

CONTENTS

サンプルの Experiment – Transform Axis Demo.....	2
クイックノート	2
はじめに	2
Transform Axis（軸変換）パッケージのヘルプ	2
概要.....	2
Transform Axis パッケージの読み込み	3
鏡像軸の変換.....	6
軸変換はどのように動作するのか.....	6
Modify Axis タブを使って軸変換を修正.....	7
軸変換オプションの変更	8
変換軸の非変換化	9
目盛の編集	9
独自の変換を追加する.....	12
独自のプロシージャで使うことができる関数.....	14
Transform Axis の以前のバージョンからのグラフ	15
Transform Axis バージョン 1.2 の新機能.....	16

サンプルの Experiment – Transform Axis Demo

クイックノート

メニュー **File** → **Example Experiments** → **Graphing Techniques** → **Transform Axis Demo**

この Experiment は、Transform Axis パッケージを使う方法を説明するデモです。
これは、TransformAxis のヘルプファイルに示されているサンプルデータセットを使っています。
関連する Experiment の例の「Probability Graph Demo」も参照してください。

はじめに

クイックノートにはすでにデータが格納された状態での簡単な操作方法となっているため、ここからは Transform Axis Package のヘルプ (TransformAxis Help.ihf) の内容を記載します。
サンプルの Experiment のグラフウィンドウの上部の Display Transform Axis Help ボタンをクリックするとヘルプが表示されます。
新しい Experiment を作成し、このヘルプを表示するには、コマンドウィンドウで次を実行します：
`DisplayHelpTopic "Transform Axis Package"`

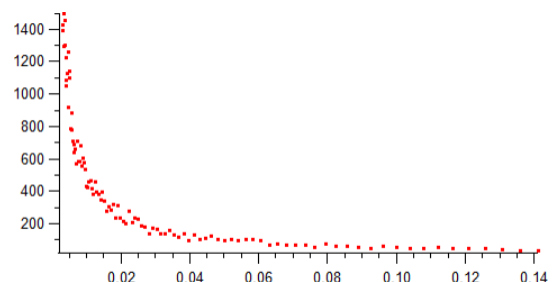
Transform Axis (軸変換) パッケージのヘルプ

概要

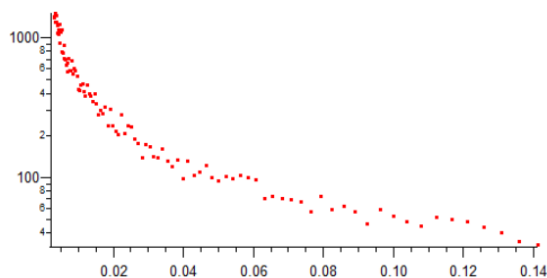
グラフ化する前にデータを加工すると、グラフが直線になるようにしたり、データの範囲によってグラフの割合が変化したりする場合に便利です。
この作業を行う時には、通常、変換前の元の単位でラベル付けされた軸目盛を設定するのが便利です。
これを軸変換 (Transform Axis) と呼びます。

よく知られている軸変換の例としては、対数軸があります。

例えば、右のグラフでは下と左の軸の近くに多くのデータがあります。



左軸に対数変換を適用すると、小さな値でもより詳細な値がわかります。



変換を適用する理由の一つは、データが特定の関数系に従うと予想できる理由がある場合です。正しい変換を適用すると、データは直線に近づくはずですが、これは、元のグラフの形状から、逆数変換が適している可能性があることを示唆しています。

Igor には、対数軸変換が組み込みオプションとして用意されていますが、その他の変換機能は提供されていません。

そこで、Transform Axis パッケージを使います。

Transform Axis パッケージの読み込み

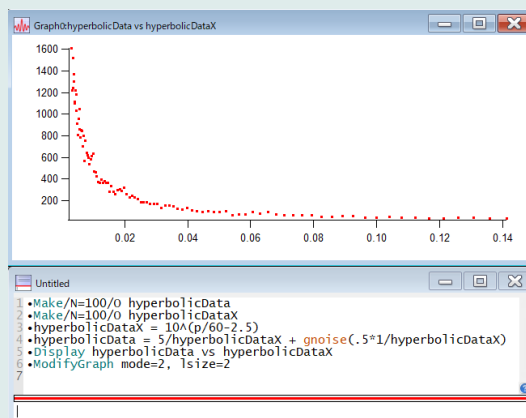
Transform Axis パッケージを読み込むもっとも簡単な方法は Graph メニューの Packages から選択することです。

新しい Experiment を作成したところからの手順で確認します。

1. コマンドラインで次を実行してサンプルデータを作ります。

このデータは、右図のようなグラフを生成します。

```
Make/N=100/O hyperbolicData
Make/N=100/O hyperbolicDataX
hyperbolicDataX = 10^(p/60-2.5)
hyperbolicData = 5/hyperbolicDataX
                + gnoise(.5*1/hyperbolicDataX)
Display hyperbolicData vs hyperbolicDataX
ModifyGraph mode=2, lsize=2
```



