

CONTENTS

ビジュアルヘルプ – Function Grapher	2
Function Grapher ガイドツアー.....	2
Function Grapher リファレンス.....	6
関数の作成	6
Function Grapher ウィンドウは単なるグラフウィンドウ.....	6
関数はどこにありますか？	6
関数はファイルとして保存される.....	7
Function Grapher はどのように関数をロードするか.....	8
Save Function ボタン.....	8
Kill Function ボタン.....	9
Save Coefs to Wave ボタン.....	9
Load Coefs From Wave ボタン.....	9
Function Grapher で作られたものではない関数	10
Function Grapher の関数はフィッティング関数	10
フィッティング関数は Function Grapher に適している	10
データはどこにありますか？	11

ビジュアルヘルプ – Function Grapher

メニュー Analysis → Packages → Function Grapher

このメニューを選択すると、次のメニュー項目が追加されます。

メニュー Analysis → Function Grapher

Function Grapher は、数学的表現を簡単にグラフ化できるように設計された Igor プロシージャパッケージです。

これは、式の中の定数の変更による影響、独立変数の範囲、式のグラフ化に使われるポイントの密度などを簡単に探索できるコントロール付きのグラフで構成されています。

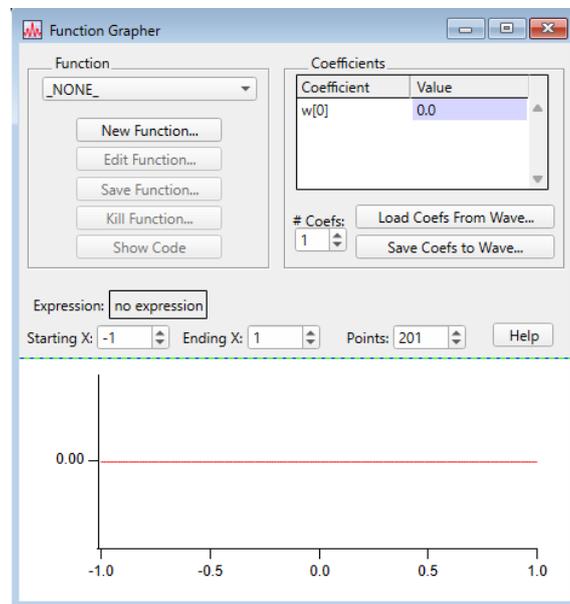
補助コントロールパネルに式を入力します。

このパネルでは、式の中の定数に記憶しやすい名前を付けたり、参照しやすいように式に名前を付けることができます。

Function Grapher は、Igor のユーザー定義関数に、入力した式をラップすることで動作します。

これはユーザーに代わって行われます。

Function Grapher により、ユーザー定義関数として式を実装する簡単な Igor コードが作成されます。



グラフ化する数式を入力できることに加え、必要な関数形式に一致する任意のユーザー定義関数または外部関数を選択することができます（外部関数は、XOP によって定義された関数です。ヘルプ Igor Extensions (Using Igor.ihf) を参照）。

Function Grapher を使うと、これらの関数も視覚化して調べることができます。

ただし、必要な係数の数と、それらが何を意味するのかを知っておく必要があります。

Function Grapher ガイドツアー

Function Grapher の紹介のために、Sinc 関数 $\sin(x)/x$ を使います。

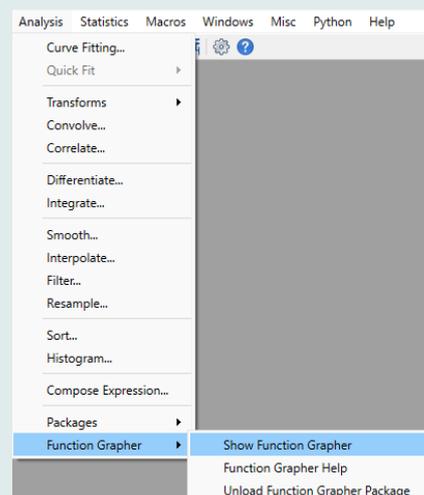
定義により、 x がゼロのとき、 $\text{sinc}(x)$ は 1.0 の極限值を返します。

