

CONTENTS

ビジュアルヘルプ - 画像プロット.....	3
偽色画像.....	3
インデックスカラー画像.....	3
ダイレクトカラー画像.....	3
画像の読み込み.....	4
画像プロットの作成.....	4
X、Y、Zのウェーブリスト.....	5
画像プロットの修正.....	5
Modify Image Appearance ダイアログ.....	5
画像の X 座標と Y 座標.....	6
画像の X 座標と Y 座標 - 等間隔の場合.....	7
画像の X 座標と Y 座標 - 間隔が不均一な場合.....	7
1D の X と Y の中心データを持つ 2D Z ウェーブのプロット.....	8
グリッド化された XY データを持つ 1D X、Y、Z ウェーブのプロット.....	9
グリッド化されていない XY データを持つ 1D X、Y、Z ウェーブのプロット.....	10
画像の向き.....	11
画像の長方形のアスペクト比.....	11
画像の極性.....	12
画像カラーテーブル.....	12
画像のカラーテーブルの範囲.....	13
例：背景画像の上にデータを重ねる.....	14
カラーテーブルの範囲 - ルックアップテーブル（ガンマ）.....	15
例：ルックアップを使った高度な色・コントラスト効果.....	15
専用カラーテーブル.....	16
カラーテーブルの詳細.....	17
Igor Pro 4 互換のカラーテーブル.....	17
Igor Pro 5 互換のカラーテーブル.....	17
Igor Pro 5 グラデーションカラーテーブル.....	18
Igor Pro 5 の特殊用途のカラーテーブル.....	18
Igor Pro 6 互換のカラーテーブル.....	19
Igor Pro 6.2 互換のカラーテーブル.....	20

Igor Pro 9 互換のカラーテーブル.....	20
Igor Pro 10 互換のカラーテーブル.....	20
カラーテーブルウェーブ.....	21
インデックスカラーの詳細.....	21
線形インデックスカラー.....	22
対数インデックスカラー.....	22
例：ポイントスケール付きカラーインデックスウェーブ.....	22
ダイレクトカラーの詳細.....	23
ダイレクトカラーのパッキングモード.....	24
色の凡例の作成.....	25
画像インスタンス名.....	26
画像プロットのプレファレンス.....	27
Image Appearance プレファレンス.....	27
Image Axis プレファレンス.....	28
画像プレファレンスの使い方.....	28
画像プロットのショートカット.....	28

ビジュアルヘルプ - 画像プロット

画像データをグラフウィンドウに画像プロットとして表示することができます。

画像データとしては、2D ウェーブ、3D または 4D ウェーブのレイヤー、RGB 値を含む 3 つのレイヤーのセット、あるいは不透明度を表す「アルファ」値 (A) を含む RGBA 値の 4 つのレイヤーのセットが指定可能です。

画像プロットについて説明する時、画像データの構成要素を指すには「ピクセル」という用語を使い、画像プロットにおけるデータ要素の表現を指すには「長方形」という用語を使います。

各画像データの値は、画像プロット上の長方形の色を定義します。

長方形のサイズと位置は、グラフ軸の範囲、グラフの幅と高さ、およびピクセルの端の X 座標と Y 座標によって決まります。

画像データが浮動小数点型の場合、欠損データを NaN で表現することができます。

これにより、グラフの背景色が透けて見えるようになります。

画像は、ProgBack と UserBack 描画レイヤーと背景色を除き、グラフ内の他のすべてのオブジェクトの後ろに表示されます。

画像プロットは、偽色、インデックスカラー、またはダイレクトカラーのいずれかになります。

偽色画像

偽色画像では、2D ウェーブまたは 3D、4D ウェーブの層からのデータ値が、カラーテーブルを使って色にマッピングされます。

これは画像データを表示する上で非常に有効な手法で、多くの場合、サーフェスプロットやコンタープロットよりも効果的です。

同じデータの偽色画像の上に、コンタープロットを重ねて表示することができます。

ヘルプ Color Tables で説明されているように、Igor には多くの組み込みカラーテーブルが用意されています。

また、ヘルプ Color Table Waves で説明されているように、ウェーブを使って独自のカラーテーブルを定義することもできます。

さらに、ヘルプ Indexed Color Details で説明されているように、カスタムカラーテーブルを定義するカラーインデックスウェーブを作成することもできます。

インデックスカラー画像

インデックスカラー画像は、2D ウェーブまたは 3D、4D ウェーブのレイヤーに格納されたデータ値を、ユーザーが指定する RGB または RGBA のカラー値ウェーブへのインデックスとして使います。

ビデオカメラやスキャナーから得られるような True Color 画像は、一般的にインデックスカラーを使っています。

インデックスカラー画像は、ダイレクトカラーよりもメモリ消費量が少ないため、より一般的です。

「インデックスカラーの詳細」のセクションを参照してください。

ダイレクトカラー画像

ダイレクトカラー画像は、3D RGB または RGBA ウェーブを使います。

ウェーブの各レイヤーは、赤、緑、青、またはアルファの各色成分を表します。

特定の行と列に対する一連の色成分の値が、対応する画像の長方形の色を指定します。

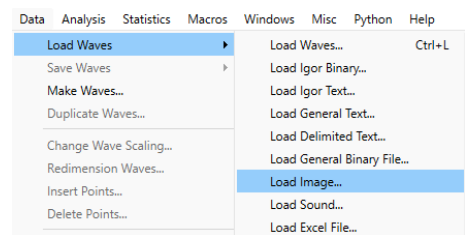
ダイレクトカラーを使うと、すべての長方形に固有の色を設定できます。
「ダイレクトカラーの詳細」のセクションを参照してください。

画像の読み込み

Data メニューの Load Waves→Load Image または ImageLoad コマンドを使って、TIFF、JPEG、PNG、BMP、Sun Raster 形式の画像ファイルを行列ウェーブに読み込むことができます。

また、プレーンテキストファイル、HDF5 ファイル、GIS ファイル、カメラハードウェアから画像を読み込むこともできます。

詳細はヘルプ Loading Image Files を参照してください。



画像プロットの作成

画像プロットは、通常のグラフウィンドウに表示されます。
グラフのすべての機能（軸、線のスタイル、描画ツール、コントロールなど）は、画像プロットにも適用されます。
ヘルプ Graphs を参照してください。

Windows→New→Image Plot を選択すると、新しいグラフウィンドウに画像プロットを作成できます。

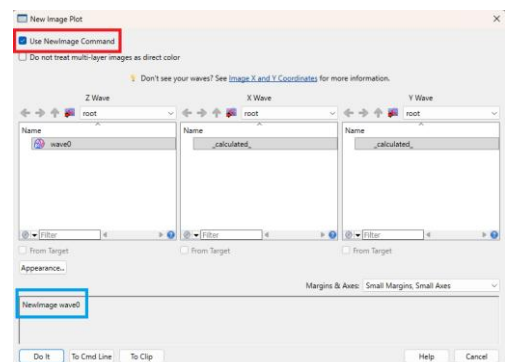
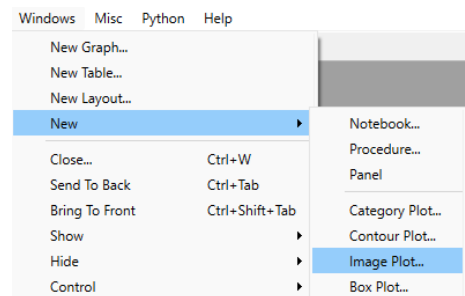
これにより、New Image Plot ダイアログが表示されます。
このダイアログでは、プロットが追加される空白のグラフが作成されます。

通常、このダイアログは2つのコマンドを生成します。
1つは空白のグラフウィンドウを作成する Display コマンド、もう1つは画像プロットをそのグラフウィンドウに追加する AppendImage コマンドです。

これにより、他のグラフと同様にグラフが作成されますが、多くの場合、NewImage コマンドを使ったほうが便利です。

Use NewImage Command チェックボックスをオンにすると、Display と AppendImage コマンド組み合わせが NewImage に置き換えられます。

NewImage は、画像のピクセル数に合わせてグラフウィンドウのサイズを自動的に調整し、画像が正しい向きで表示されるように縦軸を反転させます。



画像を含むグラフにコンター図を追加することで、画像の値が一定である領域を示すことができます。

Igor は、画像プロットの上にコンター図を描画します。

グラフ内で等高線図と画像を組み合わせる例については、ヘルプ Creating a Contour Plot を参照してください。

画像は通常、トレースや軸の下に描画されます。

通常、画像プロットはプロット領域全体を覆うため、トレースの下に描画することでトレースが見えるようになります。

画像プロットを軸やトレースの上に描画するように設定することもできます。

これは通常、特殊な効果を得るためにのみ必要となります。

画像の範囲がプロット領域を完全に覆っていない場合、または画像内に透明な領域がある場合にのみ有効です。

X、Y、Zのウェーブリスト

Z ウェーブは、画像データを含み、画像プロット内の各長方形の色を決定するウェーブです。

必要に応じて、X 方向の長方形の辺を定義する X ウェーブと、Y 方向の長方形の辺を定義する Y ウェーブを指定することができます。

これにより、幅や高さが異なる長方形を含む画像プロットを作成することができます。

Z ウェーブを選択すると、Igor は X ウェーブと Y ウェーブのリストを更新し、選択した Z ウェーブとの併用が可能なウェーブ（該当する場合）のみを表示します。

X ウェーブと Y ウェーブのリストには、適切な長さのウェーブのみが表示されます。

詳細については、「画像の X 座標と Y 座標」のセクションを参照してください。

X ウェーブリストから「_calculated_」を選択すると、Z ウェーブの行スケーリング（X スケーリング）を使って、画像の長方形の中心点の X 座標が算出されます。

Y ウェーブリストから「_calculated_」を選択すると、Z ウェーブの列スケーリング（Y スケーリング）が使われ、画像の長方形の中心点の Y 座標が算出されます。

参照

ヘルプ New Image Plot Dialog

AppendImage、ModifyImage、RemoveImage コマンド

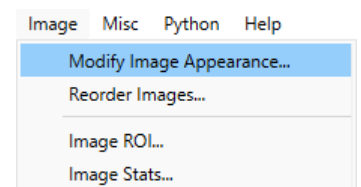
画像プロットの修正

Image→Modify Image Appearance を選択すると、画像プロットの外観を変更できます。

これにより、Modify Image Appearance ダイアログが表示されます。

このダイアログは、New Image Plot ダイアログのサブダイアログとしても利用可能です。

注記： プレファレンスを使って画像のデフォルトの表示を変更すれば、同じ変更を何度も繰り返す必要がなくなります。「画像プロットのプレファレンス」のセクションを参照してください。