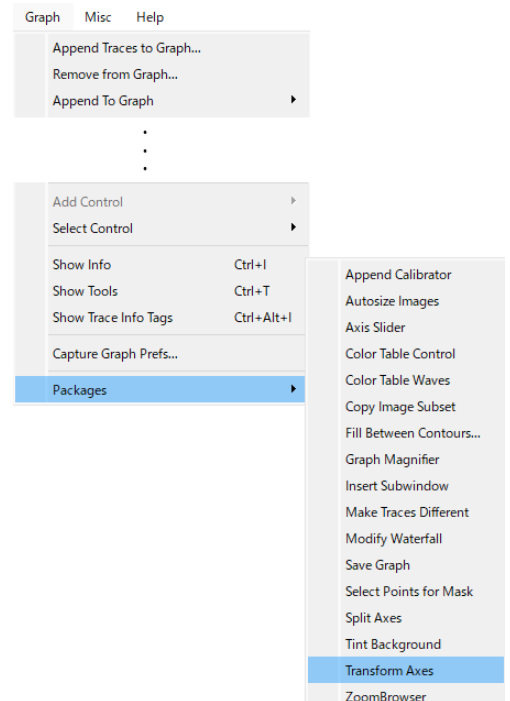
2. グラフウィンドウを前面にし、Graph メニューが表示されるようにします。

メニュー Graph → Packages → Transform Axes を選択します。

Transform Axis Control ダイアログが表示されます。

Graph メニューに Transform Axis という項目が追加され、Procedure Windows にも次の行が追加されます。

```
#include <TransformAxis1.2>
```

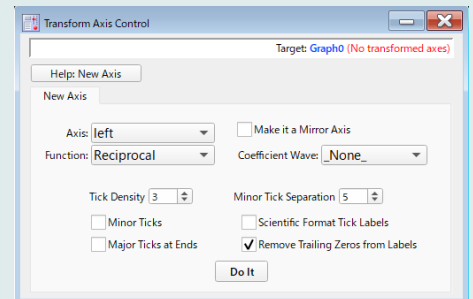


3. 上記のダイアログが表示されていない場合は、メニュー Graph → Transform Axis → New, Modify or Undo Axis を選択します (または、グラフウィンドウの空白エリアで右クリックして、ポップアップメニューから Transform Axis → Transform Axis を選択します)。

グラフにはまだ変換された軸がないため、「New Axis」タブのみが表示されています。

データの作成方法から、今回は、相互変換 (reciprocal transformation) がデータの表示に適した方法であることはすでに分かっています。

相互変換の特徴として、どちらの軸も同様の効果で変換できます。デフォルトでは、コントロールパネルは次の設定になっています。



Function:	Reciprocal
Axis:	left
Coefficient Wave:	_None_
Make it a Mirror Axis:	unchecked
Minor Ticks:	unchecked
Major Ticks at Ends:	unchecked
Scientific Format Tick Labels:	unchecked
Tick Density:	3
Minor Tick Separation:	5

4. これらは今回の目的に適した設定のため、Do It をクリックします。

グラフは右図のようになります。

データは $1/x$ の依存関係で作成されたため、変換後のデータは（ノイズの追加によるばらつきはありますが）直線に近くなります。

Transform Axis パッケージには、いくつかの組み込みの変換機能が用意されています。

「組み込み」の変換は、TransformAxis.ipf プロシージャファイルの一部であるユーザー関数によって定義されています。

独自の変換関数を定義することもできます。

詳細は、後半の「独自の变換を追加する」を参照してください。

提供される変換には、次の逆変換の 2 つのバージョンが含まれています（ModifiedReciprocal については後述）。

Reciprocal
ModifiedReciprocal

いくつかの簡単な温度変換もあります。

DegreesCtoF
DegreesFtoC
DegreesCtoK
DegreesKtoC
DegreesFtoK
DegreesKtoF

また、正規確率グラフを作成するための 2 つの変換があります。

Probability_Percent
Probability

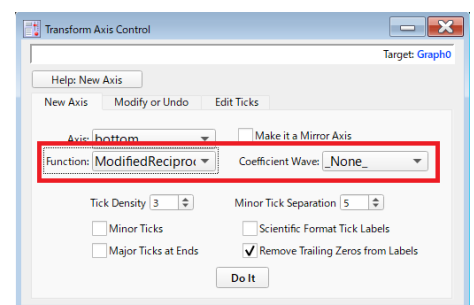
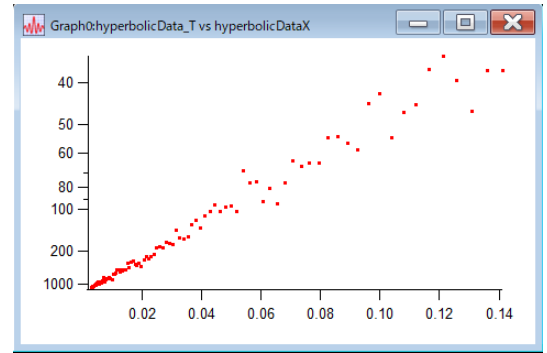
最後の 2 つの変換、Probability_Percent と Probability は、Probability Graph パッケージでの使用を目的として含まれていますが、すでに確率（0 から 1 の範囲）または確率パーセンテージ（0 から 100 の範囲）の値を持つデータセットがある場合は、これらを使うこともできます。

温度変換では、単純なオフセットと線形スケール要素により、ある温度スケールから別の温度スケールへの変換を行います。

これらは線形であるため、鏡像軸を作るときにのみ有用です。

ModifiedReciprocal 変換は、 $f(x) = w[0]/(x+w[1])$ という変換を適用します。係数 $w[0]$ と $w[1]$ は、係数ウェーブを介して指定され、変換軸を作成する前に、適切な値で作成し、入力しておく必要があります。