1. Function Grapher パッケージをロードするには、メニュー Analysis → Packages → Function Grapher を選択します。

これにより、メインの Function Grapher ウィンドウが表示されます。

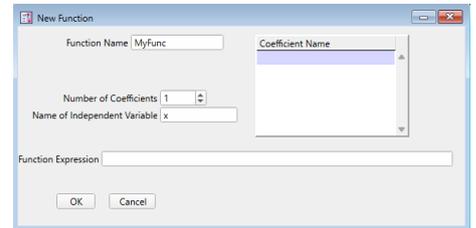
パッケージがすでにロードされている場合は、この操作は必要ありません。

Function Grapher ウィンドウが表示されていない場合は、メニュー Analysis → Function Grapher → Show Function Grapher を選択してください。



2. New Function ボタンをクリックします。

New Function 補助コントロールパネルが表示されます。



3. Function Name と書かれたボックスに「MySinc」（クオートなし）と入力します。

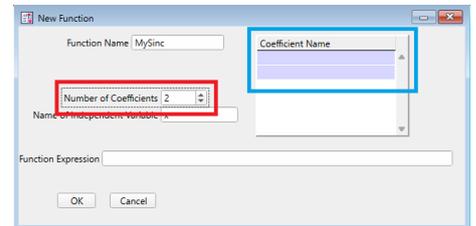


4. Number of Coefficients と書かれたボックスに「2」と入力します。

これにより、Coefficient Name リストに 2 つの空白セルが表示されます。

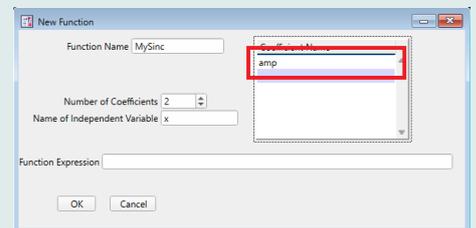
係数の数をゼロに設定することで、係数ゼロの関数を作成することができます。

関数内の定数を変更するには、関数を編集する必要があります。



5. Coefficient Name リストの一番上のセルをクリックして編集を開始します。

セルに「amp」と入力します。



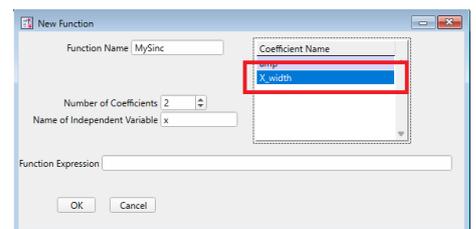
6. 下矢印キーを押して次のセルに移動します。

「X_width」と入力します。

Enter キーを押して編集を終了します。

アンダースコアに注意してください。

係数名は、非リベラルな名前であればなりません。



7. Function Expression ボックスに次の行を入力します。

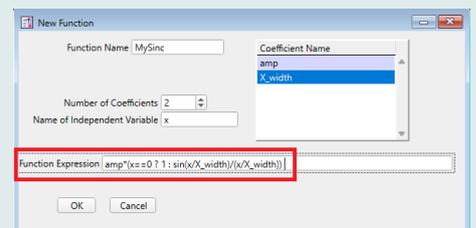
$$\text{amp} * (x == 0 ? 1 : \sin(x/X_width) / (x/X_width))$$