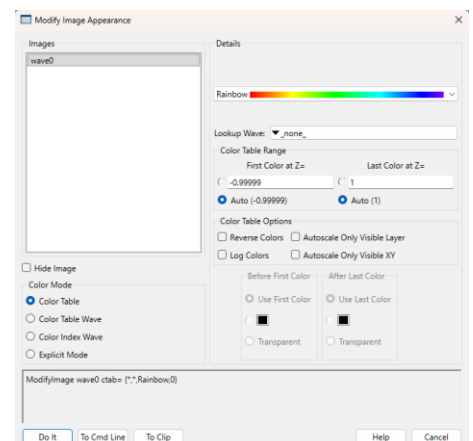
Modify Image Appearance ダイアログ

Modify Image Appearance ダイアログは、偽色画像およびインデックスカラー画像に適用されますが、ダイレクトカラー画像には適用されません。

「ダイレクトカラーの詳細」のセクションを参照してください。

インデックスカラーを使うには、Color Index Wave ラジオボタンをクリックし、カラーインデックスウェーブを選択します。

カラーインデックスウェーブの詳細については、「インデックスカラーの詳細」のセクションを参照してください。



偽色を使うには、Color Table ラジオボタンをクリックして組み込みのカラーテーブルを選択するか、Color Table Wave ラジオボタンをクリックしてカラーテーブルウェーブを選択します。

自動スケーリングによるカラーマッピングでは、カラーテーブルの最初の色が画像データの最小値に、最後の色が最大値に割り当てられます。

このダイアログでは、画像ウェーブの値を「Z」で表します。

詳細については、ヘルプ Color Tables を参照してください。

インデックスカラーとカラーテーブルの色は、Log Colors チェックボックスで設定される ModifyImage の log パラメーターに基づき、最小 Z 値と最大 Z 値の間で線形または対数的に分布します。

Explicit Mode を使うと、画像内の特定の Z 値に対して特定の色を選択できます。

画像の要素がダイアログに入力された数値と完全に一致する場合、その要素は割り当てられた色で表示されます。

これは浮動小数点データで作成された画像にはあまり有用ではなく、整数データ向けに設計されています。

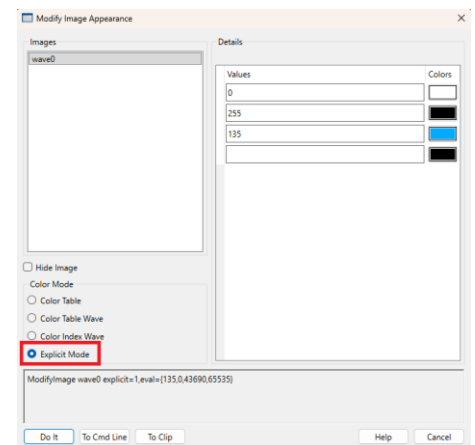
浮動小数点データに対して完全に一致する数値を入力することは、ほぼ不可能です。

初めて Explicit Mode を選択すると、白を 0、黒を 255 とする 2 つのエントリが自動的に作成されます。

新しい値を入力するための 3 行目の空白行が追加されます。

その空白行に何かを入力すると、さらに別の空白行が追加されます。

項目を削除するには、リスト内の行の空白部分をクリックして選択し、Backspace キーを押してください。



画像の X 座標と Y 座標

画像では、ウェーブデータ要素が長方形として表示されます。

これらは、XY プロットと同様に、座標軸に対して表示されます。

各画像長方形の輝度や色は、行列（2D）ウェーブの対応するデータ要素、3D または 4D ウェーブのレイヤー、または 3D RGB または RGBA ウェーブのレイヤー群によってコントロールされます。

画像プロットについて説明する時、画像データの構成要素を指すには「ピクセル」という用語を用い、画像プロットにおけるデータ要素の表現を指すには「長方形」という用語を用います。

空間次元である X と Y のそれぞれについて、各画像長方形の辺は、以下のいずれかによって定義されます。

- ・ 画像データを含むウェーブの次元スケーリング
- ・ 1D 補助 X ウェーブまたは Y ウェーブ

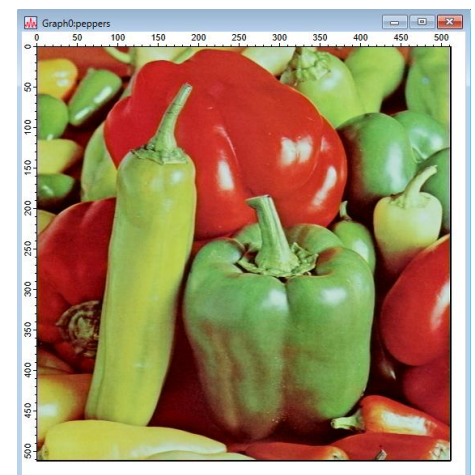
最も単純なケースでは、すべてのピクセルの幅と高さが同じであるため、ピクセルはすべて同じ大きさの正方形になります。

もう 1 つの一般的なケースは、正方形ではないものの、すべてのピクセルの幅と高さが同じである長方形のピクセルで構成されるものです。

これらはいずれも、等間隔に配置されたデータの一例です。

これらのケースでは、次元（X と Y）のスケーリングを使って長方形の中心を指定します。

これについては、「画像の X 座標と Y 座標 - 等間隔の場合」のセクションでさらに詳しく説明します。



まれに、幅や高さが不均一なピクセルが含まれる場合があります。
その場合は、画像の長方形の境界を指定する補助的な X ウェーブまたは Y ウェーブを指定する必要があります。
これについては、「画像の X 座標と Y 座標 - 間隔が不均一な場合」のセクションでさらに詳しく説明しています。

これらのケースを組み合わせたことも可能です。

例えば、ピクセルの幅は均一でも、高さが不均一な場合があります。

その場合は、一方の次元にはあるテクニックを、もう一方の次元には別のテクニックを使います。

ピクセルの幅や高さが明確に定義されていないため、厳密には画像データとは言えないものの、行列 (2D) のウェーブとして保存されているデータがある場合があります。

このようなデータは散布図に適している場合が多いですが、画像としてプロットすることも可能です。

これについては、「1D の X と Y の中心データを持つ 2D Z ウェーブのプロット」のセクションでさらに詳しく説明します。

その他のケースでは、1D X、Y、Z ウェーブが存在する場合があります。

これらのケースについては、「グリッド化された XY データを持つ 1D X、Y、Z ウェーブのプロット」と「グリッド化されていない XY データを持つ 1D X、Y、Z ウェーブのプロット」で説明しています。

以降のセクションにはコマンドの例が記載されています。

ヘルプファイルを開いている場合は、必要に応じて、コマンドを選択し、Ctrl+Enter キーを押してそのコマンドを実行することができます。

画像の X 座標と Y 座標 - 等間隔の場合

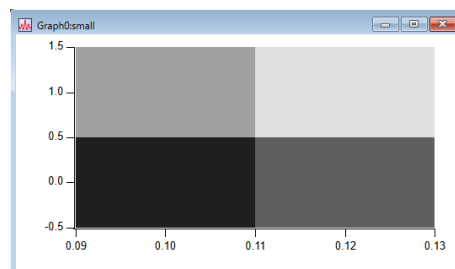
データが等間隔のピクセルで構成されている場合は、画像ウェーブの次元スケーリングを使って、画像の長方形座標を指定します。

スケーリングの設定は、Change Wave Scaling ダイアログ (Data メニュー) または SetScale コマンドを使って行うことができます。

特定のピクセルのスケーリング後の次元の値は、対応する画像長方形の中心を指定します。

ここでは、その効果を強調するために 2×2 の行列を使った例を示します。

```
Make/O small={{0,1},{2,3}}  
// x 次元のスケーリングを設定  
SetScale/P x 0.1,0.02,"", small  
// y 次元のスケーリングを設定  
SetScale/P y 0.0,1.0,"", small  
Display;AppendImage small  
ModifyImage small ctab= {-0.5,3.5,Grays}
```



X 軸上の長方形は、行列ウェーブの X スケーリングによって定義される X (行) インデックスである 0.10 と 0.12 を中心に配置されています。

Y 軸上の長方形は、行列ウェーブの Y スケーリングによって定義される Y (列) インデックスである 0.0 と 1.0 を中心に配置されています。

いずれの場合も、長方形のエッジは、対応するインデックス値から 0.5 ピクセル分離されています。

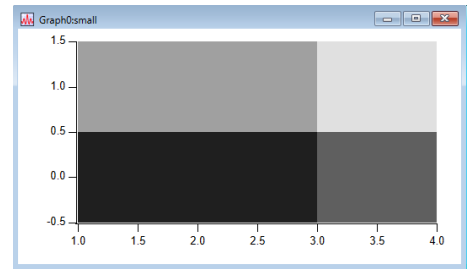
画像の X 座標と Y 座標 - 間隔が不均一な場合

ピクセルデータが X 方向または Y 方向に不均一な間隔で配置されている場合は、画像の長方形のエッジの座標を定義するために、X または Y のウェーブを指定する必要があります。

警告： 各長方形のエッジを定義するためには、これらのウェーブには、画像ウェーブの X (行) または Y (列) の次元よりも1つ多いデータポイントが含まれている必要があります。

この例では、行列ウェーブは Y 方向では等間隔ですが、X 方向では不均等な間隔になっています。

```
Make/O small={{0,1},{2,3}}
// Y 次元のスケーリングを設定
SetScale/P y 0.0,1.0,"", small
// smallx を持つ X の端を定義
Make/O smallx={1,3,4}
Display;AppendImage small vs {smallx,*}
ModifyImage small ctab= {-0.5,3.5,Grays}
```



X 座標のウェーブ (smallx) は、各画像長方形の垂直方向のエッジをコントロールします。

smallx は3つのデータポイントで構成されており、これらは画像プロット内の2つの長方形の垂直方向のエッジを定義するために必要です。

smallx の値は次のように解釈されます。

Point 0: 1.0 最初の長方形の左端を設定します

Point 1: 3.0 最初の長方形の右端と、2番目の長方形の左端を設定します

Point 2: 4.0 最後の長方形の右端を設定します

1D のエッジウェーブは、厳密に増加するか、あるいは厳密に減少するかのいずれかでなければなりません。

エッジを指定する X ウェーブや Y ウェーブがあり、それらに追加のポイントが含まれていない場合は、単に追加のポイントを追加するだけで処理を進めることができる場合があります。

これは、テーブル内のウェーブを編集するか、InsertPoints コマンドを使うことで実行できます。

これが適切でない場合は、次のセクションで別の方法について説明していますので、そちらを参照してください。

1D の X と Y の中心データを持つ 2D Z ウェーブのプロット

画像において、各ピクセルには明確に定義された幅と高さがあります。

データが特定の X 座標と Y 座標でサンプリングされており、ピクセルの幅と高さが明確に定義されていない場合、あるいは各ピクセルの幅と高さがわからない場合は、厳密には適切な画像とは言えません。