この係数ウェーブは、Transform Axis コントロールパネルの Coefficient Wave メニューから選択する必要があります。

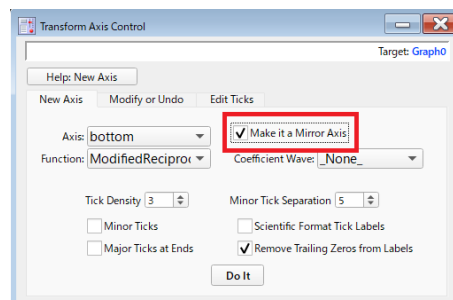


鏡像軸の変換

プロットしたデータを変換したくない場合もあります。

例えば、変換された数値に従ってラベル付けられた鏡像軸を作成したい場合です。

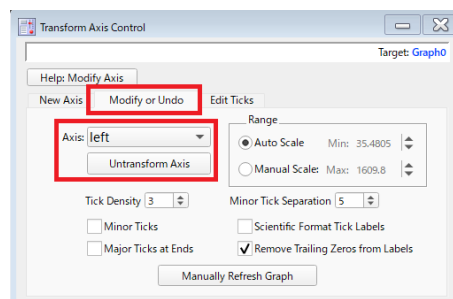
Do It をクリックする前に、Make It a Mirror Axis チェックボックスにチェックを入れると、Transform Axis コントロールパネルが自動的に処理してくれます。



今回の例のグラフで逆数変換を適用するために、まず左軸の変換を元に戻します。

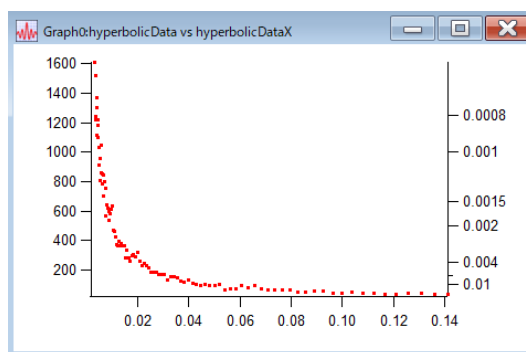
コントロールパネルの「Modify or Undo」タブで、Axis で left を選択し、Untransform Axis ボタンをクリックします。

左の軸が元に戻ります。



コントロールパネルで、Make It a Mirror Axis にチェックを入れ、Do It をクリックします。

右図のような結果になります。



変換軸はどのように動作するのか

変換軸は、Igor のユーザー目盛機能を使って作成されます。

この機能によって、ウェーブを使って任意の目盛と目盛ラベルを指定することができます。

変換軸を作成するには、まず、Transform Axis パッケージが、選択した変換に従って変換されたデータを含む新しいウェーブを作ります。

グラフ内のウェーブは、これらのウェーブに置き換えられます。

次に、変換後の空間における軸上の適切な位置に収まるように、元のデータ空間の適切な数値で目盛が計算されます。

これは容易なことではありません。

変換が非常に非線形である場合は、うまくいかないこともあります。

鏡像変換軸には、変換データは必要なく、変換された軸だけが必要です。

これは Truly Free Axis (自由軸) と軸フック関数を使って行われます。

Truly Free Axis についての詳細は、NewFreeAxis、ModifyFreeAxis、KillFreeAxis のヘルプを参照してください。

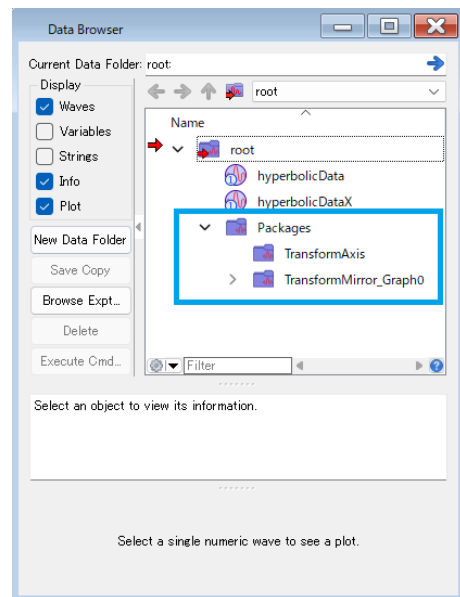
変換軸をプロットするために使われるウェーブは、秘密のデータフォルダーに保存されています。

それらを見るには、データフォルダーの内容を表示するために Data Browser を使用してください。

Packages:AxisTransform_ggg:AxisTransform_aaa、ggg は変換軸を含むグラフの名前、aaa は変換軸に変換された軸の名前です。

軸が鏡像変換軸である場合、データフォルダーのプリフィックスは AxisTransform_ ではなく、TransformMirror_ となります。

長いグラフ名は、データフォルダー名の許容できる長さを超えることがあり、その場合は名前が切り捨てられることに注意してください。



Modify Axis タブを使って変換軸を修正

変換軸は実際の Igor の軸であるため、その外観を変更するには Modify Axis ダイアログ (メニュー Graph → Modify Axis) を使います。

線の幅、ラベルのスタイル、フォント種類、フォントサイズの変更、グリッドの追加などを行うことができます。

ただし、変換軸に対しては、いくつかやってはいけないことがあります。

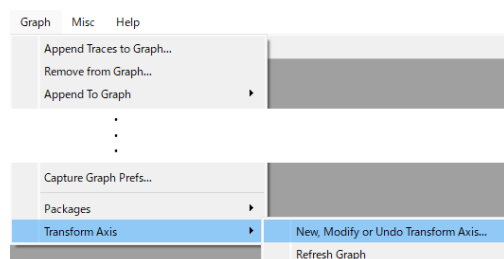
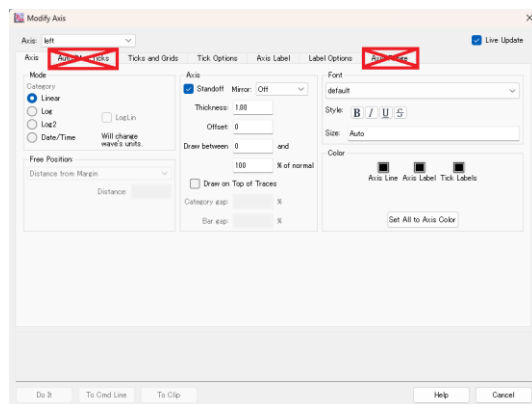
まず、Modify Axis ダイアログの「Axis Range」タブは使わないでください。