?: 演算子は、テストの結果に応じて 2 つの値のうちの 1 つを選択します。

ここでは、x がゼロであるかどうかを確認するために使われています。

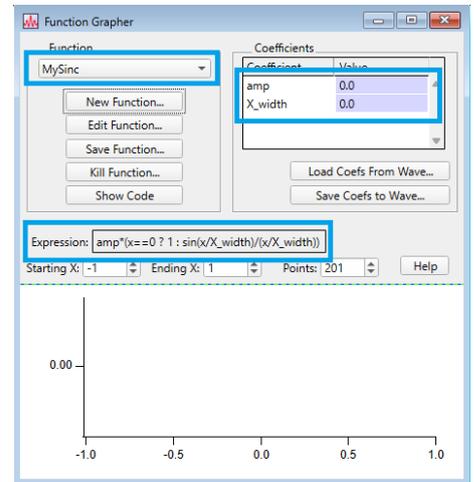
もしゼロであれば、ゼロによる除算となるため、式は評価できません。

代わりに、関数の限界値である 1.0 が代入されます。

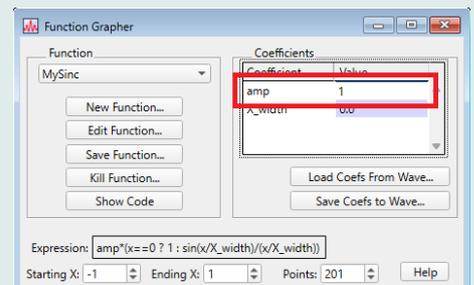


8. OK ボタンをクリックします。

New Function ウィンドウが閉じ、Function メニューで新しい関数が選択された状態で Function Grapher ウィンドウに戻ります。しかし、amp と X_width の値が 0.0 であるため、グラフ化されていない可能性が高いです。

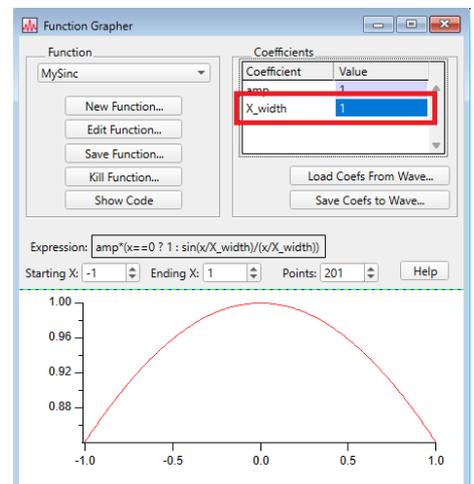


9. Coefficients の Value 列の一番上のセルをクリックして、amp 係数の編集を開始します。セルに「1」と入力します。



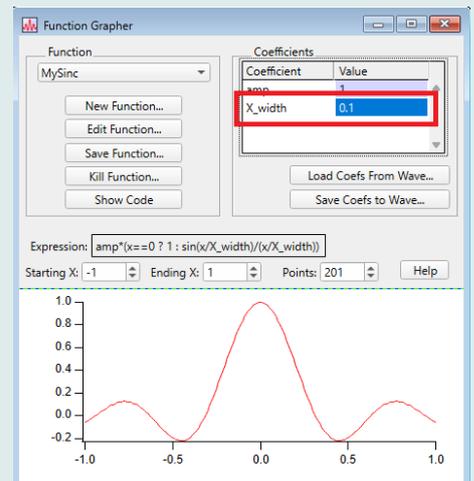
10. 下矢印キーを押して、X_width セルに移動します。そこにも「1」を入力します。Enter キーを押して、新しい値を受け入れます。

