

CONTENTS

ビジュアルヘルプ - グラフ (2)	3
トレースの修正	3
修正するトレースの選択.....	3
表示モード.....	3
マーカー	4
マーカーの色.....	5
テキストマーカー.....	6
矢印マーカー.....	7
線のスタイルと太さ	8
塗りつぶし.....	9
ウェーブフォームを持つ棒グラフ.....	9
XY ペアを持つ棒グラフ.....	10
グループ化、積み重ね、モードの追加	10
トレースの色.....	12
補助 (Z) ウェーブからのトレースのプロパティ設定	13
f(z) で色を指定	13
f(z) で色を指定する例.....	14
f(z) でマーカーサイズを指定.....	14
f(z) でマーカー番号を指定	14
軸の単位を使った f(z) でマーカーの大きさを指定	15
f(z) でパターン番号を指定	15
f(z) による色の指定の凡例の例.....	15
トレースのオフセット.....	17
トレースの乗算器.....	18
トレースを隠す	18
複素数表示モード.....	18
ギャップ	19
エラーバー	19
エラーシェーディング	20
エラー楕円.....	21
エラー楕円の色.....	21

エラー楕円の例.....	22
ポイントでのカスタマイズ.....	23

ビジュアルヘルプ – グラフ (2)

トレースの修正

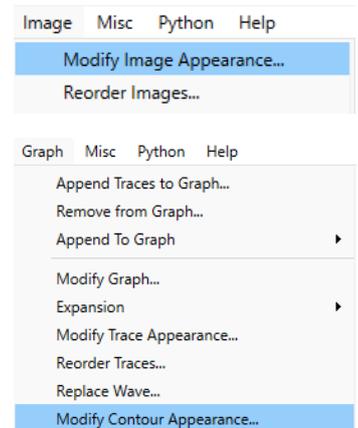
Graph メニューから Modify Trace Appearance を選択するか、グラフ内のトレースをダブルクリックすることで、各トレースのグラフ上での表示方法を指定できます。

画像プロットの場合、Modify Trace Appearance ではなく、Image メニューから Modify Image Appearance を選択してください。

コンタープロットでは、通常、Modify Contour Appearance を選択する必要があります。

これを使って、すべてのコンターの表示を一度にコントロールします。

ただし、1本の等高線を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

外観を変更したいトレースを Traces リストから選択してください。

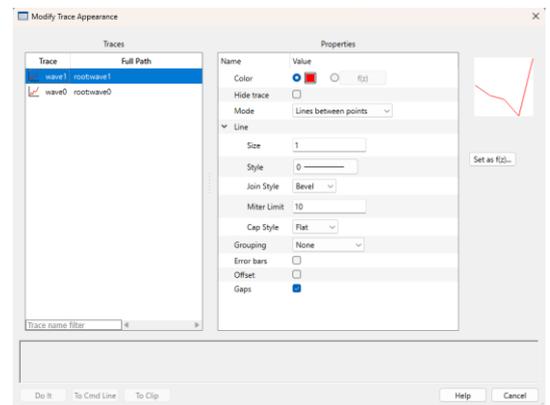
グラフ内のトレースをダブルクリックしてこのダイアログを表示した場合、そのトレースは自動的に選択されます。

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

選択が完了したら、選択したトレースの設定を変更できます。

この操作を行った後、別のトレースまたはトレースのセットを選択し、さらに変更を加えることができます。

Igor は行ったすべての変更を記憶するため、ダイアログを一度開くだけで全ての操作を完了できます。

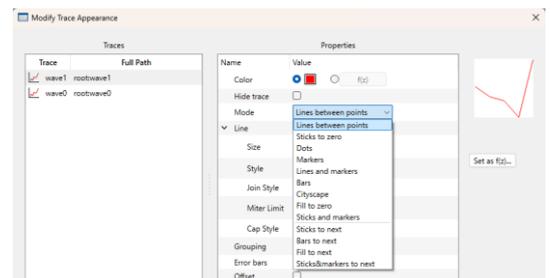


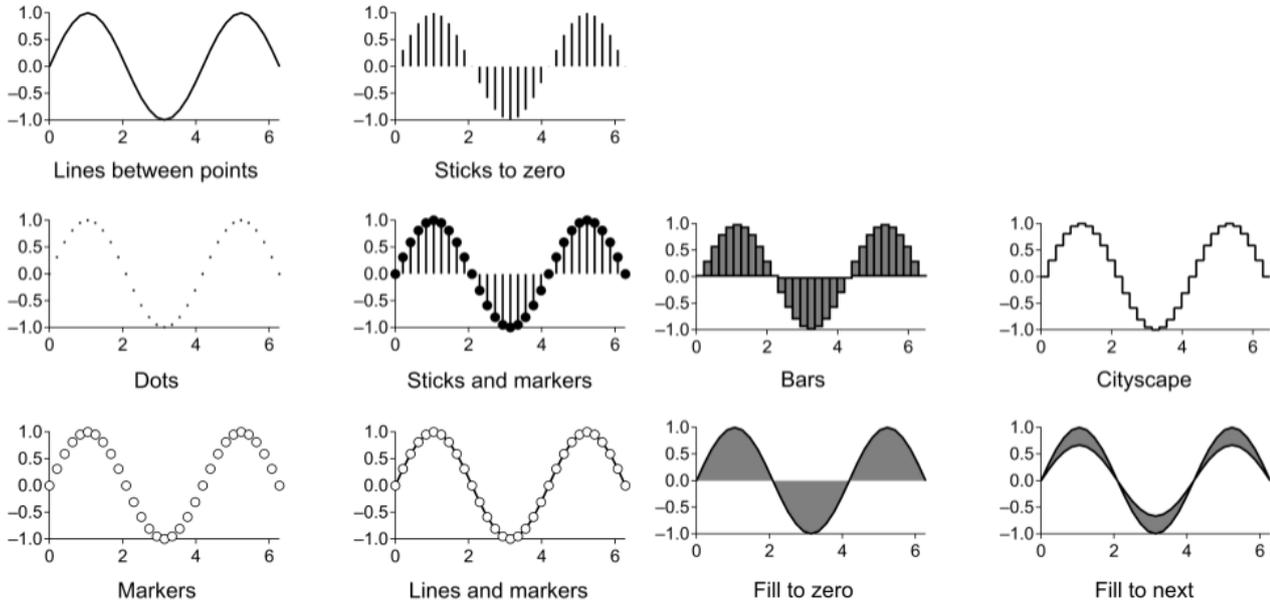
表示モード

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

一般的に使われるモードは、Lines Between Points、Points、Markers、Lines and Markers、Bars です。

他のモードでは、Y=0 または別のトレースに対して、Sticks、Bars、Fills を描画できます。





マーカー

Markers または Lines and Markers モードを選択した場合、マーカーの種類、マーカーのサイズ、マーカーの太さ、およびマーカーを透明にするか色で塗りつぶすかを選択することもできます。

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、Igor がグラフのサイズに適したマーカーサイズを選択します。

マーカーの太さは 0 から 50 ポイントまでの小数です。

小数点は画面上では表示されない場合がありますが、エクスポートや印刷されたグラフィックでは確認できます。

マーカーの太さを 0 に設定すると、マーカーが消えます。

マーカーのサイズと太さの間には相互作用があります。

マーカーを対称にするために必要であれば、Igor はマーカーのサイズを調整します。

マーカーの調整前の幅と高さは、マーカーサイズの設定値 s に対して $2*s+1$ ポイントです。

マーカーと対応するマーカー番号の一覧表は以下の通りです。

+	×	*	⊗	⊠	□	△	◇
0	1	2	3	4	5	6	7
○	—		田	⊠	□	⊗	⊠
8	9	10	11	12	13	14	15
■	▲	◆	●	/	\	▽	▼
16	17	18	19	20	21	22	23
▽	◇	◆	◇	◇	◆	◇	△
24	25	26	27	28	29	30	31
▲	△	▲	▽	▼	▽	▲	#
32	33	34	35	36	37	38	39
◇	○	⊕	⊗	△	◁	◀	◁
40	41	42	43	44	45	46	47
▷	▶	▷	◊	◆	◊	○	●
48	49	50	51	52	53	54	55
◊	▲	▲	◇	◆	◊	×	
56	57	58	59	60	61	62	

カスタムの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

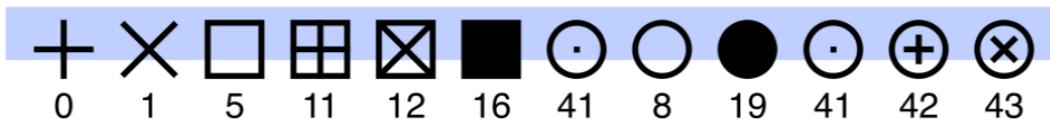
Igor はグラフマーカーの3つの色（トレース色、ストローク色、塗りつぶし色）をコントロールします。