次に、「Auto/Man Ticks」タブは使わないでください。

これは、変換軸を実装するときを使う方法の核心に影響するものです。

変換軸の範囲を変更するには、Transform Axis パネルを表示します。

これを表示するには、メニュー Graph → Transform Axis → New, Modify or Undo Transform Axis を選択するか、グラフ内の空白部分を右クリックして Transform Axis → Transform Axis を選択します。



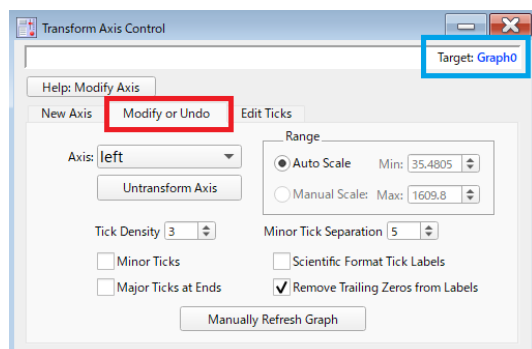
次に、パネル内の「Modify or Undo」タブに切り替えます。

このタブが表示されていない場合は、おそらく間違ったグラフが前面にあると思われます。

その場合、変換軸を持つグラフを前面に移動し、パネルをもう一度クリックします。

パネルは常に前面のグラフに対して動作します (パネルの右上隅に現在のターゲットが表示されます)。

ここでは、Axis ドロップダウンメニューで選択した変換軸の範囲を設定できます。



変換軸にゼロ線を加えても問題ありません。

ただし、ゼロ線は変換された空間ではゼロの位置に配置されますが、必ずしもゼロとラベルされた目盛の位置とは限らないことに注意してください。

したがって、確率グラフでは、ゼロ線は確率 0.5 の位置に配置されます。

主軸のスタイルに変更を加えると、変換鏡像軸がそれを検知し、鏡像軸が変更されます。

鏡像軸を変更しても主軸は変わりません。

鏡像軸の変更は、主軸で同じ個所を変更しない限り、主軸はそのまま維持されます。

変換軸オプションの変更

Transform Axis コントロールパネルの「Modify or Undo」タブを使って、変換軸の特定の要素を変更できます。

これには、軸の範囲、目盛の密度、補助目盛の追加などが含まれます。

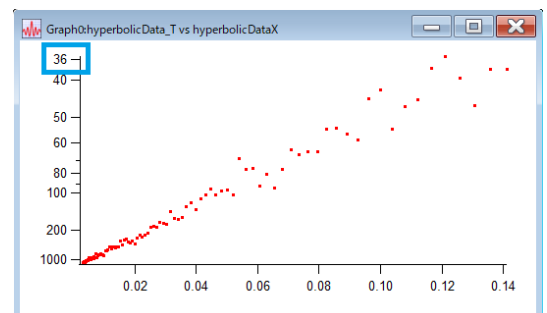
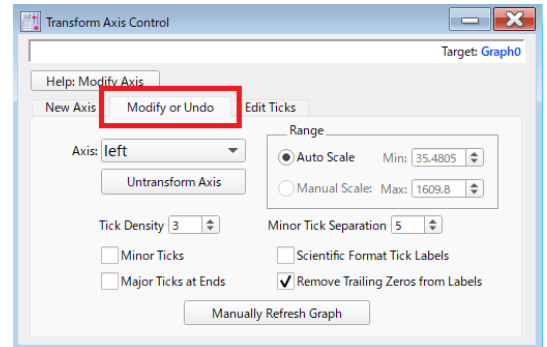
これを表示するには、メニュー Graph → Transform Axis → New, Modify or Undo Transform Axis を選択するか、グラフ内の空白部分を右クリックして Transform Axis → Transform Axis を選択します。

Minor Ticks チェックボックスをオンにすると、補助目盛が追加されます。

変換の非線形性が、軸の終点に非常に近い場所で、適切な目盛を見つけるのを難しくすることがあります。

実際、この例のグラフには最後に大きなスペースがあります。

Major Ticks at Ends チェックボックスにチェックを入れると、通常では表示されない追加の目盛が追加されます。



Tick Density 設定は、Modify Axis ダイアログの「Auto/Man Ticks」タブの Automatic Ticks セクションにある Approximately 設定のようなものです。

通常の軸の場合、この設定により、軸上にその数の目盛を配置するように指示します。

Igor は、その目盛数になるように、きれいに目盛を計算します。

線形変換を使っている場合、Tick Density 設定は通常の軸の Approximately 設定と同じように機能します。

この対応は、コードが変換の非線形性を把握することは極めて困難であるため、変換が非線形であるときには崩壊します。

ベストなアプローチは、最善の環境になるまで試行錯誤することです。

通常の軸の Approximately 設定とは異なり、Tick Density 設定は整数である必要はありません。

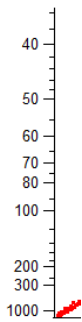
場合によっては、特に非線形性の高い変換関数では、小数点以下の値が最適な場合があります。