右図のようなグラフが表示されているはずですが。



11. Coefficients リストの X_width 値セルをクリックし、0.1 を入力して Enter キーを押します。

グラフは右図のように更新されます。

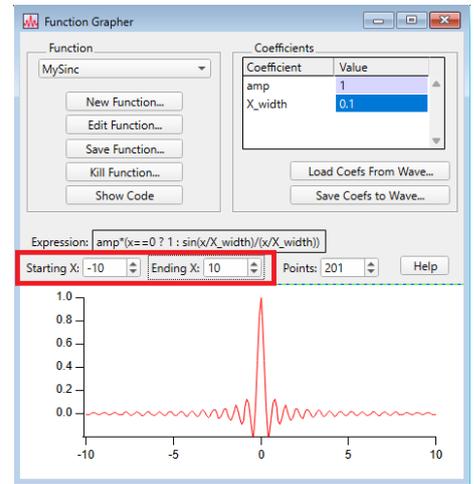


12. Starting X ボックスをクリックし、-10 と入力します（または、-10 と表示されるまで下矢印キーを押します）。

Tab キーを押して Ending X のボックスに移動し、10 を入力します。

Enter キーを押して新しい値を受け入れます。

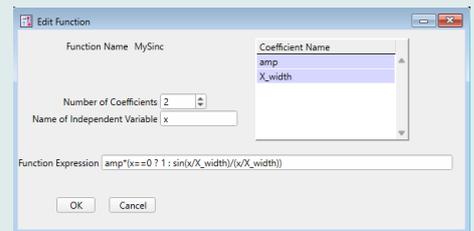
グラフは右図のような表示に更新されます。



13. 次に、ピークの水平位置を設定するために、3 番目の係数を追加します。

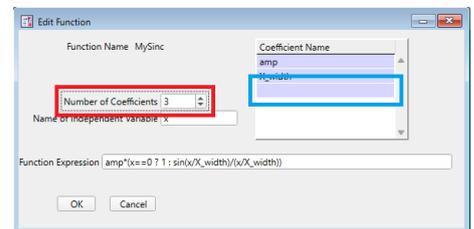
Edit Function ボタンをクリックします。

Edit Function 補助コントロールパネルが開きます（Edit Function パネルに「MySinc」関数の情報がすでに入力されています）。



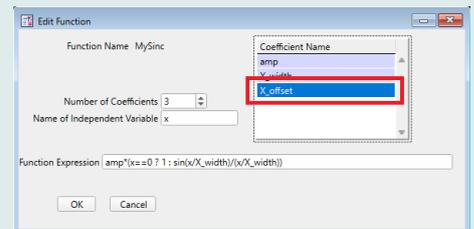
14. 係数の数を 3 に変更します。

Coefficient Name リストに新しい行が追加され、新しい係数を追加できるようになります。



15. 新しい行をクリックして編集を開始し、「X_offset」と入力します。

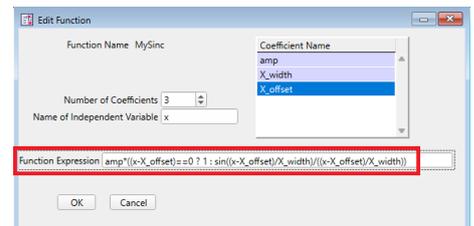
Enter キーを押して編集を終了します。



16. Expression ボックス内の x をすべて (x-X_offset) に変更します。

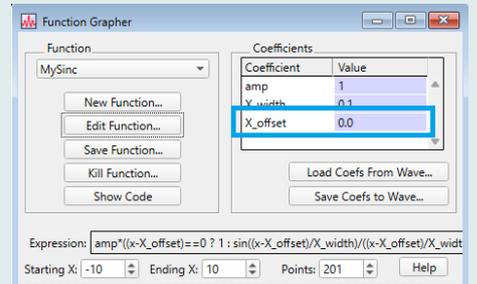
結果は次のようになります。

```
amp*((x-X_offset)==0 ? 1 : sin((x-X_offset)/X_width)/((x-X_offset)/X_width))
```



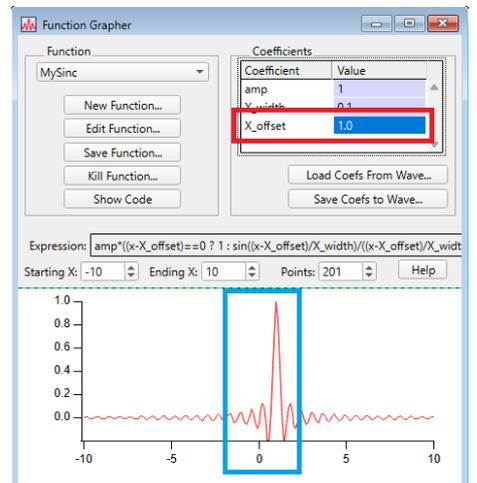
17. OK をクリックして、Function Grapher ウィンドウに戻ります。