しかし、この種のデータは X ウェーブや Y ウェーブと関連付けられた行列ウェーブとして保存されることが多いため、X ウェーブと Y ウェーブをピクセルの中心座標を含むものとみなして、画像として表示すると便利な場合があります。

これを行うには、画像の長方形の境界を指定するために、新しい X ウェーブと Y ウェーブを作成する必要があります。

新しい X ウェーブは、行列ウェーブの行数より1つ多いポイント数を持つ必要があります、新しい Y ウェーブは、行列ウェーブの列数より1つ多いポイント数を持つ必要があります。

画像の長方形の中心点の集合だけでは、その長方形の辺を一意に決定することはできません。

これを理解するために、(0,0) を中心とする 1×1 の画像を考えてみてください。

その辺はどこにあるでしょうか？

どこにでもあり得ます。

追加の情報がない場合、最善の策は、次の関数で行っているように、妥当なエッジのセットを生成することです。

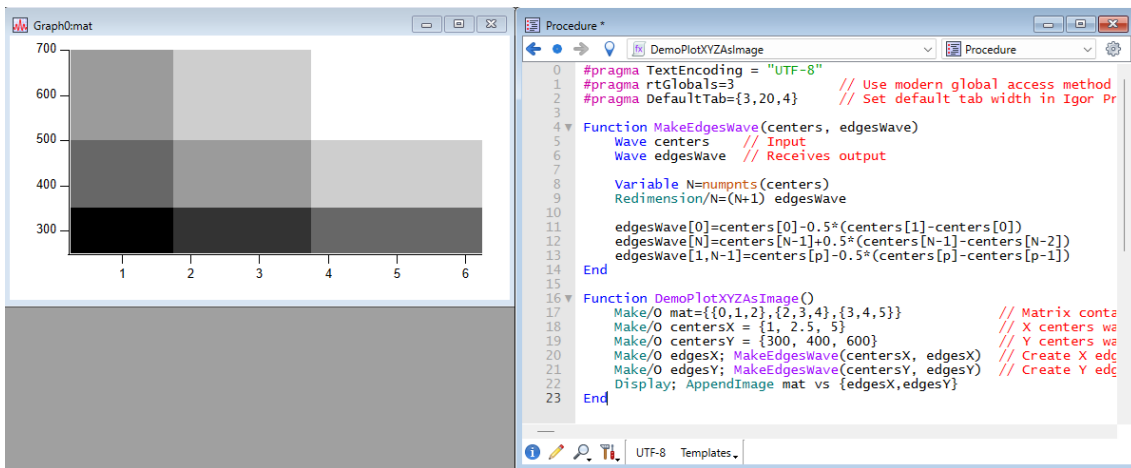
```
Function MakeEdgesWave(centers, edgesWave)
    Wave centers          // 入力
    Wave edgesWave      // 出力を受ける

    Variable N=numpts(centers)
    Redimension/N=(N+1) edgesWave

    edgesWave[0]=centers[0]-0.5*(centers[1]-centers[0])
    edgesWave[N]=centers[N-1]+0.5*(centers[N-1]-centers[N-2])
    edgesWave[1,N-1]=centers[p]-0.5*(centers[p]-centers[p-1])
End
```

次の関数は、MakeEdgesWave の使い方を示しています。

```
Function DemoPlotXYZAsImage()
    Make/O mat={{0,1,2},{2,3,4},{3,4,5}} // Matrix containing Z values
    Make/O centersX = {1, 2.5, 5} // X centers wave
    Make/O centersY = {300, 400, 600} // Y centers wave
    Make/O edgesX; MakeEdgesWave(centersX, edgesX) // Create X edges wave
    Make/O edgesY; MakeEdgesWave(centersY, edgesY) // Create Y edges wave
    Display; AppendImage mat vs {edgesX,edgesY}
End
```



エッジウェーブを作成できる追加情報がある場合は、それを使って作成してください。

グリッド化された XY データを持つ 1D X、Y、Z ウェーブのプロット

この場合、XYZ 空間内のポイント群を定義する、同じ長さの 1D X、Y、Z ウェーブが存在します。X ウェーブと Y ウェーブは等間隔のサンプリンググリッドを構成しますが、X 方向の間隔は Y 方向の間隔とは異なる場合があります。

このようなデータを表示する良い方法は、Z データの値に応じて色を変化させた散布図を作成することです。ヘルプ [Setting Trace Properties from an Auxiliary \(Z\) Wave](#) を参照してください。

「1D X と Y 中心データを持つ 2D Z ウェーブのプロット」で説明しているように、データを画像としてプロットできるように変換することも可能です。

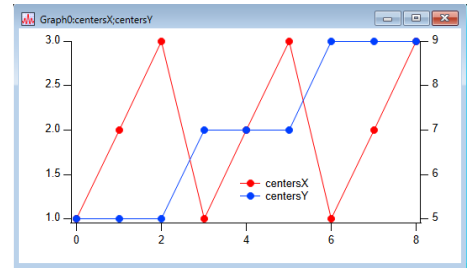
これを行うには、1D の Z ウェーブを 2D の行列ウェーブに変換し、さらに X ウェーブと Y ウェーブを、ピクセルの水平および垂直の中心座標を含むように変換する必要があります。

例えば、次のような X、Y、Z のデータから始めます。

```
Make/O centersX = {1,2,3,1,2,3,1,2,3}
Make/O centersY = {5,5,5,7,7,7,9,9,9}
Make/O zData = {1,2,3,4,5,6,7,8,9}
```

X と Y のデータをグラフにプロットすると、X ウェーブと Y ウェーブが周期的なパターンを示していることがわかります。

```
Display/K=0 root:centersX
ModifyGraph mode=4,marker=19
AppendToGraph/R centersY
ModifyGraph mode=4,marker=19,rgb(centersY)=(1,16019,65535)
Legend/C/N=text0/F=0/A=MC
```



これを画像として表示するために、Z ウェーブをピクセル値を表す 2D 行列に変換し、X ウェーブと Y ウェーブがピクセルの行と列の中心を表すようにデータを変換します。

```
Redimension/N=(3,3) zData
Make/O/N=3 xCenterLocs = centersX[p] // 1, 2, 3
Make/O/N=3 yCenterLocs = centersY[p*3] // 5, 7, 9
```

Point	xCenterLocs	yCenterLocs
0	1	5
1	2	7
2	3	9
3		

これで、「1D X と Y の中心データを持つ 2D Z ウェーブのプロット」の項で説明したデータが揃いました。

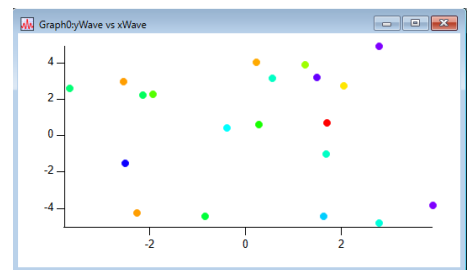
グリッド化されていない XY データを持つ 1D X、Y、Z ウェーブのプロット

この場合、XYZ 空間内のポイント群を定義する、長さが等しい 1D ウェーブ (X、Y、Z ウェーブ) が存在します。X ウェーブと Y ウェーブは格子を形成していないため、前のセクションの手法は適用できません。

2D 散布図は、このようなデータを視覚的に表現するのに適した方法です。

// ランダムなポイント

```
Make/O/N=20 xWave=noise(4),yWave=noise(5),zWave=noise(6)
Display yWave vs xWave
ModifyGraph mode=3,marker=19
ModifyGraph zColor(yWave)={zWave,*,*,Rainbow,0}
```



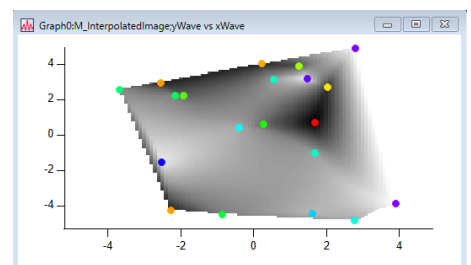
このデータは通常の画像ではありませんが、散布図ではなく画像として表示したい場合もあるでしょう。

ImageFromXYZ コマンドを使うと、XYZ データに対応する行列ウェーブを作成できます。

このマトリクスウェーブは、単純な画像プロットとしてプロットすることができます。

また、Voronoi 補間を使って、XYZ データからマトリクスウェーブを作成することもできます。

```
Concatenate/O {xWave,yWave,zWave}, tripletWave
ImageInterpolate/S={-5,0.1,5,-5,0.1,5} voronoi tripletWave
AppendImage M_InterpolatedImage
```



Voronoi 補間のアルゴリズムは計算負荷が高いため、非常に大きなウェーブに対しては実用的ではない可能性があることに注意してください。

無順序な散布データから滑らかな表面を生成する代替手法として、Loess や ImageInterpolate クリギングも参照してください。

この種のデータを 3D サーフェスとして表示するためのその他のオプションについては、ヘルプ 3D Scatter Plot のセクションと、https://www.youtube.com/watch?v=kggo0B43n_c にあるビデオチュートリアル Creating a Surface Plot from Scatter Data で説明しています。

画像の向き

デフォルトでは、Y 値（行列の列インデックス）の増加方向を上方向、X 値（行列の行インデックス）の増加方向を右方向として描画します。

ほとんどの画像形式では、Y 値は下方向に向かって増加することが想定されています。その結果、次のようにして画像プロットを作成すると

```
Display; AppendImage <image wave>
```

図が上下逆さまに表示されます。

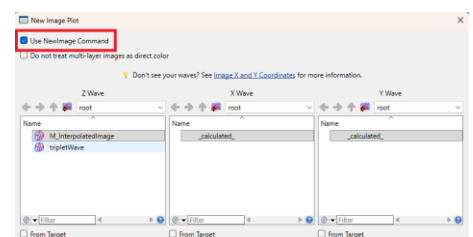
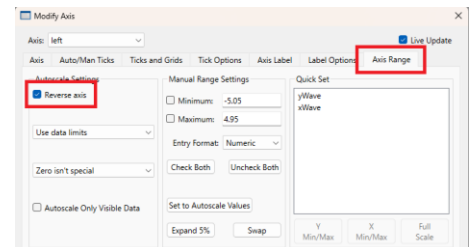
Graph メニューから開くことができる Modify Axis ダイアログの Axis Range ダイアログを使うと、Y 軸を反転させて画像を垂直方向に反転させたり、X 軸を反転させて水平方向に反転させたりすることができます。