トレース色は、トレース全体に対する選択された色であり、どのトレースモードでも同じです。

ストローク色は、マーカーの輪郭線となる線の色です。
デフォルトでは、ストローク色はトレース色と同じです。

塗りつぶし色は、中空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色は設定されておらず、中空マーカーの内部を通して背景オブジェクトが見えるようになっています。

以下は、トレースカラーを黒、マーカーサイズを 10、マーカーのストローク太さを 2 ポイントに設定した一部のマーカーのサンプルです。



青い長方形は、中空マーカー（5、11、12、8、41、42、43）の内部に透けて見えます。
ソリッドマーカー（16、19）はトレース色が黒であるため、黒一色で表示されます。

ストローク色は Modify Trace Appearance ダイアログで設定できます。
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 コマンドも使用できます。
次の図では、コマンドを使ってストロークの色を緑に設定しました：

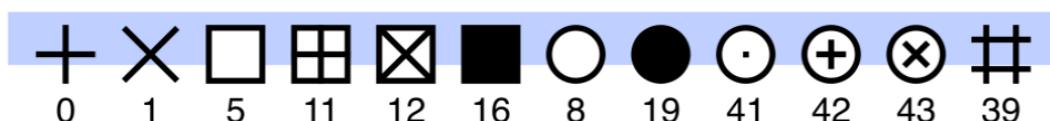
```
ModifyGraph mrkStrokeRGB=(1,52428,26586)
```



ストローク色がトレース色を上書きするため、すべてのマーカーの輪郭が緑色になりました。
ソリッドのマーカーは黒で、緑の縁取りが施されています。

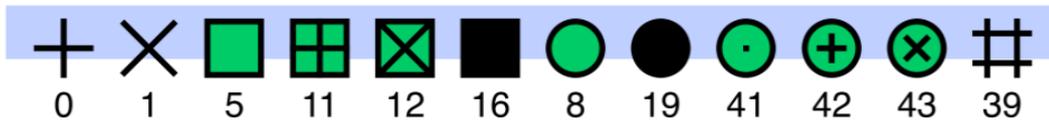
中空マーカーの内部を不透明にすることができます。
Modify Trace Appearance ダイアログで、Fill チェックボックスをオンにします。
このコマンドは：

```
ModifyGraph opaque=1
```



デフォルトでは、不透明な中空マーカーの内部の色は白です。
内部の色は、Modify Trace Appearance ダイアログで塗りつぶし色を選択するか、ModifyGraph の mrkFillRGB キーワードを使って変更できます。
コマンドを使って塗りつぶしを設定します：

ModifyGraph mrkFillRGB=(1,52428,26586)

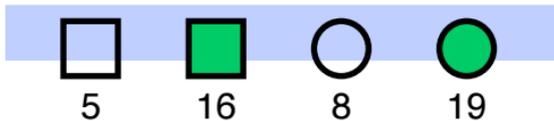


実線マーカーのストローク色または中空マーカーの塗りつぶし色を設定することで、次の例のように同じ結果を得ることが可能です。

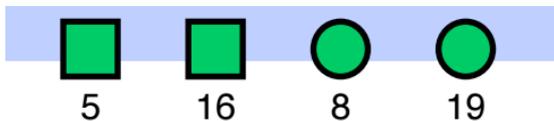
トレース色を緑に設定：



ストローク色を黒に設定すると、マーカーは黒の輪郭と緑の内部を持つようになります：



最後に塗りつぶし色を緑に設定し、中空のマーカーが緑で塗りつぶすと、次の結果となります：



これで、塗りつぶしと中空のマーカーは同じ見た目です。

これは歴史的な特異点です - Igor Pro 9.0 以前では塗りつぶし色を白から変更できませんでした。

Igor 9 以前では、11 や 42 などの内部にマークがある中空マーカーは、色を塗りつぶす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、以下のいずれかをテキストマーカーとして使うように Igor に指示することもできます：

- フォントの1つの文字
- テキストウェーブの内容
- 数値ウェーブの内容

フォント内の1つの文字が主に注目されるのは、フォントで利用可能な特殊記号を使いたいが、それが Igor の組み込みマーカーに含まれていない場合です。

指定された文字は全てのデータポイントに使われます。

残りのオプションは、XY プロットに第3の値を表示する方法を提供します。

例えば、地震の規模と発生日を示すグラフでは、テキストウェーブを使ってテキストマーカーを供給することで、地震の発生地点も表示できます。

あるいは、地震の発生地点と日付のグラフでは、数値ウェーブを用いてマグニチュードを表示し、テキストマーカーを提供することも可能です。

XY プロットの各データポイントに対して、対応するテキストまたは数値ウェーブのポイントがマーカーのテキストを提供します。

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

テキストマーカーを作成するには、Markers または Lines and Markers 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 ボタンを選択します。

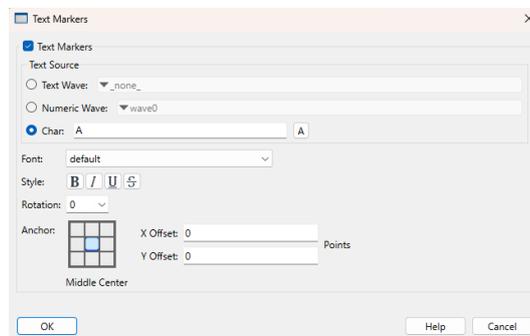
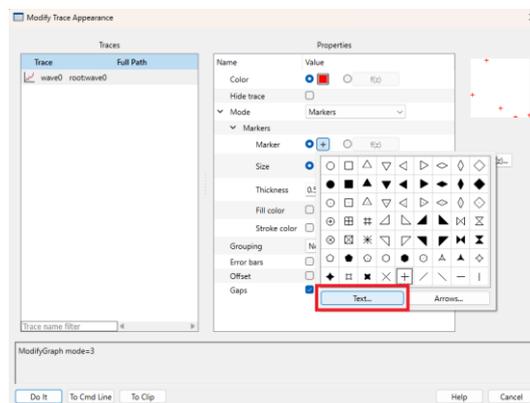
これにより Text Markers サブダイアログが表示され、テキストのソースやマーカーのフォント、スタイル、回転などのプロパティを指定できます。

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させることはできますが、個々のデータポイントに対してオフセットと回転を設定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。

データポイント上に文字マーカーを正確に中心に配置するには、X および Y オフセットを何度か調整する必要がある場合があります。

例えば、テキストを各データポイントの真上に中央揃えにするには、Anchor のセクションで Middle Bottom を選択し、Y Offset を 5~10 ポイントに設定します。

一部の項目を他の項目とは異なる方法でオフセットする必要がある場合は、タグを使う必要があります（ヘルプ Tags を参照）。



Igor は、マーカーサイズ（Modify Trace Appearance ダイアログで設定）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。

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の両方を表示したい場合があります。

そのためには、グラフ上でウェーブを 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後、Graph→Append Traces to Graph を選択して、ウェーブの 2 つ目のコピーを追加します。

このコピーには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、流れや勾配場を示すベクトルプロットの作成に使用できます。

矢印マーカーはかなり特殊な用途で、少し事前準備が必要となります。

非常に簡単な例を以下に示します：

// XY データの作成

```
Make/O xData = {1, 2, 3}, yData = {1, 2, 3}
```

```
Display yData vs xData
```

```
ModifyGraph mode(yData) = 3
```

// グラフの作成

// マーカーモード

// 各ポイントの長さや角度を制御するための矢印データウェーブを作成

```
Make/O/N=(3,2) arrowData
```

// 矢印の長さや角度を制御

```
Edit /W=(439,47,820,240) arrowData
```

// arrowData にいくつかデータを投入

```
arrowData[0][0] = {20,25,30}
```

// 列 0: ポイント単位の矢印の長さ

```
arrowData[0][1] = {0.523599,0.785398,1.0472}
```

// 列 1: ラジアン単位の矢印の角度

// トレースを矢印モードに設定して矢印を表示

```
ModifyGraph arrowMarker(yData) = {arrowData, 1, 10, 1, 1}
```

// RGB カラーウェーブを作成

```
Make/O/N=(3,3) arrowColor
```

```
Edit /W=(440,272,820,439) arrowColor
```