Coefficients リストに新しい行が追加されました。



18. 新しい行をクリックし、1.0 と入力します。

グラフの軌跡が移動し、ピークが X=1 の位置に移動します。



Function Grapher リファレンス

関数の作成

Function Grapher で使う関数の作成と編集の手順については、「Function Grapher ガイドツアー」を参照してください。

Function Grapher ウィンドウは単なるグラフウィンドウ

Function Grapher メインウィンドウは、単なるグラフウィンドウです。いくつかの追加機能がありますが、基本的にはグラフのみです。

つまり：

印刷が可能です。コントロール用のグレーの領域を除いたグラフ部分のみが印刷されます。

グラフの外観は、通常の Igor の方法で変更できます。

つまり、Modify Axis ダイアログを使って軸スタイルを変更したり、トレースをダブルクリックして Modify Trace Appearance ダイアログを表示してトレースの外観を変更したりすることができます。

Function Grapher ウィンドウにトレースを追加することは可能ですが、お勧めできません。追加すると、Function Grapher の動作に予期せぬ問題が生じる可能性があります。

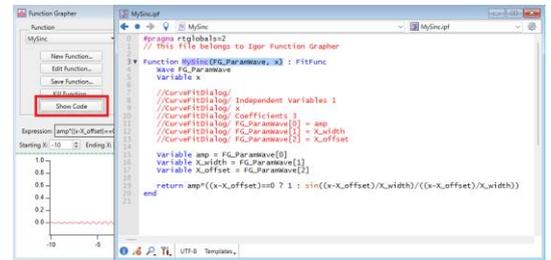
関数はどこにありますか？

Function Grapher は、Igor のユーザー定義関数でユーザーの式をラップします。

Function Grapher で作成された各関数は、個別のプロシージャウィンドウに入ります。

Function Grapher が作成したコードは、Show Code ボタンをクリックすると見ることができます。

Function Grapher ガイドツアーで作成した関数のコードは次のようになります。



```
#pragma rtglobals=2
// This file belongs to Igor Function Grapher

Function MySinc(FG_ParamWave, x) : FitFunc
    Wave FG_ParamWave
    Variable x

    //CurveFitDialog/
    //CurveFitDialog/ Independent Variables 1
    //CurveFitDialog/ x
    //CurveFitDialog/ Coefficients 3
    //CurveFitDialog/ FG_ParamWave[0] = amp
    //CurveFitDialog/ FG_ParamWave[1] = X_width
    //CurveFitDialog/ FG_ParamWave[2] = X_offset

    Variable amp = FG_ParamWave[0]
    Variable X_width = FG_ParamWave[1]
    Variable X_offset = FG_ParamWave[2]

    return amp*((x-X_offset)==0 ? 1 : sin((x-X_offset)/X_width)/((x-X_offset)/X_width))
end
```

「This file belongs to...」というコメントにより、このプロシージャウィンドウは Function Grapher に属することが示されます。

これを削除すると、一部の機能が動作しなくなります。

「//CurveFitDialog/」で始まるコメントには、係数の名前と係数の数に関するコード情報が含まれています。これらのコメントを変更すると、機能を使う時に不便になる場合があります。

これらのコメントは、Curve Fit ダイアログの New Fit Function サブダイアログで作成されたコメントの一部であることに注意してください。

実際、カーブフィッティングに Function Grapher を使うことができます。

関数はファイルとして保存される

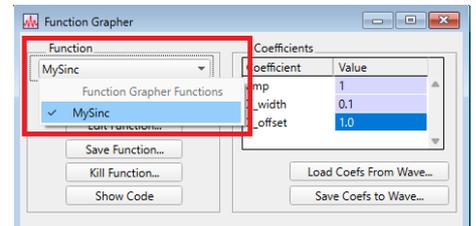
Function Grapher は、各関数を個別のプロシージャウィンドウに配置するだけでなく、そのプロシージャウィンドウをファイルに保存します。