また、画像のウェーブの Y 軸のスケーリングを反転させることで、画像を垂直方向に反転させることもできます。

より簡単な方法として、AppendImage の代わりに NewImage を使う方法があります。

これは、New Image Plot ダイアログで Use NewImage command チェックボックスをオンにすることで実行できます。

NewImage は、左軸を自動的に反転させます。

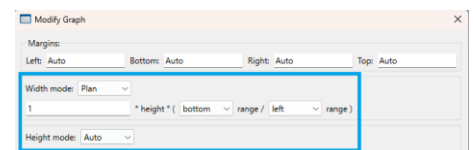


画像の長方形のアスペクト比

デフォルトでは、画像の長方形を正方形にはしません。

これを修正するには、Graph メニューにある Modify Graph ダイアログを開き、グラフの幅モードとして Plan を選択してください。

Plan の高さモードを使っても、同様の結果が得られます。



DimDelta(imageWave,0) が DimDelta(imageWave,1) と等しくない場合、これら2つの値の比率（または逆数）を「Plan の幅または高さ」に入力する必要があります。

```
SetScale/P x 0,3,"", mat2dImage
SetScale/P y 0,1,"", mat2dImage
ModifyGraph width=0, height={Plan,3,left,bottom}
// または
ModifyGraph height=0, width={Plan,1/3,bottom,left}
```

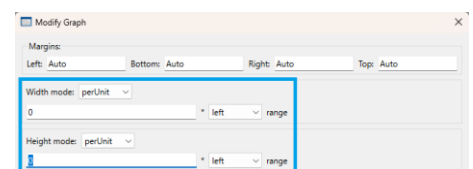
Aspect の幅または高さのモードは使わないでください。

これらのモードを使うと、本来正方形ではない画像でも、全体が正方形になって表示されてしまいます。

Plan モードでは、画像の長方形が正方形になるように設定されますが、サイズは任意に指定できます。

各画像の長方形の幅と高さを 1 ピクセル単位にしたい場合は、perUnit width と per Unit height モードを使ってください。

Swap XY チェックボックスをオンにすると、画像を対角線に沿って反転させることもできます。



画像の極性

画像のピクセル値が反転している場合もあります。

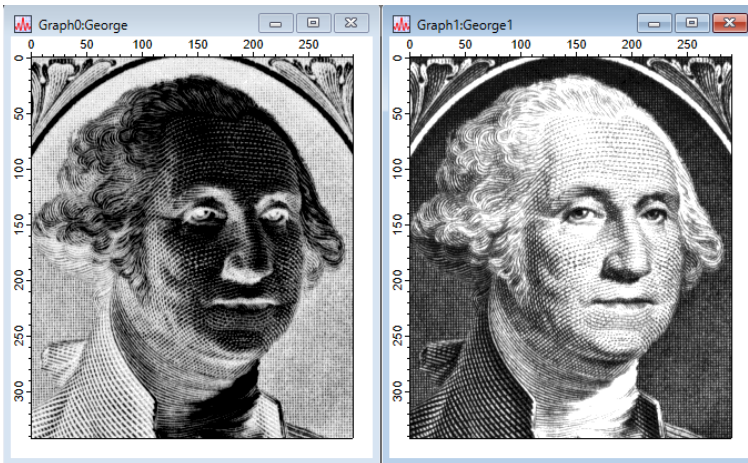
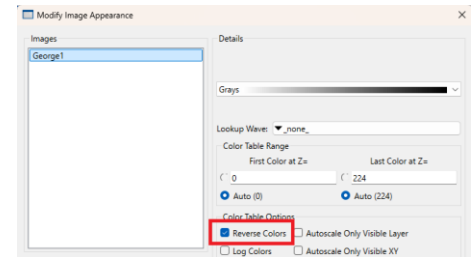
偽色画像は、カラーテーブルを反転させることで反転させることができます。

Modify Image Appearance ダイアログで、Reverse Colors チェックボックスを選択してください。

ヘルプ Color Tables を参照してください。

インデックスカラープロットの色を反転させるのはより困難です。

カラーウェーブの行を反転させる必要があります。



画像カラーテーブル

偽色プロットでは、通常、2D ウェーブのデータ値は、閲覧者がデータ値を容易に識別できる一連の色を含むカラーテーブルに線形に変換されます。

データ値は、ModifyImage log=1 オプションを使うことで対数変換することができ、データ値が数桁にわたる場合に役立ちます。

偽色画像に使用できる組み込みのカラーテーブルは多数用意されています。

また、ウェーブを使って独自のカラーテーブルを作成することも可能です。

詳細は「カラーテーブルウェーブ」のセクションを参照してください。

CTabList 関数は、すべての組み込みカラーテーブル名のリストを返します。

ColorTab2Wave 関数を使うと、任意の組み込みカラーテーブルからカラーインデックスウェーブまたはカラーテーブルウェーブを作成できます。

```
Print CTabList()
```

```
Grays;Rainbow;YellowHot;BlueHot;BlueRedGreen;RedWhiteBlue;PlanetEarth;Terrain;Grays256;Rainbow256;YellowHot256;BlueHot256;BlueRedGreen256;RedWhiteBlue256;PlanetEarth256;Terrain256;Grays16;Rainbow16;Red;Green;Blue;Cyan;Magenta;Yellow;Copper;Gold;CyanMagenta;RedWhiteGreen;BlueBlackRed;Geo;Geo32;LandAndSea;LandAndSea8;Relief;Relief19;PastelsMap;PastelsMap20;Bathymetry9;BlackBody;Spectrum;SpectrumBlack;Cycles;Fiddle;Pastels;RainbowCycle;Rainbow4Cycles;BlueGreenOrange;BrownViolet;ColdWarm;Mocha;VioletOrangeYellow;SeaLandAndFire;GreenMagenta16;EOSOrangeBlue11;EOSSpectral11;dBZ14;dBZ21;Web216;Mud;Classification;Turbo;RedYellowGreen;RedYellowGreen10
```

デモ実験 ColorsMarkersLinesPatterns では、すべての組み込みカラーテーブルを紹介しています。

画像のカラーテーブルの範囲

テーブル内の色範囲に対応するデータ値の範囲は、Modify Image Appearance ダイアログを使って、手動または自動で設定できます。

最初の色または最後の色を自動スケーリングするように選択すると、画像のウェーブデータを解析し、検出された最小値または最大値を使います。

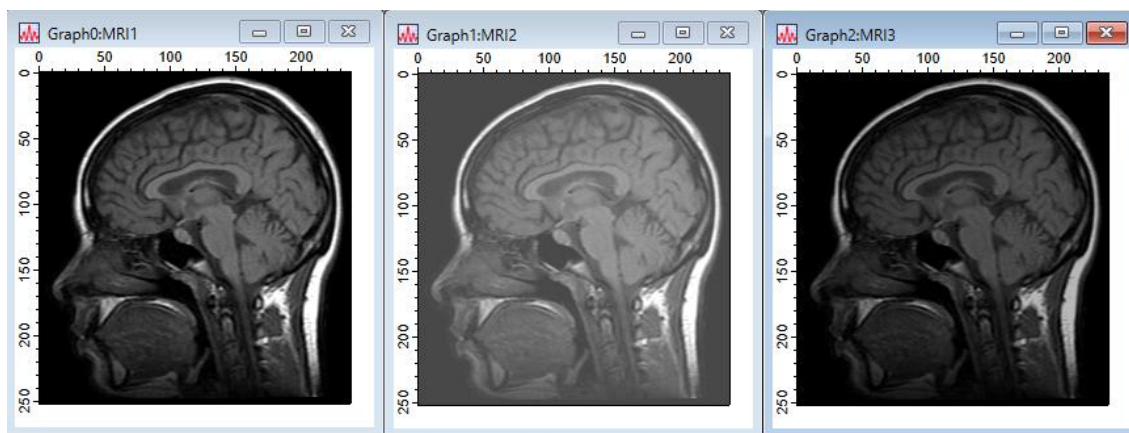
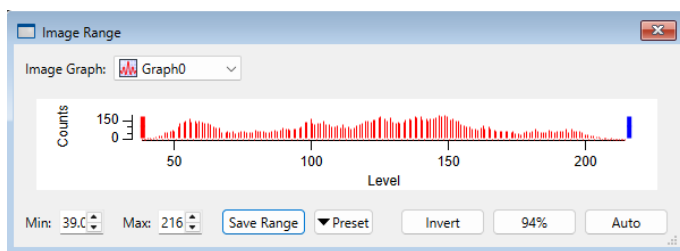
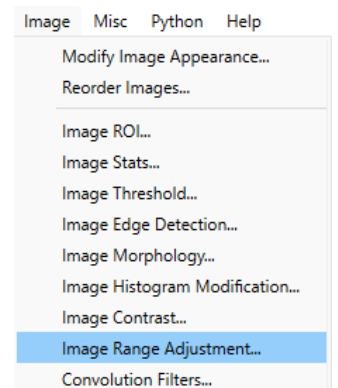
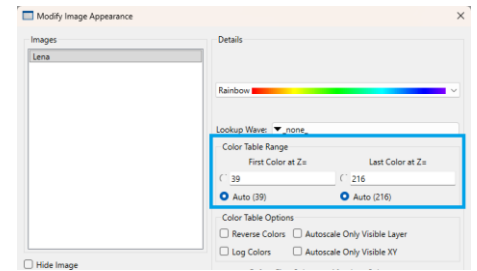
"First Color at Z=" と "Last Color at Z=" の値を変更することで、データの詳細を確認することができます。

例えば、Grays カラーテーブルを使う場合、最初の色（黒）を画像の最小値よりも小さい数値に割り当てることで、画像を明るくすることができます。

これにより、明るい色が画像の最小値に割り当てられます。

画像の最大値を暗くするには、最後の色を画像の最大値よりも大きい数値に割り当て、暗い色が画像の最大値に割り当てられるようにします。

Image→Image Range Adjustment を選択すると、これらの設定をその場で調整できます。



ctab={0,255,Grays}

ctab={-100,255,Grays}

ctab={0,355,Grays}

範囲の最大値を超えるデータ値には、カラーテーブルの最後の色が割り当てられます。

または、それらすべてを単一の色に割り当てるか、透明にすることもできます。

同様に、範囲の最小値を下回るデータ値には、カラーテーブルの最初の色が割り当てられます。

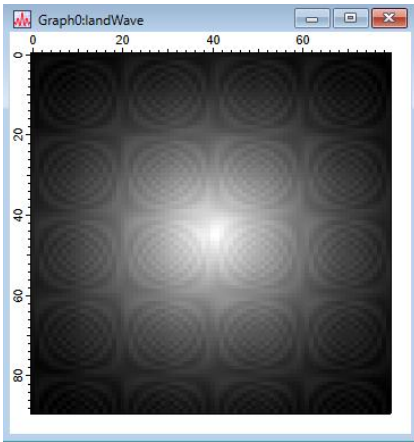
または、それらすべてを単一の色（最大値の色とは異なる場合もあります）に割り当てるか、透明にすることもできます。

例：背景画像の上にデータを重ねる

画像の範囲設定で、小さい値を透明に表示するようにすると、その部分の下の画像が見えるようになります。これにより、背景画像を基準にして、不透明な値がどこにあるかを視覚的に把握しやすくなります。以下は、架空の気象レーダーの例です。