// カラーウェーブにいくつかの色を格納

```
arrowColor[0][0] = {65535,0,0}
```

// 赤

```
arrowColor[0][1] = {0,65535,0}
```

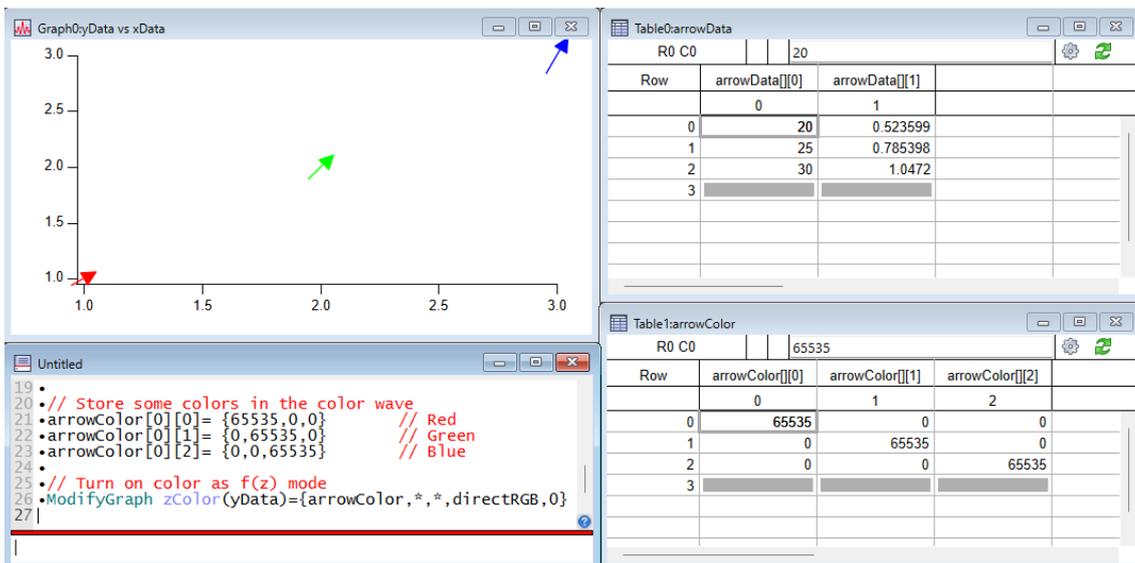
// 緑

```
arrowColor[0][2] = {0,0,65535}
```

// 青

// f(z) モードとして色を設定

```
ModifyGraph zColor(yData)={arrowColor,*,*,directRGB,0}
```



詳細については、ModifyGraph コマンドにおける arrowMarker キーワードの説明を参照してください。

線のスタイルと太さ

Lines between points、Lines and markers、または Cityscape モードを選択した場合、線のスタイルも選択できます。

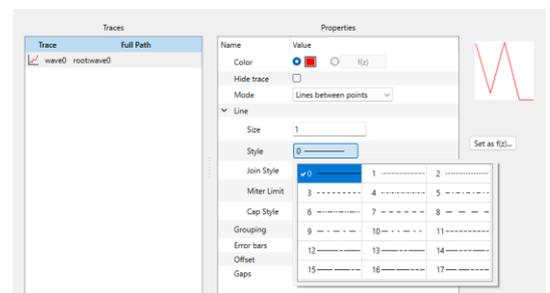
Line セクションの Style 項目を使用して、破線のパターンを変更できます。

Markers モード以外のモードでは、線の太さを設定できます。

線の太さはポイント単位で指定でき、小数点以下の値も指定可能です。

線の太さがゼロの場合、線は表示されなくなります。

詳細は、ヘルプ Dashed Lines を参照してください。

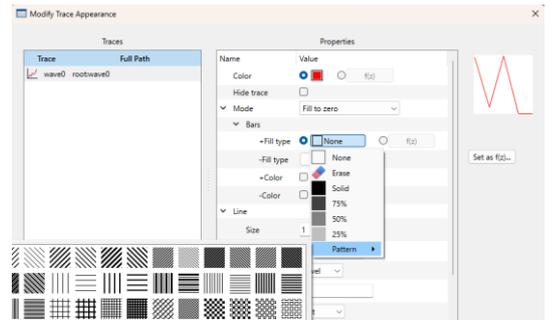


塗りつぶし

Bars と Fill to zero モードにおけるトレースに対して、Igor は塗りつぶしタイプの選択肢を提供します。

塗りつぶしタイプは、透明を意味する None、白で不透明な塗りつぶしを意味する Erase、Solid、または3種類のグレーパターンから選択できます。

パレットからパターンを選択することもでき、正方向領域と負方向領域に対して、それぞれ独立して塗りつぶしの種類と色を選択できます。



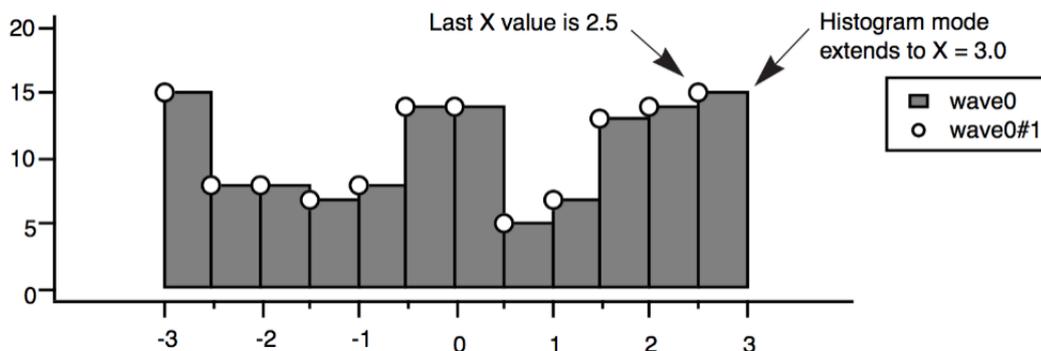
詳細は、ヘルプ Fill Patterns と Gradient Fills を参照してください。

ウェーブフォームを持つ棒グラフ

Bars モードが通常の連続 X 軸（カテゴリ軸ではない）上にプロットされたウェーブフォームに使われる場合、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、特定のポイントの X 値から、次のポイントの X 値（その値自体は含まない）まで描画されます。

このようなバーは、通常ヒストグラムにおいて2つの X 値の間に収まるカウント数を示すために用いられるため、「ヒストグラムバー」と呼ばれます。



バーを X 値で中央揃えにしたい場合は、カテゴリプロットを作成してください。

これは従来の棒グラフに適しています（ヘルプ Category Plots を参照）。

ただし、ウェーブの X 値を調整することで、平らな部分が従来の棒グラフと同様に元の X 値を中心として表示されるようになります。

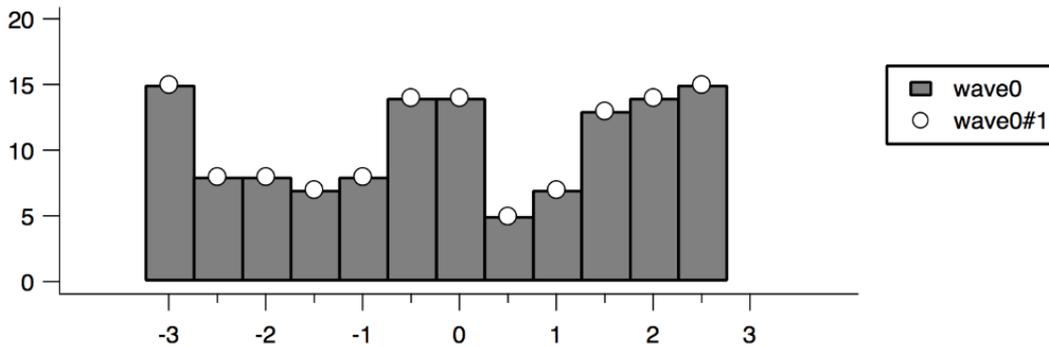
データを実際に変更せずにこれを行う方法の一つは、グラフ上のトレースをバー幅の半分だけオフセットすることです。

ドラッグするだけで済みます。

あるいは 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 0.5 X 単位です：

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



XY ペアを持つ棒グラフ

X 軸が XY ペアによってコントロールされている場合、各バーの幅は2つの X 値によって決定されます。1つ目の X 値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。

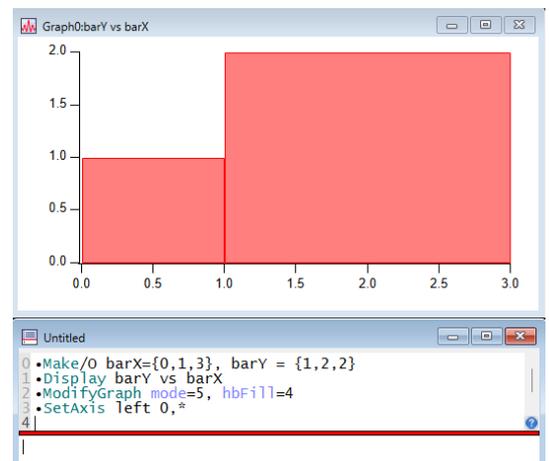
1本の棒グラフを想像すると、これは2ポイントの XY ペアを必要とします。

1つ目の X 値でバーの左端を設定し、次の X 値で右端を設定します。

一般化すると、N本のバーを持つ棒グラフの場合、X ウェーブと Y ウェーブには N+1 ポイントが必要です。

以下は3つの XY ポイントを用いて2本のバーを生成する例です：

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



グループ化、積み重ね、モードの追加

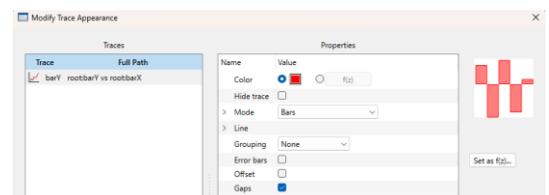
通常 $y=0$ に描画する4つのモード (Sticks to zero, Bars, Fill to zero, Sticks and markers) では、次のトレースの Y 値まで描画するバリエーションを選択できます。

4つのバリエーションモードは以下の通りです：Sticks to next, Bars to next, Fill to next, Sticks&markers to next.