Minor Tick Separation パラメーターは、通常の軸の同じ設定と同様に機能します。

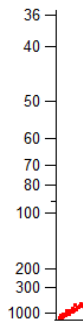
少なくともポイントの間隔がその大きさになるように補助目盛が追加されます。

(実際には、これは単純化しすぎです。実際にコントロールされているのは、(Major Tick Spacing)/(N Minor Ticks) です。つまり、2つの主目盛間の補助目盛の平均間隔は、Minor Tick Separation パラメーターよりも大きくしなければなりません。)

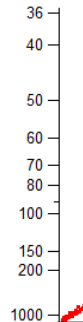
この例における、これらのオプションの効果を示したものをいくつか示します。



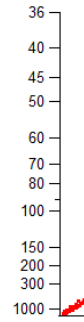
Minor Ticks: ON
 Ticks at Ends: OFF
 Tick Density: 3
 Minor Sep: 5



Minor Ticks: OFF
Ticks at Ends: ON
 Tick Density: 3
 Minor Sep: 5



Minor Ticks: OFF
 Ticks at Ends: ON
Tick Density: 5
 Minor Sep: 5



Minor Ticks: OFF
 Ticks at Ends: ON
Tick Density: 7
 Minor Sep: 5

元のデータがゼロに近づくとき、逆数変換の極端な非線形性により、下部の目盛が非常に接近していることに注意してください。

変換が非線形であるほど、きれいな目盛を計算することは難しくなります。

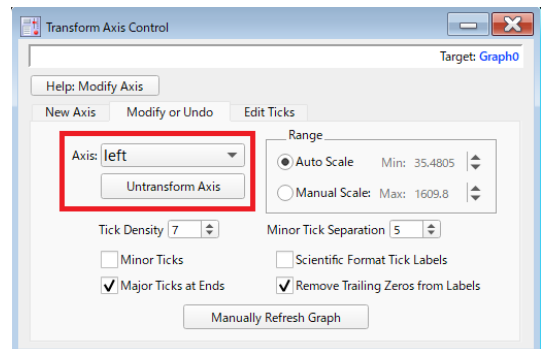
また、Minor Ticks オプションがオフになっている場合でも、補助目盛が表示されることに注意してください。これは、ラベルを表示するスペースがない主目盛が原因です。

変換軸の非変換化

軸を変換軸にしたいと判断した場合は、Transform Axis パネルの「Modify or Undo」タブで変換を元に戻すことができます。

Axis ドロップダウンメニューから目的の軸を選択し、Untransform Axis ボタンをクリックします。

元のデータがグラフに復元され、軸の外観が変換軸になる前の状態に戻ります。

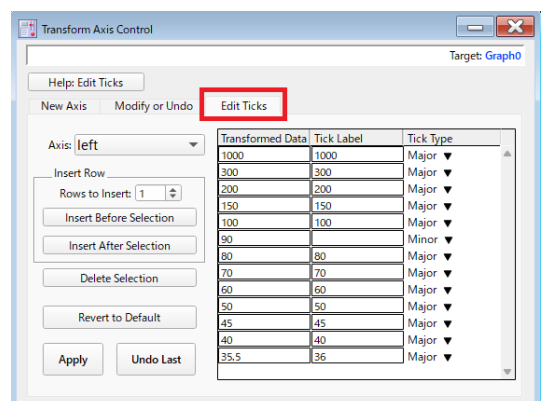


目盛の編集

時々、変換軸の目盛を選択するアルゴリズムがうまく機能しないことがあります。これは、特に、非常に非線形な変換の場合に起こります。

より良くできる可能性があるため、Transform Axis パッケージでは、変換軸の目盛を編集できるようにしています。

これは、Transform Axis パネルでも同様に行うことができます。これを表示するには、メニュー Graph → Transform Axis → New, Modify or Undo Transform Axis を選択するか、グラフ内の空白部分を右クリックして Transform Axis → Transform Axis を選択します。パネル内の「Edit Ticks」タブを選択します。

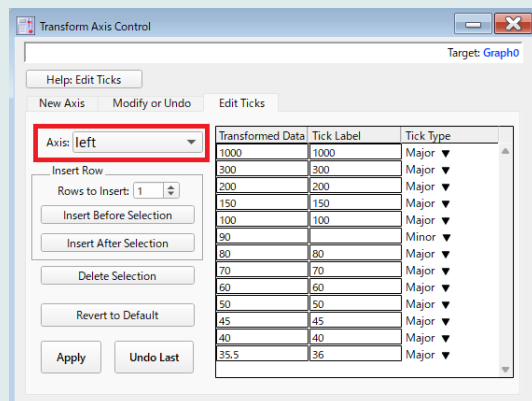


このタブが表示されていない場合は、おそらく間違ったグラフが前面にあると思われます。
その場合、変換軸を持つグラフを前面に移動し、パネルをもう一度クリックします。
パネルは常に前面のグラフに対して動作します（パネルの右上隅に現在のターゲットが表示されます）。

