrorBars コマンドでは、ELLIPSE キーワードを使って 0（完全に透明）から 65535（完全に不透明）までの値でアルファを指定します。



アルファについての補足情報は、マニュアル IV-13 RGBA Values を参照してください。

エラー楕円の例

次のコマンドは、エラー楕円の使い方の例を示しています。

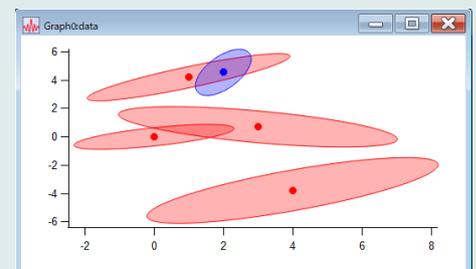
新しい Experiment を作成して、コマンドウィンドウで次のコマンドを実行します（コメントの入力は任意です）。

```
Make/O/N=5 data = 5*sin(x)
Display data
ModifyGraph mode=3, marker=19 // マーカーモード solid
circle marker

Make/O/N=(5,3) ellipseData // エラー楕円のパラメーター

// 列 0: x の標準偏差
ellipseData[0][0]={0.944527,1.19611,0.32858,1.64494,1.7
1742}

// 列 1: Y の標準偏差
ellipseData[0][1]={0.357613,0.680198,0.665561,0.581878,
0.948245}
```



```
// 列 2: X と Y の相関。[-1, 1] の範囲
ellipseData[0][2]={0.562396,0.707809,0.396551,-
0.516456,0.554322}

ErrorBars data, ELLIPSE={0, .95, .3*65535},
ewave=ellipseData

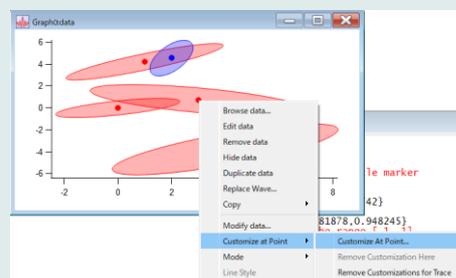
// ポイント 2 を青に変更
ModifyGraph rgb(data[2])=(0,0,65535)
```

マニュアルには、Function...End が記載されていますが、直接実行するときには不要です。

データポイントにおけるカスタマイズ

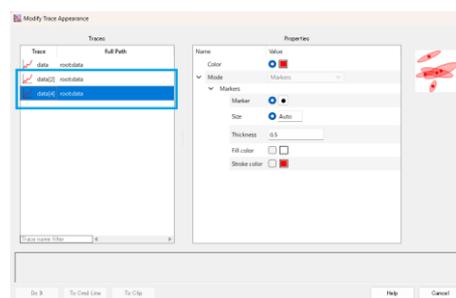
棒、マーカー、ドット、ラインで表示されるグラフのトレース上の個々のポイントの外観を、ゼロモードにカスタマイズできます。

1. インタラクティブに行うには、トレース上の目的のポイントを右クリックし、コンテキストメニューから **Customize at Point** → **Customize At Point** を選択します。

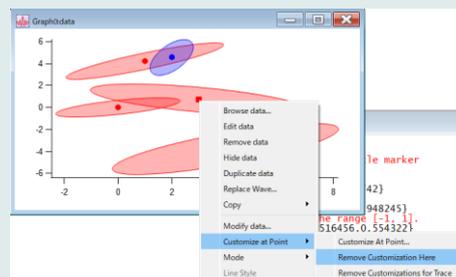


2. Modify Trace Dialog に、トレースリストの項目に各括弧で囲まれた番号が表示されます。

そのような項目が選択されると、カスタマイズ可能なプロパティのみがダイアログに表示されます。



3. ポイントを右クリックし、Customize at Point → Remove Customization Here を選択することで、1つのポイントのカスタマイズを削除することができます。



4. トレースを右クリックし、Customize at Point → Remove Customization from Trace を選択すると、トレースからすべてのカスタマイズを削除することができます。