この文脈における「next」とは、Modify Trace Appearance および Reorder Traces ダイアログ内のトレース一覧において、選択したトレースの直後 (下) にリストされているトレースを指します。

これらの4つのモードのいずれかを選択すると、Igor は自動的に Grouping ポップアップメニューから Draw to next を選択します。

このメニューから Add to next モードと 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われます。

利用可能なモードは None、Keep with next、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、ヘルプ Category Plots で説明しています。

Draw to next は、通常 $y=0$ 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ペアに対してプロット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 値と同じである必要があります。
そうでない場合、次のトレースは現在のトレースの下端と一致しなくなります。

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 値に追加するよう Igor に指示します。

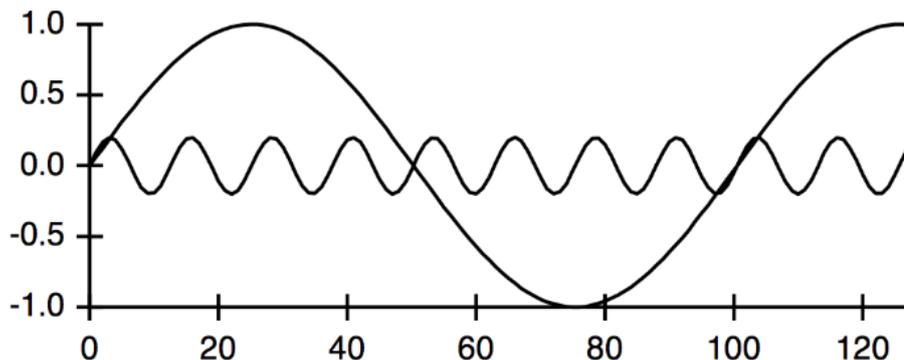
次のトレースも「Add to next」を使っている場合、その追加が最初に行われ、以降も同様です。

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場合、このモードは Draw to next と同様に動作します。

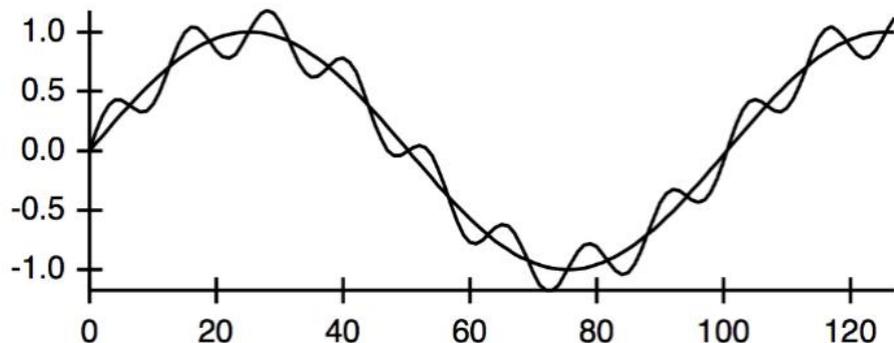
Stack on next は Add to next と同様に機能しますが、Y 値のバックトラックが許可されません。

つまり、次のトレースの Y 値が正の場合、負の値はゼロとして扱われ、次のトレースが負の場合、正の値はゼロとして扱われます。

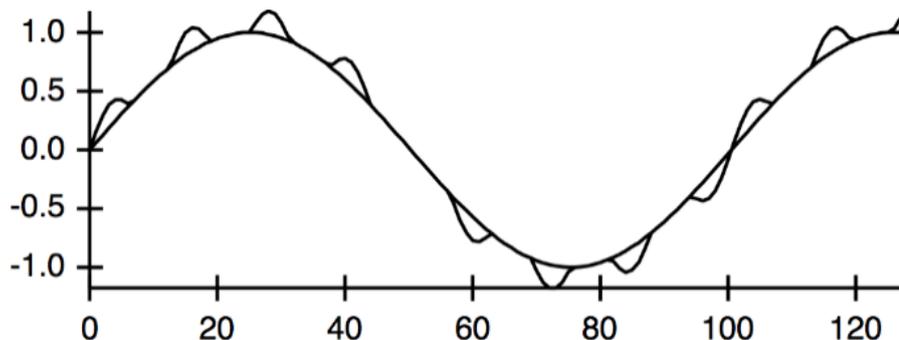
以下は、小さな正弦波と大きな正弦波の通常のプロットです：



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：

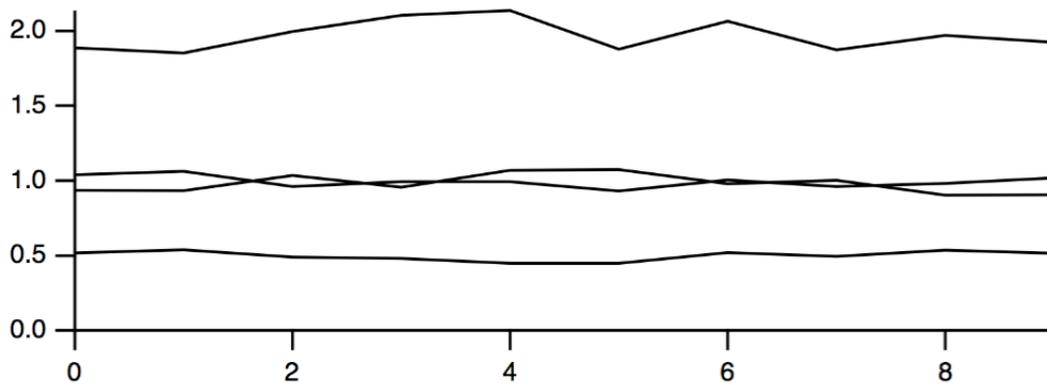


Stack on next を適用すると：

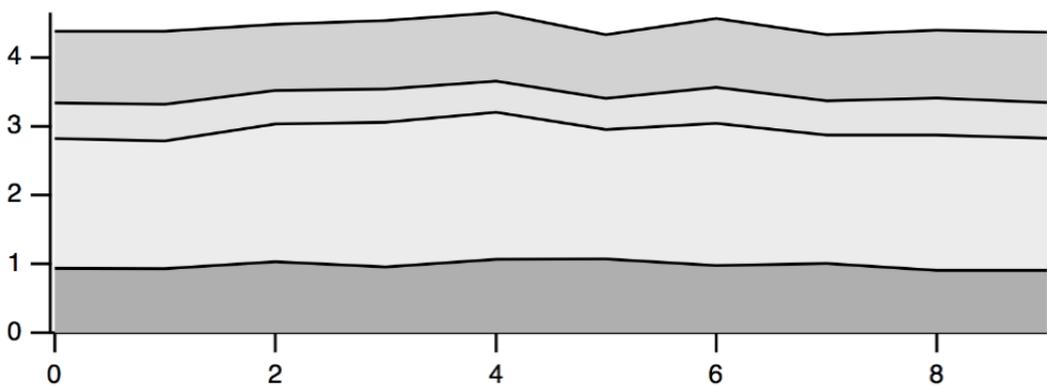


Fill to next モードを使って複数のウェーブをグラフにプロットすることで、レイヤーグラフを作成できます。
データによっては、次のグループに追加モードを使うこともできます。

例えば、次の通常のプロットでは、各トレースが地層の厚さを表している可能性があります：



Fill to next と Add to next モードを使うことで、レイヤーをより理解しやすい形で表示できます：

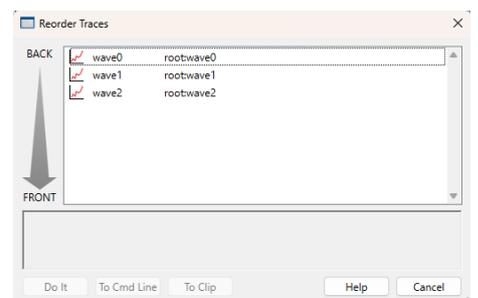
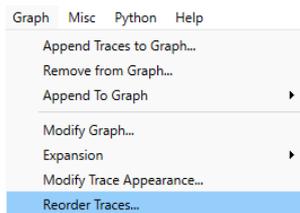