まず、背景画像として使う「土地」を作成します。

```
Make/O/N=(80,90) landWave = 1-sqrt((x-40)*(x-40)+(y-45)*(y-45))/sqrt(40*40+45*45)
landWave = 7000*landWave*landWave
landWave += 200*sin((x-60)*(y-60)*pi/10)
landWave += 40*(sin((x-60)*pi/5)+sin((y-60)*pi/5))
NewImage landWave
```

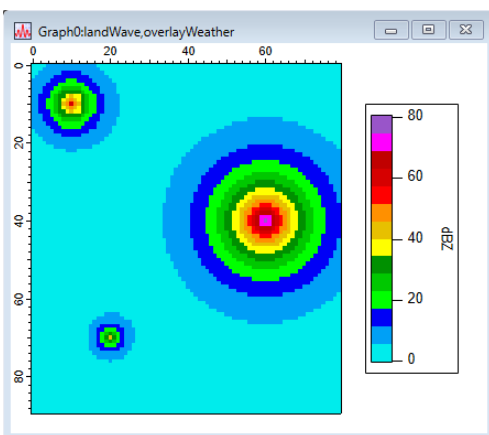


次に、約 0~80 dBZ の範囲の「気象」レーダーデータを作成します。

```
Duplicate/O landWave overlayWeather // "気象" レーダー値
overlayWeather=60*exp(-(sqrt((x-10)*(x-10)+(y-10)*(y-10))/5)) // 嵐1
overlayWeather+=80*exp(-(sqrt((x-60)*(x-60)+(y-40)*(y-40))/10)) // 嵐2
overlayWeather+=40*exp(-(sqrt((x-20)*(x-20)+(y-70)*(y-70))/3)) // 嵐3
SetScale d, 0, 0, "dBZ", overlayWeather
```

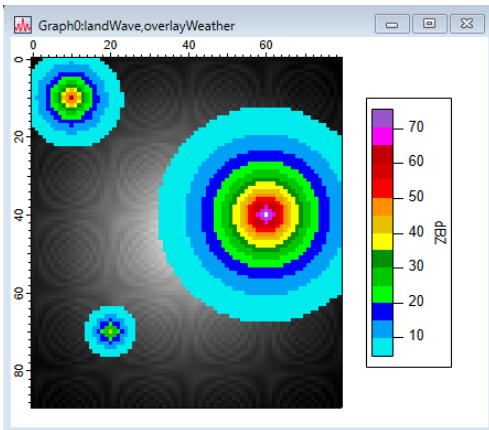
landWave と同じ軸を使って overlayWeather ウェーブを追加し、画像を重ね合わせます。デフォルトのカラーテーブルの範囲では、landWave は完全に隠れてしまいます。

```
AppendImage/T overlayWeather
ModifyImage overlayWeather ctab= {*,*,dBZ14,0}
// ColorScale を使って画像のデータ範囲を表示
ModifyGraph width={Plan,1,top,left}, margin(right)=100
ColorScale/N=text0/X=107.50/Y=0.00 image=overlayWeather
```



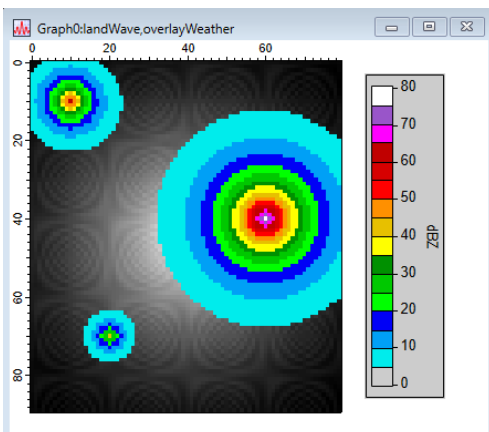
降水モードにおいて、データ値が 5 から 75 の範囲にある場合、dBZ14color テーブルを選択することで、画像プロットの色を米国気象局 (NWS) の基準値に合わせて調整します。
この際、5 未満の値は透明、75 を超える値は白となります。

```
ModifyImage overlayWeather ctab= {5,75,dBZ14,0};DelayUpdate  
ModifyImage overlayWeather minRGB=NaN,maxRGB=(65535,65535,65535)
```



カラースケールを変更し、カラーテーブルの値 (0~80) よりも広い範囲を表示するようにします。

```
ColorScale/C/N=text0 colorBoxesFrame=1,heightPct=90,nticks=10  
ColorScale/C/N=text0/B=(52428,52428,52428) axisRange={0,80},tickLen=3.00
```



カラーテーブルの範囲 - ルックアップテーブル (ガンマ)

通常、データ値の範囲と色の範囲は直線的に関連しています。

ただし、ModifyImage の log パラメーターが 1 に設定されている場合は、対数的に関連します。

また、次の例で説明するように、ルックアップ (または「ガンマ」) ウェーブを指定することで、マッピングを非線形にすることもできます。

例：ルックアップを使った高度な色・コントラスト効果

lookup パラメーターを指定した ModifyImage コマンドは、スケーリングされた Z 値から現在のカラーテーブルへのマッピングを変更する 1D ウェーブを指定します。

lookup ウェーブ内の値は 0.0 から 1.0 の範囲である必要があります。

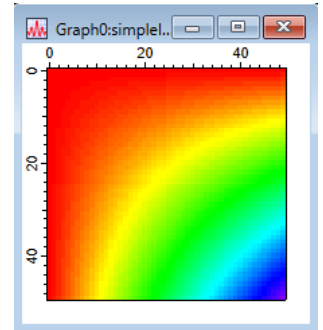
0 から 1 への線形ランプは効果をもたらしません、1 から 0 へのランプはカラーマップを反転させます。

グレースケール画像へのガンマ補正の適用や、特殊効果に使われます。

```

Make luWave=0.5*(1+sin(x/30))
Make /n=(50,50) simpleImage=x*y
NewImage simpleImage
ModifyImage simpleImage ctab= {*,*,Rainbow,0}

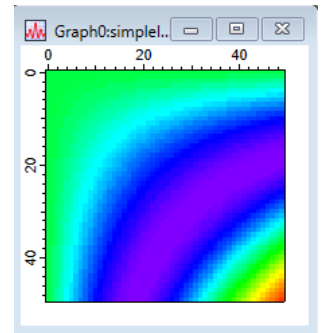
```



```

// 単純な画像を検査した後、ルックアップを適用：
ModifyImage simpleImage lookup=luWave

```



専用カラーテーブル

一部のカラーテーブルは、特定の用途や特定の数値範囲に合わせて設計されています。

BlackBody : 1,000~10,000 Kの温度範囲において、加熱された「黒体」の色（ただし、その黒体の明るさではない）を示すように設計されています。

Spectrum : 380~780 nmの範囲でナノメートル単位で測定された可視光の波長に対応する色を示すように設計されています。

SpectrumBlack : Spectrumと同様の機能を持ちますが、その範囲は355~830 nmです。黒へと徐々に変化させるのは、人間の目がこの範囲の両端では色を認識できなくなることを示すためのものです。

GreenMagenta16、EOSOrangeBlue11、EOSSpectral11、dBZ14、dBZ21 : 降水や風速のレーダー反射率などの気象関連画像における離散レベルや、地球物理学の応用分野における離散レベルを表現するために設計されています。

LandAndSea、Relief、PastelsMap、SeaLandAndFire : 海面を表すことを意図した急激な色の変化が含まれています。LandAndSea と Relief では、この変化が範囲の50%の位置で発生します。最小値を最大値の負の数に設定することで、この変化を0の位置に設定することができます。

```

ModifyImage imageName, ctab={-1000,1000,LandAndSea,0} // 画像プロット
ColorScale/C/N=scale0 ctab={-1000,1000,LandAndSea,0} // カラーテーブル

```

PastelsMap : LandAndSea等のトランジションが範囲の3分の2の位置に設定されています。最大値の負の2倍を最小値に設定することで、このトランジションを0の位置に配置できます。

```

ModifyImage imageName, ctab={-2000,1000,PastelsMap,0} // 画像プロット
ColorScale/C/N=scale0 ctab={-2000,1000,PastelsMap,0} // カラーテーブル

```

この原則を他のカラーテーブルにも応用することで、特定の色を任意の値に設定することができます。

BlackBody、Spectrum、SpectrumBlack の各カラーテーブルは、Color Science ウェブサイト (<http://www.physics.sfasu.edu/astro/color.html> : 現在は閉鎖) のアルゴリズムに基づいています。また、Wyszecki & Stiles による『Color Science』も参照してください。

参考文献

Light, Adam, and Patrick J. Bartlein, The End of the Rainbow? Color Schemes for Improved Data Graphics, Eos, 85, 385-391, 2004.

Wyszecki, Gunter, and W. S. Stiles, Color Science : Concepts and Methods, Quantitative Data and Formula, 628 pp., John Wiley & Sons, 1982.

カラーテーブルの詳細

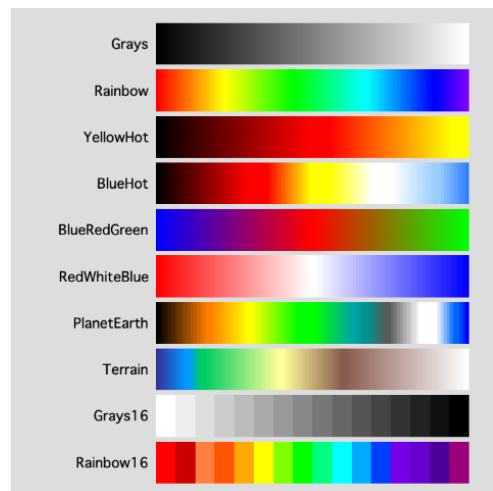
組み込みのカラーテーブルは、いくつかのカテゴリに分類できます。

Igor Pro 4 互換のカラーテーブル

Igor Pro 4 では、10 種類の組み込みカラーテーブルがサポートされていました。

Grays	Rainbow	YellowHot
BlueHot	BlueRedGreen	RedWhiteBlue
PlanetEarth	Terrain	Grays16
Rainbow16		

これらのカラーテーブルは、16 段階しかない Grays16 と Rainbow16 を除き、100 段階の色レベルを持っています。

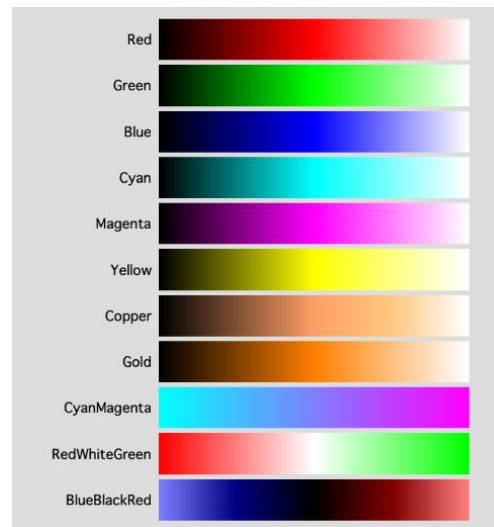


Igor Pro 5 互換のカラーテーブル

Igor Pro 5 では、Igor Pro 4 に搭載されていた 8 種類の 100 色レベルカラーテーブル (Grays256、Rainbow256 など) の 256 色バージョンに加え、新しいグラデーションカラーテーブルや、新しい専用カラーテーブルが追加されました。