そのファイルは、Igor がプロシージャパッケージのプレファレンスファイル用に確保しているハードディスク上の特別なフォルダーに保存されています。

これにより、Function Grapher を Igor Experiment で実行する時に、作成したすべての関数にいつでもアクセスできるようになります。

メインの Function Grapher ウィンドウの Function メニューには、特別なフォルダーで見つかったすべての関数がリスト表示されます。

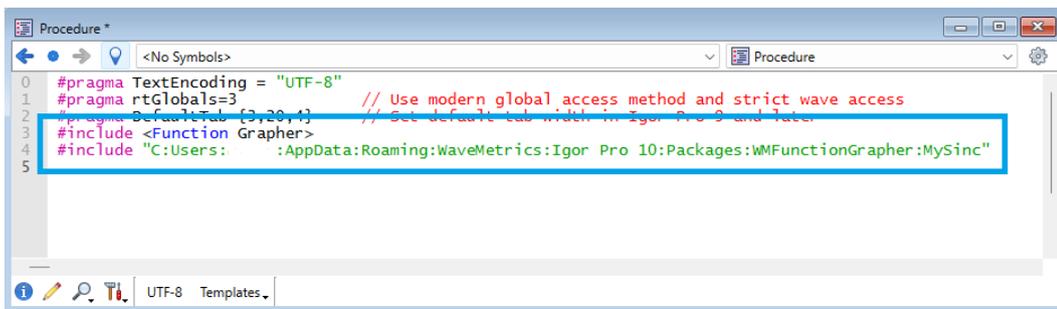
まだ読み込まれていない場合、メニューから選択すると Function Grapher が関数を読み込みます。



Function Grapher はどのように関数をロードするか

Function Grapher を使って新しい関数を作成したり、メニューから既存の関数を選択すると、Function Grapher が #include 行をメインのプロシージャウィンドウに追加します。

この #include 行は、Igor コンパイラに特定のプロシージャファイルをロードするように指示します (#include 行についての詳細はヘルプ The Include Statement (Programming Techniques.ihf) を参照)。



上記の Function Grapher ガイドツアーで関数を作った時、上図の #include 行がコンピューター上に作成されました。

この行の内容は、使っているオペレーティングシステムとハードドライブの詳細によって異なります。この行を使って、ファイルが保存されているフォルダーを見つけることができます。

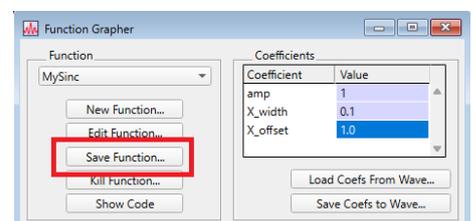
Save Function ボタン

Function Grapher が関数を保存するハードドライブ上の場所が不明瞭なため、別の場所に移動することが望ましい場合があります。

それを単に見つけて移動するだけでは、うまくいかない場合があります。

代わりに、Save Function ボタンを使うことができます。

このボタンをクリックすると、関数の保存先を変更するためのダイアログボックスが表示されます。



Igor Procedures フォルダーと User Procedures フォルダーの 2 つの場所は特別です。

どちらに保存されているプロシージャファイルも、アクセスが特に便利です。

Igor Procedures フォルダーに保存されているすべてのプロシージャファイルは、Igor が起動するたびに自動的にロードされます。

ここに関数を保存すると、関数を利用可能にするために Function Grapher を実行する必要がなくなります。

User Procedures フォルダーに保存されたプロシージャは、#include 行を使って簡単にアクセスできます。

実際には、これらの場所に関数ファイルを置くことは推奨していません。

Igor Pro フォルダーの外側に、任意のフォルダーにファイルを置くことをお勧めします。

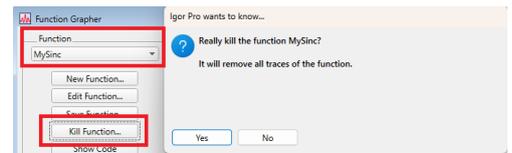
次に、このフォルダーへのショートカットを作成し、ショートカットを Igor Procedure フォルダーまたは User Procedure フォルダーに配置します。