グループ化モードは次のトレースの同一性に依存するため、グラフ内のトレースの順序を調整する必要がある場合があります。

この操作は Reorder Traces ダイアログで行えます。

Graph→Reorder Traces を選択してください。

選択したトレースをリスト内で上下にドラッグして順序を調整し、適切な位置にドロップします。



注記

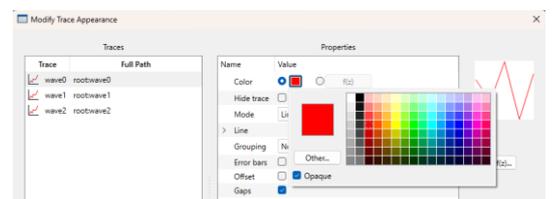
様々なグループ化、追加、重ね合わせモードで使うすべてのウェーブは、同じ数のポイントと X 方向のスケールを持ち、同じ軸を使って表示される必要があります。

トレースの色

トレースの色は、カラーポップアップパレットから選択できます。

色に加え、カラーポップアップの alpha プロパティを使って不透明度を指定できます。

アルファ値 1.0 ではトレースが完全に不透明になり、アルファ値 0.0 では完全に透明になります。



補助 (Z) ウェーブからのトレースのプロパティ設定

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、およびパターン番号をポイントごとに設定できます。

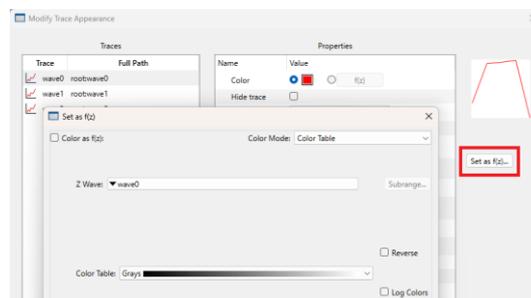
補助ウェーブは「Z ウェーブ」と呼ばれます。

これは、他のウェーブがトレース上の各ポイントの X 座標と Y 座標をコントロールする一方で、Z ウェーブが第三の特性をコントロールするためです。

例えば、地震の発生位置にマーカーを配置し、それぞれ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。

地震の深さはマーカーの色で示し、地震の種類は異なるマーカーの形状で示すことができます。

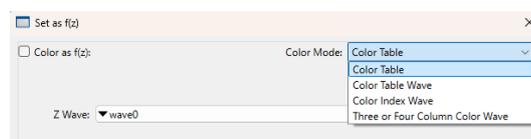
トレースのプロパティを Z ウェーブの関数として設定するには、Modify Trace Appearance ダイアログ内の Set as f(z) ボタンをクリックし、Set as f(z) サブダイアログを表示します。



f(z) で色を指定

色を f(z) として定義する場合、4つのモードがあります：Color Table、Color Table Wave、Color Index Wave、Three or Four Column Color Wave です。

Color Mode メニューからモードを選択します。



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 ウェーブ値を内蔵の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オンの場合、マッピングは対数関数となります。

それ以外の場合は線形です。

Log Colors オプションは、zWave が数十年にわたる期間をカバーし、より細かい値の変化を詳細に表示したい場合など有効です。

zMin および zMax の設定は、Z ウェーブの値の範囲をカラーテーブルにマッピングする範囲を定義します。

範囲外の値は、範囲の末尾の色を継承します。

zMin または zMax で Auto を選択した場合、Igor は Z ウェーブ内で見つけ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 はそれらのデータポイントを X または Y データが NaN の場合と同様に扱います。

これはメインダイアログの Gaps 設定に依存します。

Color Table Wave モードは、Color Table モードと同様ですが、内蔵のカラーテーブルではなく、ユーザーが提供するカラーテーブルウェーブによって色が決定される点が異なります。

詳細はヘルプ 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 軸スケーリングに線形または対数的にマッピングすることで導出されます。

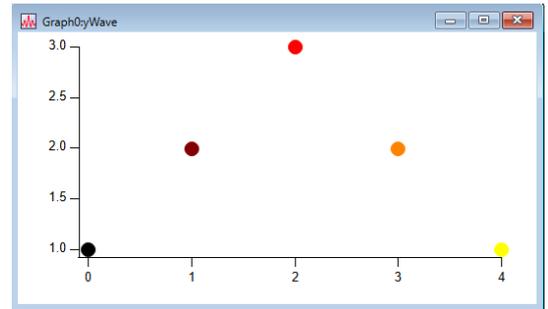
これは、ModifyImage cindex が画像値 (Z ウェーブ値の代わりに) を 3 列のカラーインデックス行列内の色にマッピングする方法に似ています。

詳細はヘルプ Direct Color Details を参照してください。

f(z) で色を指定する例

以下のコマンドは Color Table モードを使い、Z ウェーブに基づいてトレースのポイントを着色します。

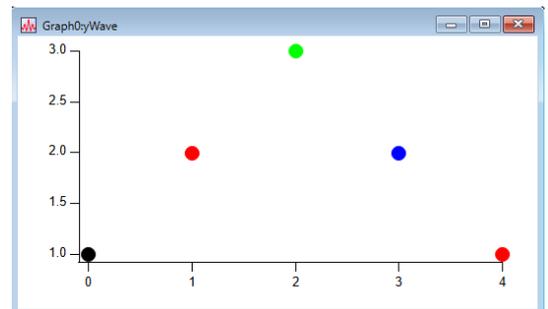
```
Make/N=5 yWave = {1,2,3,2,1}
Display yWave
ModifyGraph mode=3, marker=19, msize=5
Make/N=5 zWave = {1,2,3,4,5}
ModifyGraph zColor(yWave)={zWave, *, *, YellowHot}
```



代わりに3列のウェーブを作成し、RGB値を入力するように編集すると：

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

Row	directColorWave[][0]	directColorWave[][1]	directColorWave[][2]	
	0	1	1	
0	0	0	0	Black
1	65535	0	0	Red
2	0	65535	0	Green
3	0	0	65535	Blue
4	65535	0	26214	Hot Pink



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できます：

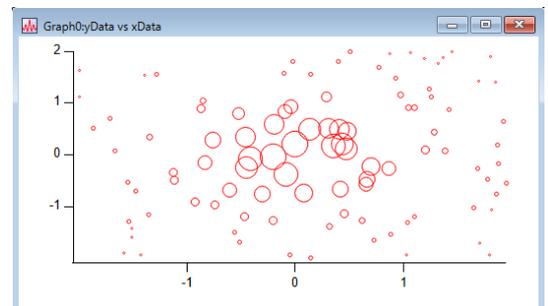
```
ModifyGraph zColor(yWave)={directColorWave, *, *, directRGB}
```

f(z) でマーカーサイズを指定

Marker size as f(z) は、カラーテーブルモードにおける Color as f(z) と同様に機能しますが、Z 値が min と max のマーカー設定で定義したマーカーサイズの範囲にマッピングされる点が異なります。