Igor Pro 5 グラデーションカラーテーブル

これらは、2色または3色の間で 256 段階のグラデーションが変化するものです。



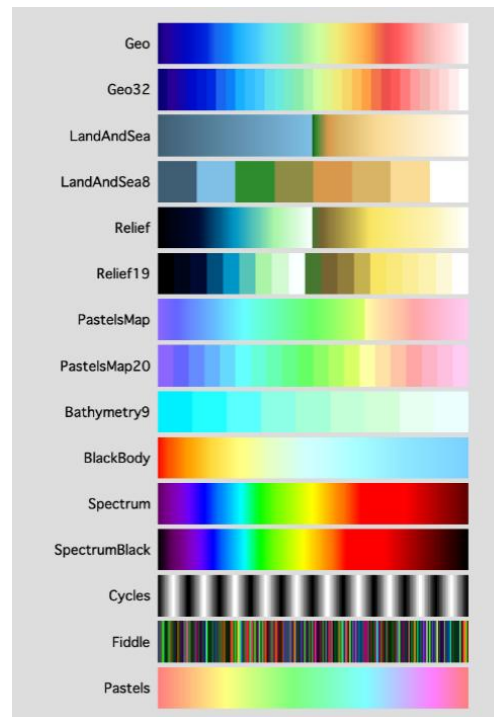
Igor Pro 5 の特殊用途のカラーテーブル

特殊用途のカラーテーブルとは、地形の高解像度デジタル標高モデル（DEM）への着色や分光分析など、特定の用途に使われるものです。

これらのカラーテーブルには、任意の数の色エントリを含めることができます。

次の表は、さまざまな専用カラーテーブルをまとめたものです。

カラーテーブル名	色数	説明
Geo	256	水域と陸域の色分けが施された一般的な地理用カラーテーブルで、海面は全体の約 50% を占めます。
Geo32	32	高度を分類するために量子化されています。海面はおよそ 50% です。
LandAndSea	255	海面（ちょうど 50% の位置）より上では、色が急速に変化します。海の深さは青灰色です。
LandAndSea8	8	概算すると、海面は約 22% です。
Relief	255	海拔 0 メートル地点（ちょうど 50% の位置）より上では、色の変化が緩やかになります。海中は黒色です。
Relief19	19	概算すると、海面は約 47.5% です。
PastelsMap	301	彩度の低い虹のような色合いで、海面付近（約 2/3、66.67% の位置）で緑から黄色へと急激に色が変わります。海の深さは淡い紫色になっています。
PastelsMap20	20	概算すると、海面は約 47.5% です。
Bathymetry9	9	海の深さの色です。海面は 100% です。

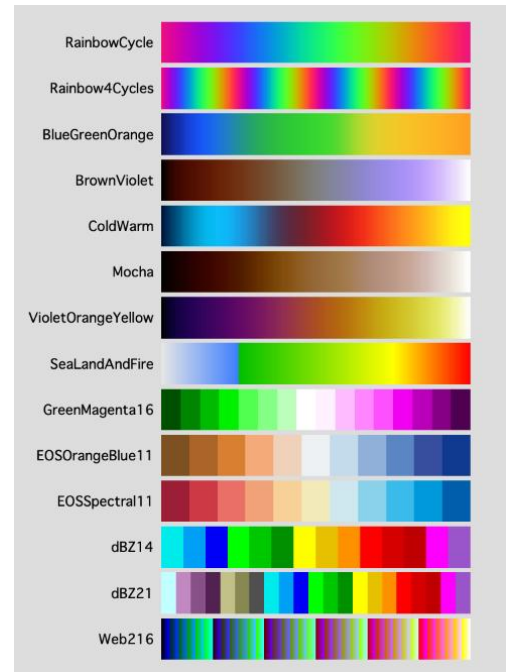


BlackBody	181	カラーテーブルの範囲を 1,000 °K から 10,000 °K に設定した場合、黒体放射の色（輝度は無視）に合わせて調整された赤→黄→青の色です。各カラーテーブルのエントリは 50 °K を表します。
Spectrum	201	カラーテーブルの範囲を 380~780 nm（波長）に設定すると、可視光スペクトルに合わせて調整された虹のような色になります。カラーテーブルの各エントリは 2 nm を表します。カラーテーブルの両端では、色が完全に黒へと変化することはありません。
SpectrumBlack	476	カラーテーブルの範囲を 355~830 nm（波長）に設定すると、可視光スペクトルに合わせて調整された虹のような色になります。カラーテーブルの各エントリは 1 nm を表します。カラーテーブルの両端では、色が黒へと徐々に変化します。
Cycles	201	0% から 100% を経て 0% に戻る、10 段階のグレースケール変化です。
Fiddle	254	画像内の微細なディテールを検出するために、画像を「いじくり回す」ためのランダムな色です。
Pastels	256	彩度を抑えた虹です。

Igor Pro 6 互換のカラーテーブル

Igor Pro 6 では 14 の新しいカラーテーブルが追加されました。

カラーテーブル名	色数	説明
RainbowCycle	360	赤、緑、青が正弦波状に変化し、それぞれ 120 度（120 段階）位相がずれます。最初と最後の色は同じです。
Rainbow4Cycles	360	4 サイクルで、角度分解能が 4 分の 1 です。
BlueGreenOrange	300	3 色のグラデーションです。
BrownViolet	300	2 色のグラデーションです。
ColdWarm	300	温度を表す多色グラデーションです。
Mocha	300	2 色のグラデーションです。
VioletOrangeYellow	300	温度を表す多色グラデーションです。
SeaLandAndFire	256	もう一つの等高線図です。海面は 25% の位置にあります。
GreenMagenta16	16	米国気象局の 14 色モーションカラーテーブル（基準速度または嵐相対値）と同じですが、赤緑色覚異常の方にも見やすい仕様となっています。
EOSOrangeBlue11	11	赤緑色覚異常の方にも配慮した発散データ用の色です。
EOSSpectral11	11	赤緑色覚異常の方にも配慮した調整された色です。
dBZ14	14	米国気象局の反射率（レーダー）カラー表示です。クリアエアモード（-28~+24 dBZ）または降水モード（5~70 dBZ）用です。

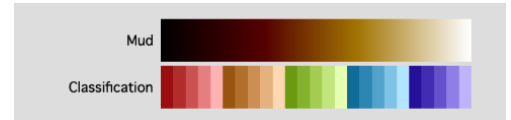


dBZ21	21	米国気象局の反射率（レーダー）カラー表示です（クリアエアモードと降水モードの合成、-30~70 dBZ）。
Web216	216	216色の「ウェブセーフカラー」は、単一のカラーテーブルに標準色を幅広く収録しています。ModifyGraph zColor(traceName)=... 機能を使ったf(z)のトレース着色を目的としています。

Igor Pro 6.2 互換のカラーテーブル

Igor Pro 6.2 では2つの新しいカラーテーブルが追加されました。

カラーテーブル名	色数	説明
Mud	256	濃い茶色から白色までです。Mocha カラーテーブルのようなピンクがかった色味はありません。Veeco 製原子間力顕微鏡用です。
Classification	25	分類用に5つの色相、各分類内のバリエーション用に5つの彩度を持ちます。



Igor Pro 9 互換のカラーテーブル

Igor Pro 9 では1つの新しいカラーテーブルが追加されました。

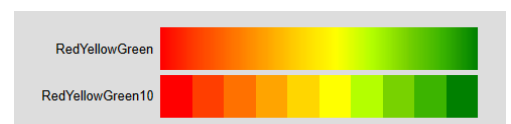
カラーテーブル名	色数	説明
Turbo	256	特定の色に割り当てられているという理由だけで特徴が過度に強調されるのを防ぐため、より滑らかな色調変化を実現した、逆順のレインボー風カラーテーブルです。 Daydream のシニアソフトウェアエンジニア、Anton Mikhailov によるデザインです。 参考： https://research.google/blog/turbo-an-improved-rainbow-colormap-for-visualization/



Igor Pro 10 互換のカラーテーブル

Igor Pro 10 では2つの新しいカラーテーブルが追加されました。

カラーテーブル名	色数	説明
RedYellowGreen	256	3色グラデーションです。
RedYellowGreen10	10	量子化された3色グラデーションです。



カラーテーブルウェーブ

インデックスカラーのウェーブ（「インデックスカラーの詳細」のセクションを参照）を、あたかもカラーテーブルであるかのように使用できます。

カラーインデックスウェーブを使う場合、画像のウェーブデータ値がカラーインデックスウェーブ内の X インデックスとして使われ、特定のポイントの色が選択されます。

結果として得られる色は、データ値とカラーインデックスウェーブの X スケーリングに依存します。

カラーテーブルのウェーブを使う場合、画像ウェーブのデータ値の全範囲、または明示的に指定した範囲が、カラーテーブルのウェーブ全体にマッピングされます。

結果として得られる色は、データ値と適用範囲のみに依存し、カラーテーブルのウェーブの X 軸スケーリングには依存しません。

カラーテーブルウェーブを生成する簡単な方法は、ColorTab2Wave コマンドを呼び出すことです。

これにより、M_Colors という名前の 3 列の RGB ウェーブが生成されます。

列 0 は赤成分、列 1 は緑、列 2 は青を表し、各値は 0（暗い）から 65535（明るい）までの範囲になります。

3 列の RGB カラーテーブルのウェーブでは、すべての色が不透明になります。

4 列目を追加して透明度をコントロールすることで、RGBA ウェーブにすることができます。

RGBA ウェーブの 4 列目は「アルファ」を表し、0 は完全に透明、65535 は完全に不透明となります。

Igor Pro Folder 内の「Color Tables」フォルダーには、多くのカラーテーブルウェーブが収録されており、それらを説明するヘルプファイルも含まれています。

ヘルプ Color Table Waves in the Igor Pro Folder を参照してください。

画像プロット、コンタープロット、グラフトレース、カラースケールでカラーテーブルウェーブを使うための構文はそれぞれ異なり、各コマンドの説明に詳述されています。

ModifyImage (ctab キーワード)、ModifyContour (ctabFill と ctabLines キーワード)、ModifyGraph (zColor キーワード)、ColorScale (ctab キーワード) を参照してください。

Igor Pro Folder 内の「Color Tables」を参照して、Igor Pro に同梱されているカラーテーブルウェーブをプレビュー、読み込むことができます。