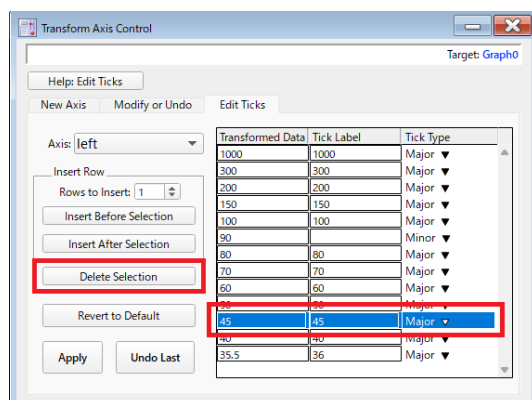
右側のリストボックスには、選択した変換軸上のすべての目盛が表示されます。

目盛を編集するには、次の手順に従います：

1. Axis メニューから適切な軸を選択します。

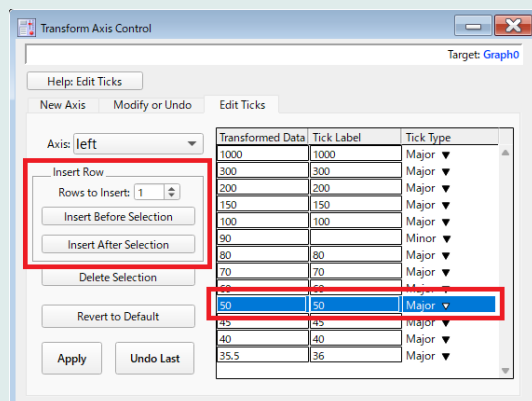


2. グラフから目盛を削除するには、該当する行または複数行を選択し、Delete Selection ボタンをクリックします。

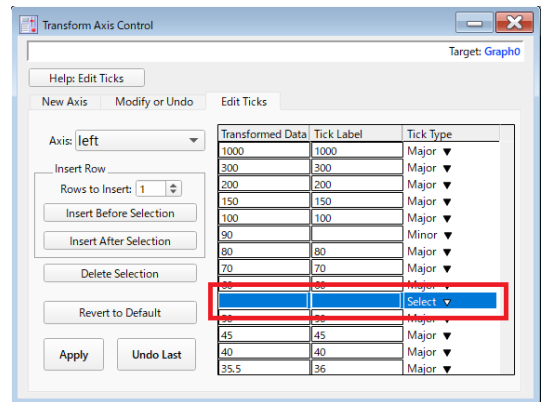


3. グラフに目盛を追加するには、行を選択し、Insert Before Selection または Insert After Selection をクリックします。

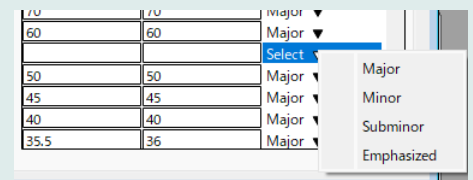
Rows to Insert を変更することで、一度に複数の行を挿入することができます。



4. これにより、リストボックスに空白の行が追加されます。



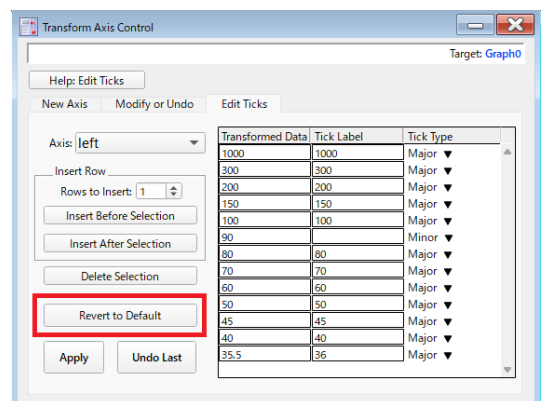
5. この空白の3つの列を埋める必要があります。



- ① Transformed Data 列は目盛を配置する値を入れます。
この数値は目盛を配置するために使われ、変換前の単位で表されます。
- ② Tick Label 列は、目盛ラベルを入れます。
任意の文字列を入力できます。
通常は、空白か Transformed Data 列と同じ数値を入力します。入力した内容がそのまま目盛ラベルとして表示されるため、任意のフォーマットを使うことができます。
グラフの注釈で使うようなテキストフォーマットのエスケープコードも許可されています。
- ③ 目盛の種類を選択します。Major (主目盛)、Minor (補助目盛)、Subminor (補助の補助目盛)、Emphasized (強調) のどれかを選びます。

6. 加えた編集は記憶されるため、軸が再描画される時(例えば、ウィンドウサイズを変更、軸の範囲を変更したとき)、Transform Axis パッケージは、削除された目盛を削除し、追加した目盛を追加しようと試みます。

Revert to Default ボタンをクリックすれば、Transform Axis パッケージで計算された目盛だけを復元することができます。
これにより、これまでの目盛編集セッションの記録が消去されます。



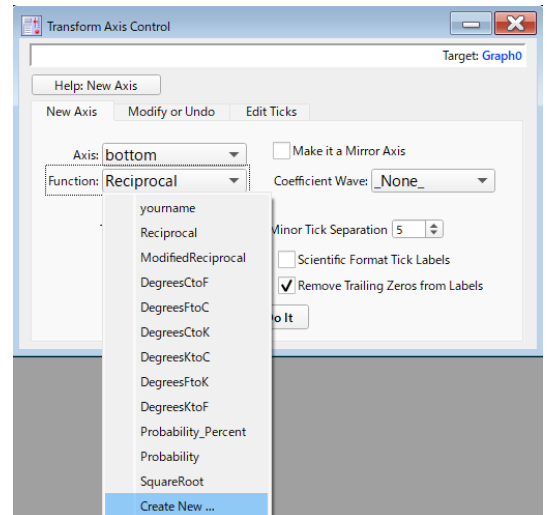
Apply ボタンをクリックするまでは、グラフに変更は反映されません。
表示結果が気に入らない場合は、Undo Last ボタンをクリックして、直近の変更を元に戻すことができます。
別の軸を選択したり、別のタブに切り替えたり、パネルを閉じたりすると、変更は確定します。
これらの操作を行い、まだグラフに適用していない変更があるとき、それらの変更を適用するかどうかを確認するメッセージが表示されます。
変更が恒久的に適用された後、変更を元に戻す唯一の方法は、Revert to Default ボタンをクリックすることです。
これにより、編集結果がすべて削除されます。