Kill Function ボタン

Function Grapher を頻繁に使っていると、かなりの数の関数が蓄積されることに気づくかもしれません。それらの関数の一部は、一時的な興味でしかなく、今すぐにでも削除したいと思っているかもしれません。

関数を Function メニューで選択して、Kill Function ボタンをクリックします。

変更するかどうかを確認するアラートが表示されます。



注：もし、関数を削除する場合は、関数の定義が含まれているファイルがハードドライブから削除されます。この操作を元に戻すことはできません。

複数の Experiment ファイルでこの関数を使い、それらのファイルを保存した場合、存在しないファイルを参照する #include 行が保存されていることとなります。

開いていない Experiment ファイルは、そのような参照を見つけることはできません。

次にファイルを開くと、コンパイルエラーが発生します。「Include file not found/インクルードファイルが見つかりません」というエラーです。

エラーメッセージを含む警告ボックスには、問題のある行も表示されます。

アラート内の行を選択し、削除して、Retry ボタンをクリックすることができます。

Save Coefs to Wave ボタン

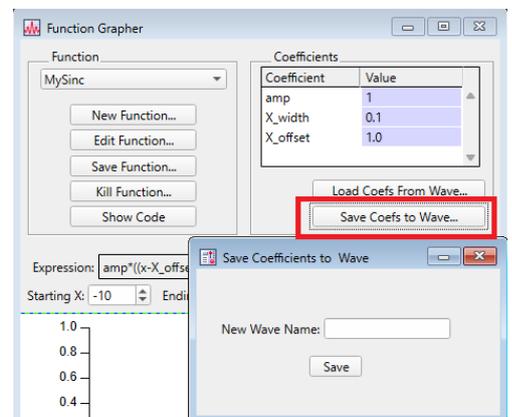
さまざまな関数を作成したり、興味のある係数のセットがいくつかある場合、後で取り出せるように係数のセットを保存しておきたいと思うかもしれません。

これを行うには、係数値リストの下にある Save Coefs to Wave ボタンをクリックします。

ウェーブ名を入力するボックス付きの小さなコントロールパネルが作成されます。

Save ボタンをクリックすると、現在のデータフォルダーにウェーブが作成されます。

同名のウェーブがすでに存在する場合は、上書きされます。



Load Coefs From Wave ボタン

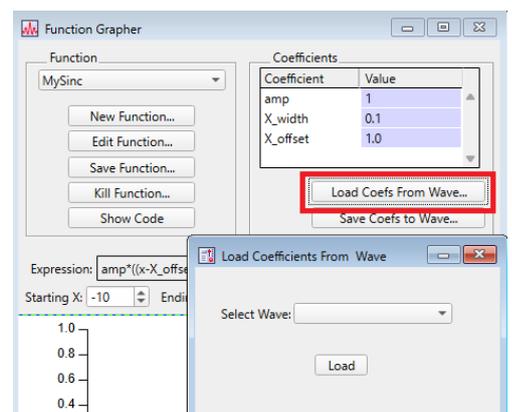
ウェーブに保存された係数セットを取得するには、Load Coefs From Wave ボタンをクリックします。