この例では、マーカーサイズを関数として第3の値を示しています：

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



f(z) でマーカー番号を指定

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 ウェーブを作成する必要があります。

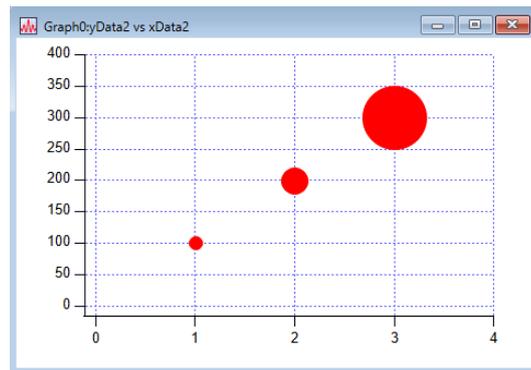
マーカー番号コードについては、「マーカー」のセクションを参照してください。

軸の単位を使った f(z) でマーカーの大きさを指定

Igor Pro 9.0 以降では、マーカーサイズを軸の単位で指定できます。

例：

```
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
// マーカーモード、塗りつぶし円マーカー
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、マーカーサイズをウェーブデータ zData2 によってコントロールするように設定します。このデータの値は左軸に合わせてスケーリングされます。

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上で 50 単位に設定します。

指定された軸はグラフ上の任意の軸で構いません。

その軸が後で削除された場合、マーカーサイズは「軸なし」状態に戻り、軸を指定していない状態となります。

軸の単位でマーカーサイズをスケーリングする場合、zmrkSize キーワードの zMin、zMax、mrkMin、mrkMax パラメータに指定した値は、マーカーサイズに影響を与えません。

ただし、指定した軸を削除した場合、これらのパラメータがトレースのマーカーサイズをコントロールします。

マーカーサイズのスケーリングが、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、そのスケールを表示するために自由軸を使用できます。

これを行うには、目的の範囲を持つ自由軸を作成し、それを zmrkSize キーワードの軸として設定します。

自由軸に関する説明は「軸の種類」のセクションを参照してください。

指定した軸が対数軸の場合、Z 値の解釈が変化します。

対数軸と同じポイント長を持つ概念的な軸を想像してください。

概念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概念的に設定します。

特定の Z 値に対するマーカーサイズは、概念軸上の Z 単位の長さです。

これを視覚化するために、この概念軸を表す自由軸を作成できます。

f(z) でパターン番号を指定

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 ウェーブを作成する必要があります。

パターン番号の一覧については、「塗りつぶしパターン」のセクションを参照してください。

f(z) による色の指定の凡例の例

色を f(z) モードとして使うグラフがある場合、各色が何を表すかを識別する凡例を作成することをお勧めします。

このセクションでは、この目的のために凡例コマンドの機能を使う方法を示します。

以下のコマンドを、一つずつ実行してください：

```
// テストデータの作成
```

```
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
```

```
// シンボルがトレースから取得される通常の凡例を作成
```

```
Legend/C/N=legend0/J/A=LT "\\k(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ためのカラーインデックスウェーブを作成
```

```
Make /O testColorIndex = {0, 127, 225}
```

```
// グラフのトレースを f(z) モードで色を使うように変更
```

```
// Rainbow256 は組み込みカラーテーブルの名前
```

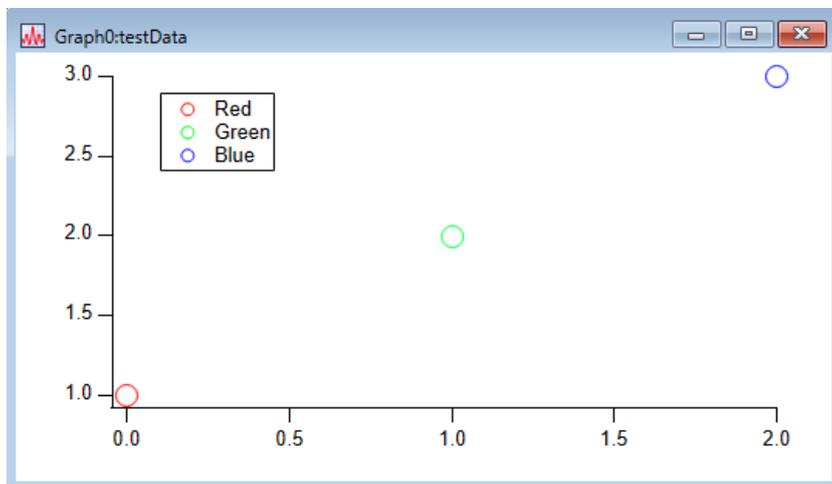
```
// 数値 0 と 255 は、カラーテーブルの最初と最後のエントリに対応するカラーインデックス値を設定
```