独自の変換を追加する

ユーザー独自の変換関数を定義することができます。
いくつかのルールに従う必要があります。

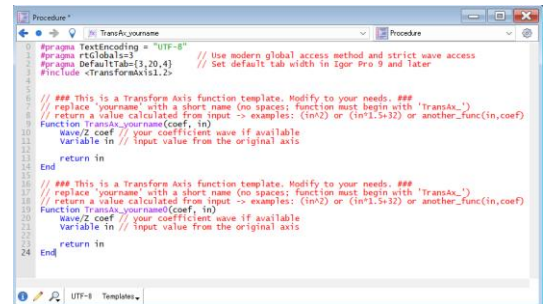
Transform Axis パネルの「New Axis」タブにある Function ドロップダウンリストの下部にある「Create New」アイテムを使うのが簡単です。

パネルを表示するには、メニュー Graph → Transform Axis → New, Modify or Undo Transform Axis を選択するか、グラフ内の空白部分を右クリックして Transform Axis → Transform Axis を選択します。



ドロップダウンメニューから Create New を選択すると、プロシージャウィンドウが表示されます。

このウィンドウには、すでにテンプレートが含まれていて、これを変更することができます。



あるいは、自分でプロシージャウィンドウを開いて、次のフォーマットで関数を作成することもできます。

```
Function TransAx_yourname (w, x)
    Wave/Z w
    Variable x
    <function body>
    return <f(x)>
end
```

yourname を自分で設定した分かりやすい関数名に置き換えてください。

関数名は、先頭に「TransAx_」を付ける必要があります。

これにより、Transform Axis パッケージが、その関数が軸変換関数として定義されていることを認識します。

「TransAx_」を名前の最初に付けないと、この関数は Transform Axis コントロールパネルの Function に表示されません。

この関数は、係数ウェーブ（上のテンプレートでは w）と入力値である変数（上では x）の2つの引数を持つ必要があります。

必ずしも、係数ウェーブは使う必要はありませんが、関数定義には含まれている必要があります。

注記： パラメーターのウェーブには、Wave/Z を使う必要があります。Igor の FUNCREF ステートメントは、同一の関数フォーマットとみなすことができるものについては厳しいです。Transform Axis パッケージは、FUNCREF を使って、変換関数を呼び出します。関数にパラメーターウェーブが必要な場合は、WaveExists() を使って確認してください。

関数内部で、入力値から変換後の値を計算する式（おそらくは1行のコード）を記述します。

変換関数に禁止された入力値の領域がある場合は、それをチェックし、該当する場合は NaN を返すべきです。Transform Axis パッケージは、これを実行すると、よりインテリジェントな処理を行うことができる可能性が高くなります。

次が、摂氏から華氏へ変換するビルトイン関数です。

```
Function TransAx_DegreesFtoC(w, x)
    Wave/Z w
    Variable x

    return (x-32)/1.8
end
```

この関数は係数ウェーブ w を使っていないことに注意してください。入力値の禁止されている領域はありません。逆関数の場合はより複雑です。

```
Function TransAx_Reciprocal(w, val)
    Wave/Z w
    Variable val

    if ( (val <= 0) )
        return NaN
    endif

    return 1/val
end
```

逆変換は、val=0 の時に特異性を持ちます。

Transform Axis パッケージは特異点を含むデータを取り扱うのが非常に難しいため、この関数はゼロ以下の値に対して NaN を返します。

負のデータに対して逆変換を使いたい場合があるかもしれません。

もしそうであれば、この関数をコピーして、名前を変更し、正の数を許可しないように if 分を変更します。

係数ウェーブを使う関数の例として、ModifiedReciprocal 関数を次に示します。

```
Function TransAx_ModifiedReciprocal(w, val)
    Wave/Z w
    Variable val

    if (!WaveExists(w) || (numpts(w) < 2))
        Abort "The ModifiedReciprocal transformation function requires a
coefficient wave with two points."
    endif

    Variable xx = (val+w[1])

    if (xx <= 0)
        return NaN
    endif

    return w[0]/xx
end
```

Abort の使用には問題があります。

これは、その関数が全く動作しない状況でのみ使うべきです。

上の場合、係数ウェーブがないことを警告するために使われます。

変換軸が作成されると、処理が早々に中断され、一度だけ警告メッセージを表示した後、変換軸の設定が破棄されま

す。
Abort を使って不正な値をレポートすると、Abort が適切なタイミングで実行されず、無限に警告が送信される可能性があります。
したがって、一部の入力値のみに影響する状況では NaN を、有効な値が一切得られない状況では Abort を使いません。

独自のプロシージャで使うことができる関数

独自の関数のコントロール下で変換軸を作成したい場合、独自のプロシージャで使うことができる4つの関数があります。
以下に列挙します。

SetupTransformTraces(theGraph, theAxis, theFunc, FuncCoefWave, tickDensity, wantMinor, minSep, TicksAtEnds [, doScientific])