インデックスカラーの詳細

インデックスカラープロットは、2D 画像ウェーブや、3D または 4D ウェーブのレイヤーとカラーインデックスウェーブを使います。

画像ウェーブのデータ値は、カラーインデックスウェーブ内の X インデックスとして使われ、特定の画像長方形の色を選択します。

結果として得られる色は、データ値とカラーインデックスウェーブの X スケーリングに依存します。

カラーインデックスウェーブとは、2D RGB または RGBA ウェーブのことです。

RGB ウェーブには 3 つの列があり、各行には 0（輝度ゼロ）から 65535（最大輝度）までの範囲の赤、緑、青の値が含まれます。

RGBA ウェーブには 3 つの色列に加え、0（完全透明）から 65535（完全不透明）までの範囲の値を持つアルファ列があります。

線形インデックスカラー

通常の線形インデックスカラーの場合、画像データ値に対応する X インデックスを持つカラーインデックスウェーブの行を選択することで、特定の画像データ値に対応する色を特定します。

Igor は、以下の計算を使って、画像データ値 `zImageValue` を行番号 `colorIndexWaveRow` に変換します。

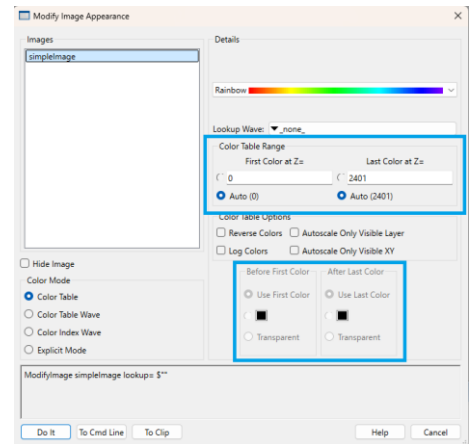
```
colorIndexWaveRow = floor(nRows*(zImageValue-xMin)/xRangeInclusive)
```

ここで

```
nRows = DimSize(colorIndexWave,0)
xMin = DimOffset(colorIndexWave,0)
xRangeInclusive = (nRows-1) * DimDelta(colorIndexWave,0)
```

`colorIndexWaveRow` が行範囲を超えている場合、Before First Color と After Last Color の設定 (Modify Image Appearance ダイアログ) が適用されます。

カラーウェーブの X スケーリングを設定することで、画像データの値をどのように色にマッピングするかをコントロールできます。これは、カラーテーブルの First Color at Z= と Last Color at Z= の値を設定することと似ています。



対数インデックスカラー

対数インデックスカラー (ModifyImage の log パラメーターが 1 に設定されている場合) では、 $\log(x \text{ scaling})$ と $\log(\text{image } z)$ を使って、次のように色がマッピングされます。

```
colorIndexWaveRow = floor(nRows*(log(zImageValue)-log(xMin))/(log(xMax)-log(xMin)))
```

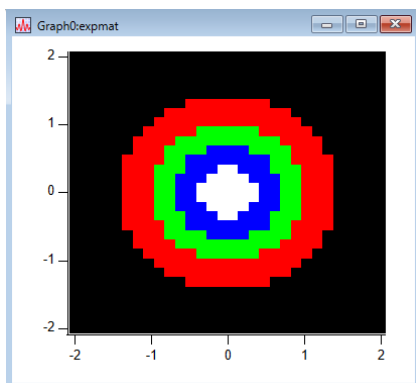
ここで

```
nRows = DimSize(colorIndexWave,0)
xMin = DimOffset(colorIndexWave,0)
xMax = xMin + (nRows-1) * DimDelta(colorIndexWave,0)
```

対数モードでの画像データの表示は、線形モードよりも処理に時間がかかります。

例：ポイントスケール付きカラーインデックスウェーブ

```
// ポイントスケールされた符号なし 16 ビット整数カラーインデックスウェーブを作成
Make/O/W/U/N=(1,3) shortindex // 最初は 1 行。今後追加される
shortindex[0][]= {{0},{0},{0}} // 最初の行に黒
shortindex[1][]= {{65535},{0},{0}} // 新しい行に赤
shortindex[2][]= {{0},{65535},{0}} // 新しい行に緑
shortindex[3][]= {{0},{0},{65535}} // 新しい行に青
shortindex[4][]= {{65535},{65535},{65535}} // 新しい行に白
// サンプルデータを生成し、カラーインデックスウェーブを使って表示
Make/O/N=(30,30)/B/U expmat // /B/U は符号なしバイト画像を作成
SetScale/I x,-2,2,"" expmat
SetScale/I y,-2,2,"" expmat
expmat= 4*exp(-(x^2+y^2)) // テスト画像は 0 から 4 の範囲
Display;AppendImage expmat
ModifyImage expmat cindex=shortindex
```



ダイレクトカラーの詳細

ダイレクトカラー画像では、赤、緑、青の絶対値を含む3つの色平面を持つ 3D RGB ウェーブ、またはアルファ平面を追加した 3D RGBA ウェーブが使われます。

一般的に、ダイレクトカラーウェーブは、符号なし 8 ビット整数または符号なし 16 ビット整数のいずれかです。

8 ビットの整数ウェーブの場合、0 は輝度 0、255 は輝度 100 を表します。

アルファ値の場合、0 は完全な透明、255 は完全な不透明を表します。

その他の数値型については、0 は輝度ゼロ、65535 は最大輝度を表します。

アルファ値については、0 は完全な透明、65535 は完全な不透明を表します。

範囲外の値は、上限または下限に切り詰められます。

次の例を、一行ずつ実行してみてください。

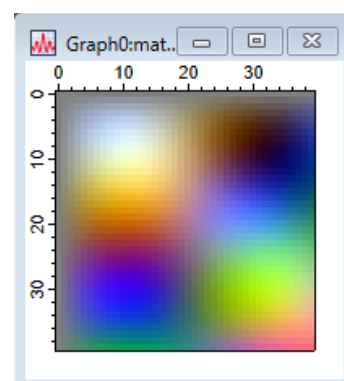
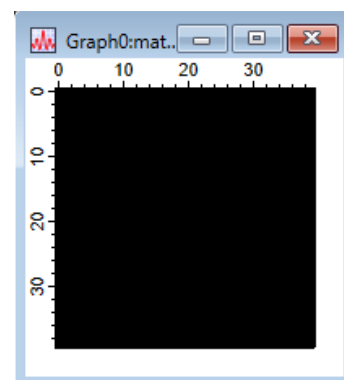
```
Make/O/B/U/N=(40,40,3) matr gb
```

```
NewImage matr gb
```

```
matr gb [][][0]= 127*(1+sin(x/8)*sin(y/8)) // 赤を指定, 0-255
```

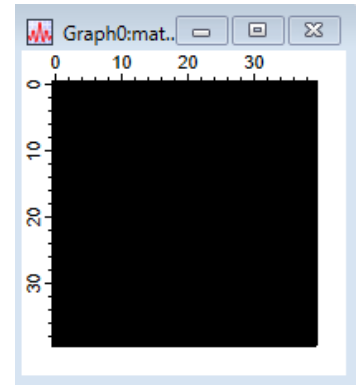
```
matr gb [][][1]= 127*(1+sin(x/7)*sin(y/6)) // 緑を指定, 0-255
```

```
matr gb [][][2]= 127*(1+sin(x/6)*sin(y/4)) // 青を指定, 0-255
```



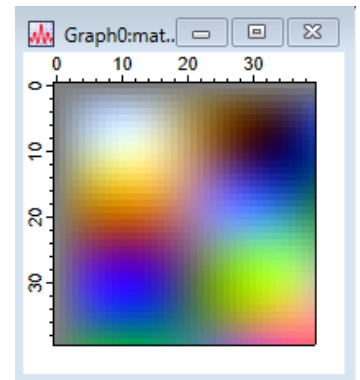
// 浮動小数点に切り替えると、画像が真っ黒になる

```
Redimension/S matrgb
```



// 浮動小数点を 0..65536 の範囲にスケーリング

```
matrgb *= 256
```



ダイレクトカラー画像の外観は画像データによって完全に決定されるため、Modify Image Appearance ダイアログはダイレクトカラー画像には影響を与えず、ダイアログは空白のまま表示されます。

ダイレクトカラーのパッキングモード

Igor Pro 10 以前のバージョンでは、ダイレクトカラー画像には、標準的な RGB(A) 色成分が連続したカラープレーンに格納された 3D ウェーブが必要でした。

多くのカメラや画像ファイル形式では、データがさまざまな順序のピクセルのストリームとして保存されます。

このような画像を効率的に表示するために、Igor Pro 10 では以下の新しいパッキングモードのサポートが追加されました。

packSTD	カラー画像が 3D ウェーブの RGB または RGBA レイヤーとして保存されるデフォルトのモードです。
packFastU8U16	8 ビットまたは 16 ビットの符号なし整数（1ピクセルにつき1つの整数）の 2D ウェーブとして保存されたモノクロ画像です。ここで、ウェーブの次元は画像のピクセル数と等しくなります。
packU32RGBFF	カラー画像は、2D 32 ビット符号なし整数ウェーブとして保存されます。色は、各色成分につき1バイト（RGB 順）でパッキングされ、その後に 0xFF に設定された固定のアルファ成分が続きます。ウェーブの列数は、画像の高さ（ピクセル単位）の4倍です。
packU32RGBA	カラー画像は、2D 32 ビット符号なし整数ウェーブとして保存されます。色は、各色成分につき1バイト（RGBA 順）でパッキングされています。ウェーブの列数は、画像の高さ（ピクセル単位）の4倍です。
packU32FFRGB	カラー画像は、2D 32 ビット符号なし整数ウェーブに格納されます。色は、各色成分につき1バイトでパッキングされます（アルファ値は常に 0xFF の ARGB 順）。ウェーブの列数は、画像の高さ（ピクセル単位）の4倍です。

packU32ARGB	カラー画像は、2D 32 ビット符号なし整数ウェーブに格納されます。色は、各色成分につき1バイト (ARGB 順) でパッキングされています。ウェーブの列数は、画像の高さ (ピクセル単位) の4倍です。
pack4U8RGBA	カラー画像は、2D 符号なしバイトウェーブとして保存されます。色は、各色成分につき1バイト (RGBA 順) でパッキングされています。ウェーブの列数は、画像の高さ (ピクセル単位) の4倍です。
pack4U8ARGB	カラー画像は、2D 符号なしバイトウェーブとして保存されます。色は、各色成分につき1バイト (ARGB 順) でパッキングされています。ウェーブの列数は、画像の高さ (ピクセル単位) の4倍です。
pack4U8BGRA	カラー画像は、2D 符号なしバイトウェーブとして保存されます。色は、各色成分につき1バイト (BGRA 順) でパッキングされています。ウェーブの列数は、画像の高さ (ピクセル単位) の4倍です。
pack3U8RGB	カラー画像は、2D 符号なしバイトウェーブとして保存されます。色は、各色成分につき1バイト (RGB 順) でパッキングされています。ウェーブの列数は、画像の高さ (ピクセル単位) の3倍です。
pack3U8BGR	カラー画像は、2D 符号なしバイトウェーブとして保存されます。色は、各色成分につき1バイト (BGR 順) でパッキングされています。ウェーブの列数は、画像の高さ (ピクセル単位) の3倍です。