```
ModifyGraph zColor(testData)={testColorIndex,0,255,Rainbow256,0}
```

```
// 凡例が色を表示するように変更
```

```
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"
```

結果は次のグラフです：



最後のコマンドでは、凡例で使うマーカーを指定するために `\W` エスケープシーケンスを使い（この場合は円形マーカーの 08）、また、マーカーの太さも指定しています（6 は 1.0 ポイントを意味します）。

`\k` エスケープシーケンスは、`\W` で指定されたマーカーのストローク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 形式で指定され、各色成分は 0 から 65535 の範囲に収まります。

この例では、1つのバックスラッシュは Igor リテラル文字列においてエスケープ文字として扱われるため二重バックスラッシュを使っています。

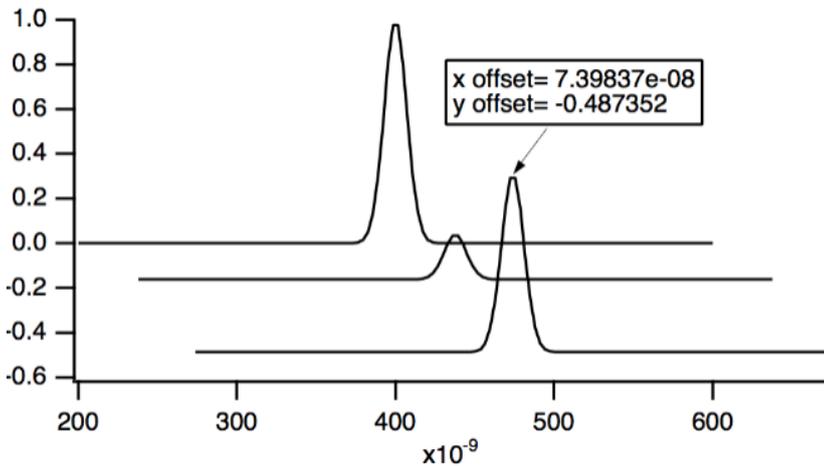
最終テキストにバックスラッシュを含める必要があるため（Igor が `\k` および `\W` にこれを要求するため）、リテラル文字列では二重バックスラッシュを使う必要があります。

Add Annotation ダイアログに凡例テキストを入力する場合、1つのバックスラッシュのみを使います。

するとダイアログは必要なコマンドを生成し、そのコマンドには二重バックスラッシュが含まれます。

トレースのオフセット

グラフ内のトレースを、関連するウェーブデータの変更なしに、水平方向または垂直方向にオフセットできます。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ために有用です。



各トレースにはX オフセットとY オフセットがあり、初期値はいずれもゼロです。

Modify Trace Appearance ダイアログで Offset チェックボックスをオンにすると、トレースの X オフセットと Y オフセットを入力できます。

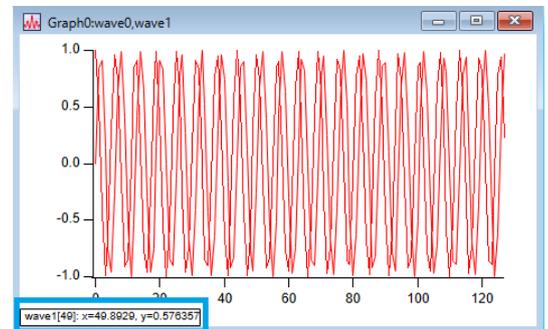
グラフ上でクリックしてドラッグすることでオフセットを設定することもできます。

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ます。マウスを約 1 秒間押し続けてください。

グラフの左下隅に読み取りボックスが表示されます。

読み取り値は、トレースをドラッグする時に X 軸と Y 軸のオフセットを表示します。

指定されたトレースの表示に時間がかかりすぎなければ、画面上でトレースをドラッグしながらその形状を確認できます。



ウェーブをオフセットする時に Shift キーを押すと、Igor はオフセットを水平方向または垂直方向に制限します。

Caps Lock キーを押すとトレースドラッグを無効にできます。

トラックボールユーザーには便利な機能です。

誤って不要な場所にトレースをドラッグした場合は、Edit メニューから Undo を選択してください。

トレースにタグを添付し、その現在のオフセット値を表示することが可能です。

詳細はヘルプ Dynamic Escape Codes for Tags を参照してください。

グラフで自動スケーリングが有効な場合、Igor はトレースオフセットを考慮しようとしています。

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は、Graphs メニューの Set Axis Range 項目を使って自動スケーリングを無効にしてください。

対数軸を使うトレースをオフセットする場合、Igor は軸が対数でない場合と同じ距離だけトレースを移動します。トレースの形状は変化せず、単に移動されます。

ウェーブデータに定数を加え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。

トレースの乗算器

トレースを相殺するだけでなく、トレースをスケーリングするための乗数も指定できます。

プロットに使われる実効値は、「乗数×データ+オフセット」となります。

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、データをゼロで乗算すべきことを意味するものではありません。

通常の（対数ではない）軸では、トレースのオフセットで説明したのと同じクリック&ホールド操作で、トレースをインタラクティブにスケーリングできます。

まず、基準点として機能させるため、カーソル A をトレース上の任意の位置に配置する必要があります。

次に、トレースをクリックして押し続けることでオフセットモードに入った後、Alt キーを押すとオフセットではなく乗数を調整できます。

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、トレース乗算器は対数軸上のトレースをオフセットする代替手段を提供します（覚えておいてください： $\log(a*b) = \log(a) + \log(b)$ ）。

互換性の理由と、トレースオフセット方式が対数軸モードと直線軸モードの切り替えをより適切に処理するため、Igor はトレースをドラッグする時に、オフセットがゼロで乗数がゼロでない場合（デフォルトは「設定なし」を意味します）にのみ乗数方式を使います。

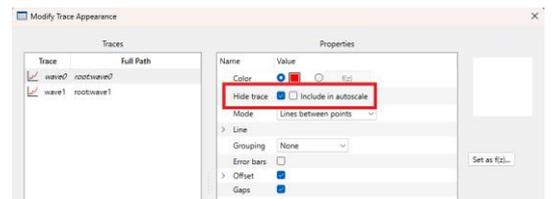
したがって、乗数のテクニックを使うには、コマンドラインまたは Modify Trace Appearance ダイアログのオフセットコントロールを使って、ゼロ以外の乗数を設定する必要があります。

この目的には 1 が適切な設定です。

トレースを隠す

グラフ内のトレースを非表示にするには、Modify Trace Appearance ダイアログで Hide Trace チェックボックスをオンにします。

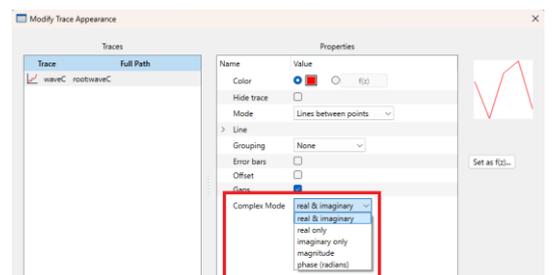
トレースを非表示にした場合、Include in Autoscale チェックボックスを使って、グラフのオートスケール時に非表示にしたトレースのデータを使うかどうかをコントロールできます。



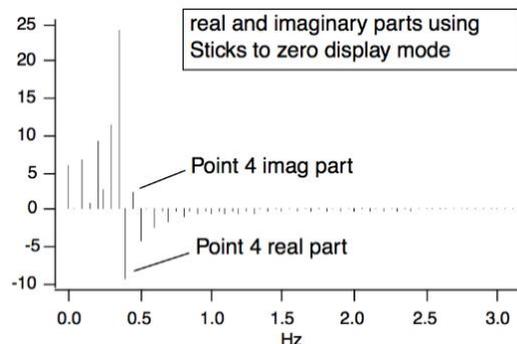
複素数表示モード

複素数データのトレースを表示する時には、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できます。

複素数成分と実数成分を一緒に表示することも個別に表示することもでき、また、実数成分（絶対値）や位相を表示することもできます。



デフォルトの表示モードは Lines between points です。
波形の実部と虚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。



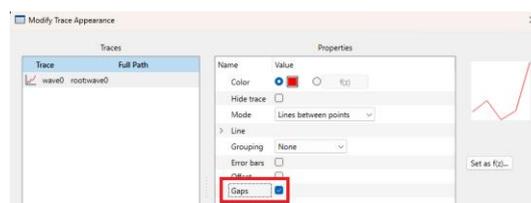
ギャップ

Igor では、ウェーブ内の欠損値または未定義値は浮動小数点値 NaN（「Not a Number／数値ではない」）として保存されます。

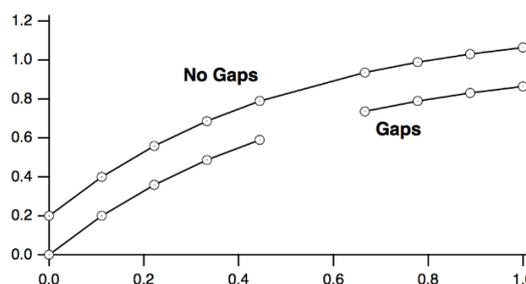
通常、Igor はグラフ上で NaN を空白として表示し、その点にデータが存在しないことを示します。

この設定は Modify Trace Appearance ダイアログの Gaps チェックボックスでコントロールできます。

このチェックボックスがオンの場合（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、欠損値の両側にある利用可能なデータポイントを接続します。

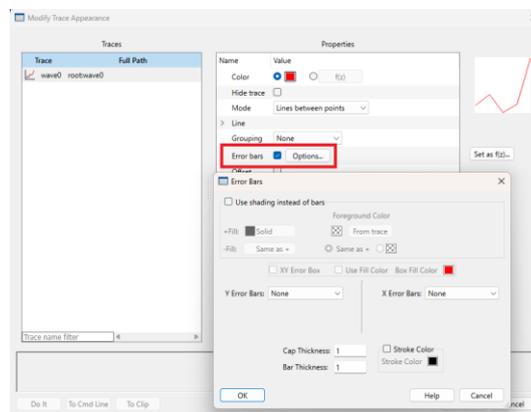


エラーバー

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 サブダイアログを表示します。

エラーバーは、グラフ内のトレースに追加できるスタイルです。誤差値は、定数、所定のポイントにおけるウェーブの値の固定割合、所定の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、誤差値は上方向、下方向、左方向、右方向ごとに個別に設定できます。



Y Error Bars と X Error Bars のポップアップメニューから、目的のモードを選択してください。

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である場合にのみ意味を持ちます。

constant モードでは、X 方向または Y 方向の定数誤差値を入力します。

+/- wave モードでは、正と負の誤差値を供給するウェーブを選択します。

+/- wave を選択すると、ポップアップメニューが表示され、そこから上下または左右の誤差値を供給するウェーブを選択できます。

これらのウェーブは誤差ウェーブと呼ばれます。誤差ウェーブの値は、各ポイントからその誤差バーまでの線の長さを指定するため、通常すべて正であるべきです。これは片側誤差バーをサポートする唯一のモードです。誤差ウェーブは、ベースウェーブと同じ数値タイプや長さを必要としません。エラーウェーブにおける、あるポイントの値が NaN の場合、そのポイントに対応するエラーバーは描画されません。

Cap Width 設定は、エラーバーの端にあるキャップの幅をポイント単位の整数で設定します。キャップ幅を auto (またはゼロ) に設定することも可能です。この場合、Igor はグラフのサイズに適したキャップ幅を選択します。

また、キャップの幅はマーカーのサイズの 2 倍に 1 を加えた値に設定されます。最適な結果を得るには、キャップの幅は奇数にする必要があります。

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。これらは小数点以下の数値とすることができます。非整数の時の太さは、エクスポートまたは印刷時に適切に生成されます。キャップの太さをゼロに設定すると、キャップは描画されません。

XY Error box チェックボックスを有効にすると、不確実性の領域を示すためにエラーバーではなくボックスが描画されます。エラー値の 1 つ以上が NaN であるポイントについてはボックスは描画されません。

右はエラーバー付きのグラフの簡単な例です。

一番上のトレースは +/- Wave モードをい、+ ウェーブのみを表示しました。エラーバーの方向を反転させるため、誤差ウェーブの最終値を負にしました。

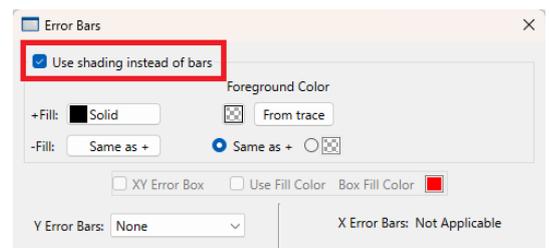
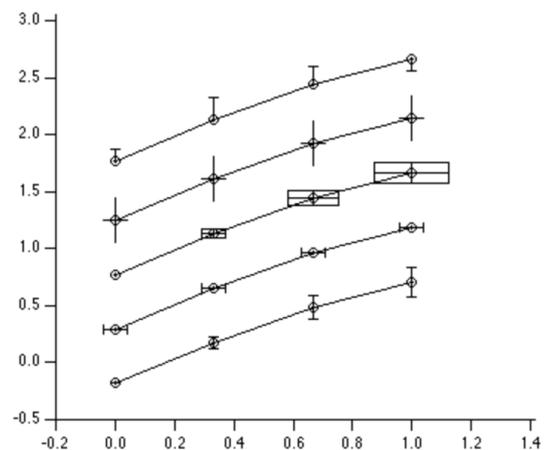
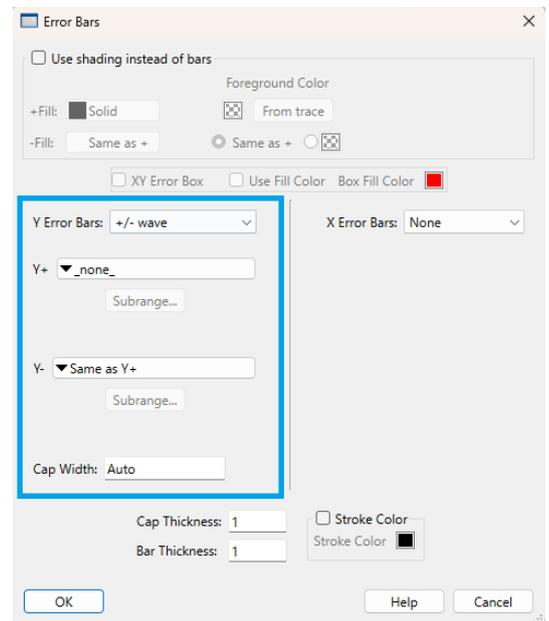
エラーシェーディング

Igor Pro 7 以降では、エラーバーの代わりにシェーディングを使用できます。

Error Bars サブダイアログで、Use shading instead of bars チェックボックスをオンにしてシェーディングを有効にします。

シェーディングモードは、+error レベルから -error レベルの間、または +error からデータ、データから -error までの間で塗りつぶします。

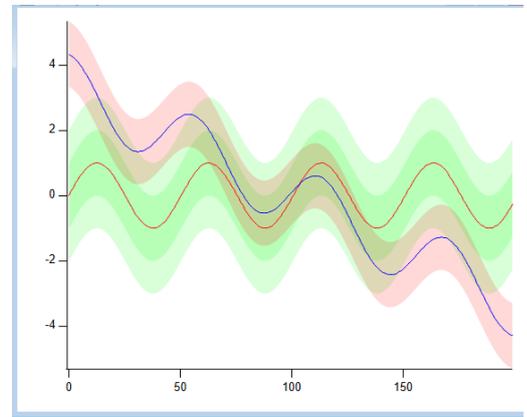
これは shade キーワードと共に使われるパラメータによって異なります。



以下のコマンドは、透明度を用いた多重かつ重なり合うエラーシェーディングを実例で示しています：