SetupTransformTraces() 関数は、通常の軸を変換軸にします。
パラメーターは次のとおりです：

theGraph	変換される軸を含むグラフの名前を示す文字列です。 theGraph が "" の場合、最前面のグラフウィンドウが使われます。
theAxis	変換する軸の名前を含む文字列です。 これはおそらく left、bottom、right、top といった標準軸の1つになると思われますが、自由軸の名前になる可能性もあります。
theFunc	変換関数の名前を含む文字列です。 これは完全な名前は「TransAx_...」ですが、プリフィックス「TransAx_」で始まる必要はありません。 プリフィックスは、コントロールパネルの機能メニューにどの機能を表示するかを決定するだけです。
FuncCoefWave	変換関数に渡される係数ウェーブへの参照です。 変換関数が係数ウェーブを必要としない場合は、この部分に "\$" を使います。
tickDensity	正の数です。 Transform Axis パネルの Tick Density 設定と同じです。
wantMinor	補助目盛を表示したい場合は 1、表示したくない場合は 0 を入力します。
minSep	隣接する補助目盛の最小平均間隔（ポイント単位）です。 任意の2つの主目盛の間のすべての補助目盛の平均間隔です。
TicksAtEnds	軸の両端に目盛を追加する場合は、1 を設定します。 これは、Transform Axis パネルの Ticks At End チェックボックスと同じです。
doScientific	このオプションパラメーターを 1 に設定すると（追加の入力 doScientific=1 で関数を呼び出す）、目盛は科学記数法でラベル付けされます。

SetupTransformTraces を呼び出す前に、Packages フォルダを作成する必要があることに注意してください。

例えば、コマンドラインで次を実行しておきます：

```
NewDatafolder/O/S root:Packages
```

SetupTransformMirrorAxis(theGraph, theAxis, theFunc, FuncCoefWave, numTicks, wantMinor, minSep, TicksAtEnds)

SetupTransformMirrorAxis() 関数は、通常の軸に変換鏡像軸を追加します。
パラメーターは、SetupTransformTraces() と同じです。

TicksForTransformAxis(theGraph, theAxis, numTicks, wantMinor, minSep, MirrorAxisName, TicksAtEnds, doScientific)

TicksForTransformAxis() 関数は、変換軸の目盛の再計算を強制的に行います。
何らかの変更が反映されなかった軸をリフレッシュするために、あるいはオプションの1つを変更するために、この操作を行うことがあります。

doScientific パラメーターは、Igor Pro 6.21 で追加されました。

CloseTransformAxisGraph(theGraph, saveTransformData)

変換軸のデータを保存するかどうかの確認メッセージを回避して、変換軸を含むグラフウィンドウを閉じるには CloseTransformAxisGraph() 関数を使います。

theGraph	変換軸を含むグラフの名前を示す文字列です。 theGraph が "" の場合、最前面のグラフウィンドウが使われます。
saveTransformData	1 の場合、ウィンドウを閉じて、グラフ固有の Transform Axis データを含むデータフォルダーが残ります。 これは、それまでに再作成マクロを保存していた場合にのみ適切です。 0 の場合、グラフ固有の変換軸データを含むデータフォルダーは、グラフが閉じられると破棄されます。 DoWindow/R を使って再作成マクロを保存することはできますが、 DoWindow/R を関数から呼び出すことはできません。 したがって、これを実行する場合は、DoWindow/R と CloseTransformAxisGraph() の両方を Execute/P を介して実行する必要があります。

Transform Axis の以前のバージョンからのグラフ

新しいバージョンの Transform Axis は、古いバージョンで作成されたグラフとは互換性がないため、新しいバージョンを使って作成されたグラフは、古いバージョンの Transform Axis では操作できません。
以前に、Transform Axis パッケージを使ったことがある場合、Transform Axis を使うすべての Experiment ファイルの Procedure Window に次の行があるはずで

```
#include <TransformAxis>
```

この行は、メニュー Graph → Packages → Transform Axes を選択したときに、Procedure Window に追加されます。

この行により、Igor は古いバージョンの Transform Axis パッケージを読み込みます。
その場合は、グラフは以前と同じように動作し、Transform Axis プロシージャファイルの旧バージョンを使います。

新しいバージョンで、メニュー Graph → Packages → Transform Axes を選択すると、新しいバージョンの Transform Axis パッケージを読み込みます。
これは、次の行を Procedure Window に追加します：

```
#include <TransformAxis1.2>
```

しかし、すでに古いバージョンの Transform Axis がロードされている場合、「You already have the older version of Transform Axis loaded. (古いバージョンの Transform Axis がすでにロードされています)」という警

告が表示されます。

この場合、新しい Transform Axis はロードされません。

Transform Axis バージョン 1.2 の新機能

- 変換軸が、グラフへのトレースの追加やグラフからのトレースの削除を正しく処理するようになりました。
以前は、新しいトレースを追加するときに変換軸を解除する必要がありました。
- 変換鏡像軸が主軸に行われたスタイル変更に従うようになりました。
鏡像軸のスタイルに変更を加えても、主軸で変更しない限り、元の変更が維持されます。
以前は、変換鏡像軸は最初に作成するときには一致するスタイルで作成されますが、その後主軸に変更を加えても更新されませんでした。
- 変換鏡像軸の主軸の範囲を変更すると、鏡像軸が更新されるようになりました。
以前は、鏡像軸は更新されず、元の状態のままになっていました。
- 変換鏡像軸の主軸の長さを変更すると、鏡像軸の長さも同じように変更されるようになりました。
これは、変換鏡像軸を含む積み重ねグラフを作るときに便利です。
- 変換軸を使ったグラフでウェーブのデータまたはデータポイントの数を変更すると、変換軸も更新されるようになりました。
以前は、データの変更により軸が更新されても、データポイントの数が変更されても反映されませんでした。
- 独自のコードで Transform Axis を使っているユーザーのために、Transform Axis を含むグラフを閉じるための関数が実装されました：

```
CloseTransformAxisGraph(theGraphName, saveTransformData)
```