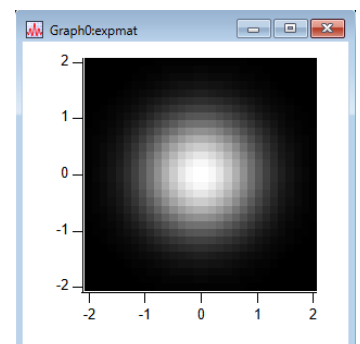
色の凡例の作成

カラースケール注釈を使って、色の凡例を作成できます。
 詳細については、ヘルプ Legends とヘルプ Color Scales を参照してください。

簡単な画像プロットを使って説明します。

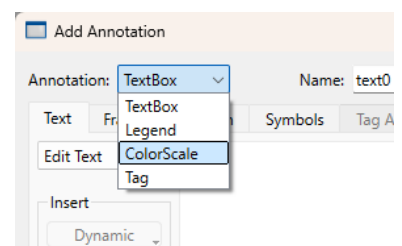
これにより、自動調整されたグレースケールカラーテーブルを使って、右のような画像が生成されます。

```
Make/O/N=(30,30) expmat
SetScale/I x,-2,2,"" expmat; SetScale/I y,-2,2,"" expmat
expmat= exp(-(x^2+y^2)) // 0 から 1 の範囲のデータ
Display;AppendImage expmat // デフォルト、左軸と下軸
ModifyGraph width={Plan,1,bottom,left},mirror=0
```

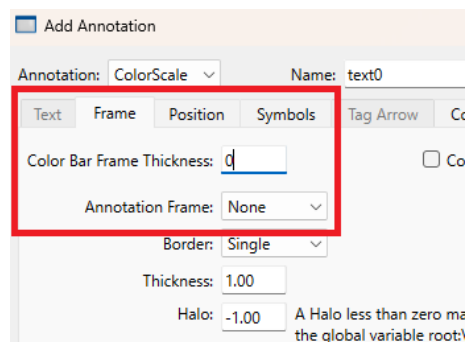


Graph→Add Annotation を選択して、Add Annotation ダイアログを表示します。

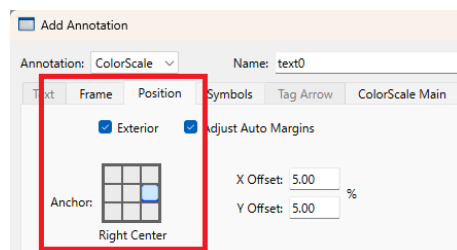
Annotation ポップアップメニューから「ColorScale」を選択します。



Frame タブに切り替え、Color Bar Frame Thickness を 0 に、Annotation Frame を None に設定します。



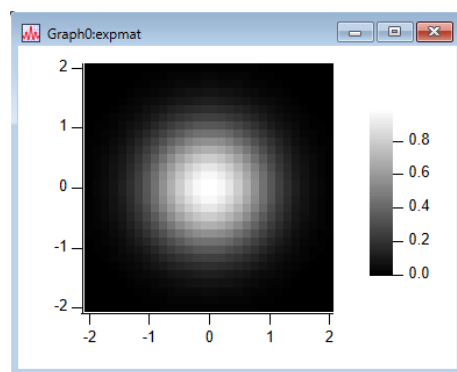
Position タブに切り替え、Exterior チェックボックスをオンにし、Anchor を Right Center に設定します。



Do It をクリックします。
Igor は次を実行します。

```
ColorScale/C/N=text0/F=0/A=RC/E image=expmat,frame=0.00
```

これにより、右のような画像プロットが生成されます。



画像インスタンス名

Igor は、Z 値を提供するウェーブの名前の画像プロットを識別します（これは、Image Plot ダイアログの Z Wave リストで選択されたウェーブです）。

この「画像インスタンス名」は、画像プロットを変更するコマンドで使われます。

次の例では、画像インスタンス名は「zw」です。

```
Make/O/N=(30,30) zw
SetScale/I x,-2,2,"" zw
SetScale/I y,-2,2,"" zw
zw= exp(-(x^2+y^2))
Display; AppendImage zw // 新しい画像プロット
ModifyImage zw ctab={*,*,BlueHot} // カラーテーブルを変更
```

グラフに同じウェーブの画像プロットが2つ含まれているという珍しいケース（例えば、データの異なるサブレンジ（一部）を並べて表示する場合など）では、2つ目のプロットを変更するために、名前にインスタンス番号を付加する必要があります。

```
Display; AppendImage zw; AppendImage/R/T zw // 2つの画像プロット
ModifyImage zw ctab={*,*,RedWhiteBlue} // 最初のプロットを変更
ModifyImage zw#1 ctab={*,*,BlueHot} // 2番目のプロットを変更
```

Modify Image Appearance ダイアログでは、正しい画像インスタンス名が自動的に生成されます。画像インスタンス名の仕組みは、グラフ内のトレースに対するウェーブインスタンス名の仕組みとほぼ同じです。ヘルプ Instance Notation を参照してください。

ImageNameList 関数は、画像インスタンス名の文字列リストを返します。

各名前は、グラフ内の1つの画像プロットに対応しています。

ImageInfo 関数は、特定の名前が付けられた画像プロットに関する情報を返します。

ImageNameList は文字列を返しますが、ModifyImage は名前を使用します。

\$ 演算子は、文字列を名前に変換します。

例えば：

```
Function SetFirstImageToRainbow(graphName)
  String graphName
  String imageInstNames = ImageNameList(graphName, ";")
  String firstImageName = StringFromList(0,imageInstNames)
  if (strlen(firstImageName) > 0)
    ModifyImage/W=$graphName $firstImageName ctab={,,Rainbow} // $ は文字列を名前に変換
  endif
End
```

// 名前 (文字列) のリスト
// 文字列での名前

画像プロットを表示し、コマンドラインで次を実行すると確認できます。

```
SetFirstImageToRainbow("Graph0")
```

画像プロットのプレファレンス

画像プロットのデフォルトの外観は、画像プロットを含むプロトタイプグラフのプレファレンスをキャプチャすることで変更できます。

よく使う設定で画像プロットを含むグラフを作成します。

次に、Graph メニューから Capture Graph Prefs を選択します。

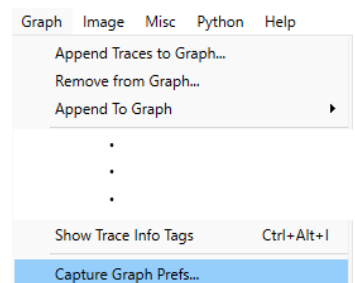


Image Plots カテゴリを選択し、Capture Prefs をクリックします。

プレファレンスは通常、手動操作に対してのみ有効であり、プロシージャによる自動操作には適用されません。

ヘルプ Procedures and Preferences を参照してください。

Image Plots カテゴリには、画像の外観設定と軸の設定の両方が含まれています。

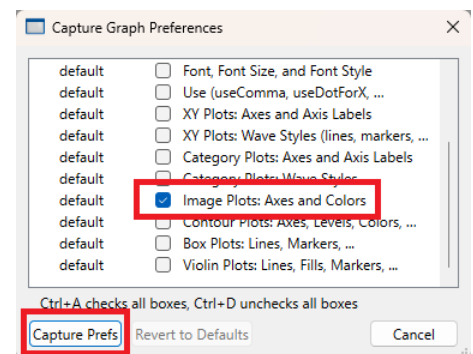


Image Appearance プレファレンス

Image Appearance の設定は、プレファレンスが有効になっている場合、画像プロットが最初に作成される時に自動的に適用されます。

また、これらの設定は、New Image Plot ダイアログのサブダイアログとして Modify Image Appearance ダイアログが呼び出された時、そのダイアログの初期設定としても使われます。

複数の画像プロットを含むグラフから画像プロットのプレファレンスを取得する場合、グラフに最初に追加された画像プロットは、プロトタイプとなるグラフに最初に追加された画像の設定を引き継ぎます。

グラフに2番目に追加された画像プロットは、プロトタイプとなるグラフに2番目に追加された画像の設定を引き継ぎます。以下同様です。

これは、XY プロットのウェーブスタイルの動作と似ています。

Image Axis プレファレンス

画像プロットで使われる軸のみ、その設定が取得されます。

XYプロット、カテゴリプロット、またはコンタープロットにのみ使われる軸は無視されます。

Image Axis プレファレンスは、AppendImage コマンドによって、キャプチャされた軸と同じ名前の軸が作成された場合にのみ適用されます。

AppendImage の実行前にその軸が存在していた場合、Image Axis プレファレンスは適用されません。

キャプチャされた画像の軸名は、New Image Plot ダイアログと Append Image Plot ダイアログの X Axis と Y Axis ポップアップメニューに一覧表示されます。

これは、XY プロットのプレファレンスの動作と同様です。

例えば、「myRightAxis」と「myTopAxis」という名前の軸を使って、画像プロットのプレファレンスを保存するとします。

これらの名前は、画像プロットダイアログの X Axis と Y Axis のポップアップメニューに表示されます。

New Image Plot ダイアログでこれらを選択し、Do It をクリックすると、「myRightAxis」と「myTopAxis」という名前の新規作成された軸を含み、キャプチャした軸の設定が適用されたグラフが作成されます。

すでに「myRightAxis」と「myTopAxis」という名前の軸が設定されているグラフがあり、Append Image Plot ダイアログでこれらの軸を選択した場合、画像は通常通りそれらの軸に追加されますが、キャプチャされた軸の設定は、これらの既存の軸には適用されません。

標準の左軸および下軸の Image Axis プレファレンスを保存することができ、Igor はこれらを、XY プロット、カテゴリプロット、等高線プロット用に保存された左軸および下軸のプレファレンスとは別に保存します。

Igor は、AppendImage コマンドに対してのみ、この Image Axis プレファレンスを使います。

画像プレファレンスの使い方

画像プレファレンスを活用するための推奨戦略は以下の通りです。

1. 1つの画像プロットを含む新しいグラフを作成します。たとえ軸が左と下であっても、普段使っている軸を使います。他の軸を使うことも可能です（New Image Plot と Append Image Plot ダイアログで New Axis を選択します）。
2. 画像のプロットを好みの表示に調整するために Modify Image Appearance、Modify Graph、Modify Axis ダイアログを使います。
3. Graph メニューから Capture Graph Prefs を選択します。Image Plots カテゴリを選択し、Capture Prefs をクリックします。

画像プロットのショートカット

画像プロット全体の外観を変更する

右クリックし、コンテキストメニューから Modify Image を選択します。これにより、Modify Image Appearance ダイアログが表示されます。