これにより、現在のデータフォルダー内の適切なウェーブのメニューを持つ小さなコントロールパネルが作成されます。

希望するウェーブを選択します。

Load ボタンをクリックすると、選択したウェーブから係数値が係数値リストに読み込まれます。

グラフも更新され、変更された係数が反映されます。



Function Grapher で作られたものではない関数

Function Grapher のメインウィンドウの Function メニューには、Function Grapher で作られていない関数も含まれます。

適切なフォーマットの関数はすべて表示されます。

適切なフォーマットとは、数値変数パラメーターを 1 つだけ取る関数、またはウェーブと数値変数の 2 つのパラメーターを取る関数を意味します。

関数は数値を返さなければなりません。

Function Grapher ではない関数には、Function Grapher に係数の数と名前を伝える特別なコメントが欠けているため、その情報を提供する必要があります。

その機能を選択すると、係数値リストの下に小さなボックスが表示され、係数の数を入力できるようになります。

このルールに対する例外は、Curve Fit ダイアログで作成されたフィッティング関数です。

これらの関数には、Function Grapher に必要なコメントが付いています。

Function Grapher で作成されていない関数は、Function Grapher で編集、保存、Kill できません。

そのため、これらの関数のいずれかを選択すると、それらのボタンは使用できなくなります。

外部関数、または XFUNCS (Igor Extensions を参照) も、正しいパラメーターの種類であれば、Function Grapher で使うことができます。

Function Grapher で作られていないユーザー定義関数と同様に、編集、削除、保存ができず、Function Grapher は係数の数や名前を認識できません。

XFUNC のコードは Igor のコードではないため、コードを表示することはできません。

代わりに、Show Code ボタンは Function Help ボタンになります。

多くの XFUNC には、関数と必要な係数を説明するヘルプファイルが付属していますが、すべてではありません。

ボタンをクリックすると、Igor は Igor Pro フォルダ内で、関数名と同じヘルプトピックを含むヘルプファイルを検索します。

見つからない場合は、その旨を伝えるアラートが表示されます。

Function Grapher の関数はフィッティング関数

Function Grapher を使って、少なくとも 1 つの係数を持つ関数を作成すると、その結果はカーブフィッティングに適した関数となります。

Function Grapher が作成するコメントは、Curve Fitting ダイアログで作成されるコメントと同じであるため、Curve Fit ダイアログに係数名が表示されます。

Function Grapher はプロシージャファイルを読み込むため、書き込み可能にしない限り、Curve Fit ダイアログで Function Grapher 関数を編集することはできません。

これを行うには、Show Code ボタンをクリックして、関数のコードを含むプロシージャウィンドウを表示します。ウィンドウの一番下に、鉛筆のアイコンに斜線が引かれたものがあります。これは、読み取り専用ウィンドウであることを示しています。

アイコンをクリックすると書き込み可能になります。

フィッティング関数は Function Grapher に適している

Function Grapher は Curve Fit ダイアログと同じコメントを使うため、Curve Fit ダイアログでフィッティング関数を作ると、Function Grapher を使ってその関数を表示することができます。

ほとんどの Function Grapher の機能は正しく動作します。

しかし、Function Grapher が関数を持っていないため、これらの関数を編集、保存、Kill することはできません。

データはどこにありますか？

Function Grapher は、データを特別なデータフォルダー（root:Packages:WM_GraphFunction）に保存します。このフォルダー内のデータには、Data Browser（Data メニュー）を使ってアクセスできます。このデータを変更または削除しないでください。

2つの項目が参考になるかもしれません。

FuncWave と呼ばれる関数をグラフ化するために使われたウェーブはここに保存されています。

ウェーブをデータフォルダーから移動させないでください。

移動させると、Function Grapher の機能に問題が生じる可能性があります。

ただし、FuncWave のコピーを作成し、好きな場所に移動させることは可能です。

特定の関数と係数の組み合わせで独自のグラフを作成したい場合は、これを行うことができます。

もう1つ、係数ウェーブ、FG_CoefWave も必要かもしれません。

自分で関数を呼び出したい場合には、役に立つかもしれません。

繰り返しになりますが、ウェーブはコピーを作成し、移動しないでください。

コピーを作る最善の方法は、「Save to Wave」ボタンを使うことです。