```
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
```

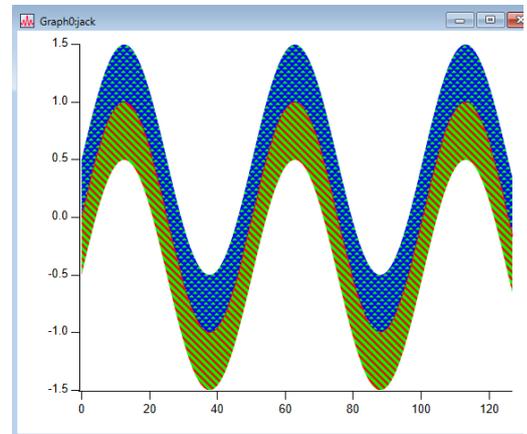
シェーディングは、旧式の GDI グラフィック技術では機能しません。詳細はヘルプ Graphics Technology を参照してください。



次のコマンドは、異なる +error および -error のシェーディングと、パターンの使用例を示しています：

```
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
```

詳細については、ErrorBars コマンドの shade キーワードを参照してください。



エラー楕円

Igor Pro 9.0 以降では、エラーバーやエラーボックスの代わりにエラー楕円を指定できます。

エラー楕円は X 方向と Y 方向の誤差を両方表示し、X と Y の相関関係も示します。

エラー楕円のデータは、各データポイントごとに 1 行を持つ 3 列のウェーブデータで提供する必要があります。

このウェーブは、エラーバーダイアログで Ellipse Wave とラベル付けされています。

ErrorBars コマンド内では ewave=ew で識別されます。

ew の各行は、1 つのデータポイントに対するエラー楕円の情報を含みます。

ew の列の解釈は、ErrorBars ダイアログで Data Type と表示され、ErrorBars コマンドの ELLIPSE キーワードの mode パラメータで識別されるモードに依存します。

mode = 0: ew には、X の標準偏差、Y の標準偏差、および X と Y の相関が含まれます。

mode = 1: ew には、X の分散、Y の分散、および X と Y の共分散が含まれます。

ErrorBars コマンド (ew) で指定する場合、各トレースのデータポイントに対して 3 列と 1 行からなる実質的な 2 次元ウェーブとなる限り、部分範囲指定を含めることができます。

「サブ範囲表示構文」のセクション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、その楕円が結び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、それは単なるトレースの色です。

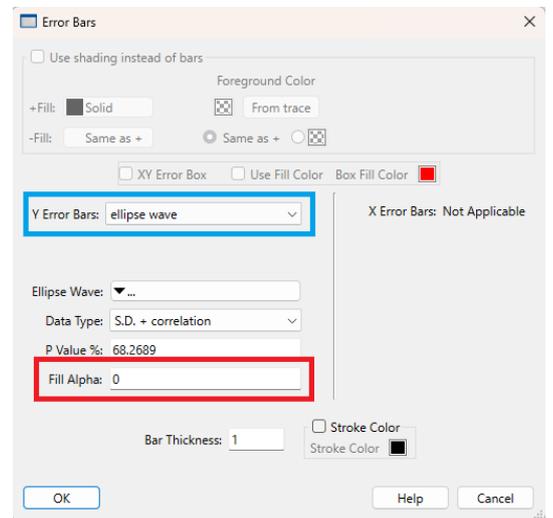
個々のデータポイントとその対応する楕円の色は、トレースカラーを f(z) として設定するか、ポイントのカスタマイズ（「ポイントのカスタマイズ」のセクションを参照）によって変更できます。

Error Bars ダイアログの Fill Alpha 設定を使うと、指定したトレースのすべてのエラー楕円の不透明度を 0（完全に透明）から 1.0（完全に不透明）の範囲で指定できます。

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

ErrorBars コマンドでは、ELLIPSE キーワードを使って、アルファ値を 0（完全透明）から 65535（完全不透明）までの値で指定します。

アルファに関する背景情報については、ヘルプ RGBA Values を参照してください。



エラー楕円の例

次の例はエラー楕円の使い方を示しています。

プロシージャウィンドウに次を入力して、コンパイルします。

```
Function ErrorEllipseDemo()
  Make/O/N=5 data = 5*sin(x)
  Display data
  ModifyGraph mode=3, marker=19 // マーカーモード、塗りつぶした円マーカー

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

  // 列 0: x における標準偏差
  ellipseData[0][0]={0.944527,1.19611,0.32858,1.64494,1.71742}

  // 列 1: y における標準偏差
  ellipseData[0][1]={0.357613,0.680198,0.665561,0.581878,0.948245}

  // 列 2: x と y の相関係数。範囲は[-1, 1]
  ellipseData[0][2]={0.562396,0.707809,0.396551,-0.516456,0.554322}

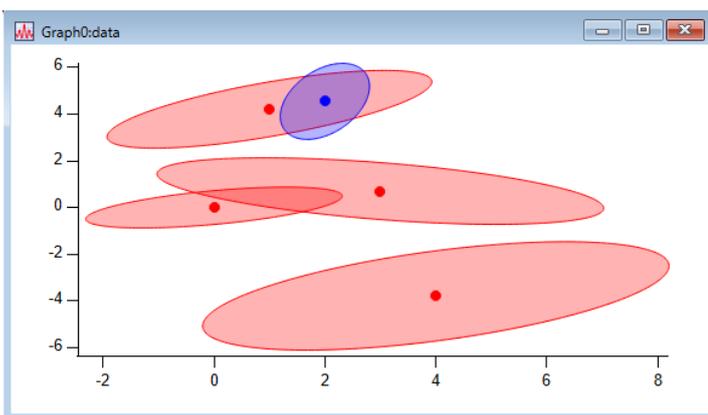
  ErrorBars data, ELLIPSE={0, .95, .3*65535}, ewave=ellipseData

  // ポイント2を青にする
  ModifyGraph rgb(data[2])=(0,0,65535)
End
```

End

コマンドラインで次を実行します。

```
ErrorEllipseDemo()
```

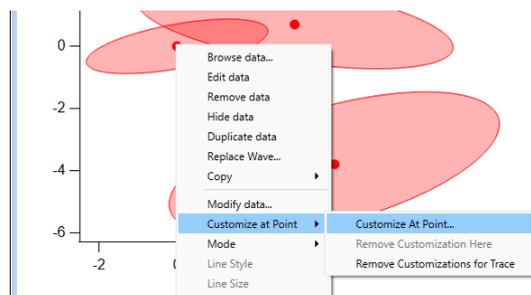


ポイントでのカスタマイズ

グラフ上で表示されるトレース上の個々のポイントの外観を、棒、マーカー、点、線からゼロモードまでカスタマイズできます。

これを対話的に行うには、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。

トレースリストにエントリが表示され、そのポイント番号が角括弧で囲まれて表示された状態で、Modify Trace ダイアログが表示されます。



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

ポイント上で右クリックし、Customize at Point → Remove Customization Here を選択すると、そのポイントのカスタマイズを削除できます。

トレースからすべてのカスタマイズを削除するには、トレースを右クリックし、Customize at Point → Remove Customizations from Trace を選択します。