

# CONTENTS

<b>Volume III User's Guide: Part 2 (III-8 Curve Fitting #1)</b> .....	2
概要 .....	2
カーブフィッティングの用語 .....	4
カーブフィッティングの概要 .....	5
反復フィッティング .....	5
初期推定 .....	6
終了基準 .....	6
カーブフィッティングでのエラー .....	6
カーブフィッティングのデータ .....	7
Quick Fit メニューを使ったカーブフィッティング .....	8
Quick Fit メニューの制限 .....	9
Curve Fitting ダイアログの使い方 .....	10
シンプルなケース：組み込み関数を使ってフィッティング (Line Fit) .....	10
関数とデータの選択 .....	12
2つの便利な追加機能：係数の保持と残差の生成 .....	13
自動推定が動作しない場合 .....	15
定数でフィッティング .....	17
ユーザー定義関数でフィッティング .....	18
関数の作成 .....	18
ユーザー定義関数の Coefficients タブ .....	21
ユーザー定義関数が常に使えるようにする .....	23
ユーザー定義関数を削除する .....	24
ユーザー定義フィッティング関数の詳細 .....	24

# Volume III User's Guide: Part 2 (III-8 Curve Fitting #1)

## 概要

Igor Pro マニュアル : III-178 ページ以降をもとに編集

Igor のカーブフィッティング機能は、最も優れた分析機能の 1 つです。  
そのハイライトをいくつか紹介します。

- 線形および一般的な非線形カーブフィッティング
- 最小二乗法または変数誤差モデルにおける最小直交距離によるフィッティング
- 暗黙的なモデルへのフィッティング
- 一般的なフィッティング用の組み込み関数
- 組み込み関数に対する自動初期値推定
- ユーザー定義の複雑な関数へのフィッティング
- グリッドデータまたは複数列データの任意の独立変数の関数へのフィッティング
- 複数のフィッティング関数へのフィッティング
- ウェーブフォームまたは XY ペアの一部のフィッティング
- エラーの推定値を生成
- 重み付けのサポート

カーブフィッティングの考え方は、データに適合する数学モデルを見つけることです。

特定の形式の関数を選択するときには、理論的な理由があると思います。

カーブフィッティングは、その関数がデータにできるだけ一致するように、特定の係数を求めます。

何千もの関数のうち、データセットに適合するものをカーブフィッティングで探すことはできません。

また、カーブフィッティングを単にデータに滑らかな曲線を表示する目的で使うこともあります。

これは時々うまくいきますが、平滑化や補間も検討すべきで、これについてはマニュアル III-7 Analysis で説明しています。

Igor では 3 種類の関数にフィッティングすることができます :

- ビルトイン関数
- ユーザー定義関数
- 外部機能 (XFUNC)

組み込みのフィッティング関数は、線形、多項式、正弦、指数、二重指数、ガウス、ローレンツ、ヒル方程式、シグモイド、ロジスティック、対数、Gauss2D (2 次元ガウスピーク)、Poly2D (2 次元多項式) です。

ユーザー定義関数は、New Fit Function ダイアログにその関数を入力することで作成します。

非常に複雑な関数は、Procedure ウィンドウに入力する必要があるかもしれません。

外部関数である XFUNC は、C または C++ で記述されます。

XFUNC を作成するには、オプションの Igor XOP Toolkit と C/C++ コンパイラが必要です。

WaveMetrics または他のユーザーから入手した XFUNC を使う時には、ツールキットは必要ありません。

カーブフィッティングは次のような形式の方程式で機能します :  $y = f(x_1, x_2, \dots, x_n)$ 。独立変数 ( $x_n$ ) の数がいくつであっても関数をフィットさせることは可能ですが、ほとんどのケースは 1 つだけです。

多変量フィッティングの詳細については、マニュアル III-200 Fitting to a Multivariate Function を参照してください。

また、暗黙的関数へのフィッティングも可能です。暗黙的関数は次の形式です： $f(x_1, x_2, \dots, x_n)$ 。  
詳細は、マニュアル III-242 Fitting Implicit Functions を参照してください。

線形制約付きのカーブフィッティングを行うこともできます（マニュアル III-227 Fitting with Constraints を参照）。

## カーブフィッティングの用語

組み込みのフィッティングは、CurveFit コマンドによって実行されます。

ユーザー定義のフィッティングは、FuncFit または FuncFitMD コマンドによって実行されます。

「カーブフィットコマンド」という言葉は、CurveFit、FuncFit、FuncFitMD のいずれか適切なものを指します。

外部関数へのフィッティングは、ユーザー定義関数へのフィッティングと同じように動作します（ただし、Curve Fitting ダイアログに関するいくつかの注意点があります）。

マニュアル III-194 Fitting to an External Function (XFUNC) を参照してください。

Curve Fitting ダイアログを使う場合は、組み込み関数とユーザー定義関数の区別について詳しく知る必要はありません。

外部関数と、その他のタイプの区別については、少し知っておく必要があるかもしれません。

カーブに当てはめる数値を「係数」と呼びます。

コマンドや関数に渡す値には「パラメーター」という用語を使います。

## カーブフィッティングの概要

カーブフィッティングでは、未知の係数を持つ関数と生データがあります。係数の値を求め、その関数が生データにできるだけ一致するようにしたいわけです。係数の「最良の」値は、カイ二乗値を最小化する値です。カイ二乗値は次の式で定義されます：

$$\sum_i \left( \frac{y - y_i}{\sigma_i} \right)^2$$

ここで、 $y$  は与えられたポイントの適合値、 $y_i$  はそのポイントの測定データ値、 $\sigma_i$  は  $y_i$  の標準偏差の推定値です。もっとも単純なケースは、直線 ( $y = ax + b$ ) にフィッティングするものです。仮に、データが直線状に並ぶはずだという理論的な理由があるとします。その場合は、データにもっとも適合する係数  $a$  と  $b$  を見つけることとなります。

直線または多項式関数については、1 ステップで最適な係数を見つけることができます。これは非反復カーブフィッティングであり、多項式フィッティングに特異値分解アルゴリズムを使っています。

## 反復フィッティング

その他の組み込みのフィッティング関数およびユーザー定義関数については、フィッティングが未知の係数をさまざまな値を試すため、処理は反復的になります。試行ごとに、カイ二乗値を計算し、カイ二乗値の最小値を与える係数の値を探索します。

カイ二乗値を最小化する係数の値を探索するために、レベンバーグ・マルカート (Levenberg-Marquardt) アルゴリズムが使われます。これは非線形最小二乗近似の一種です。

適合が進み、より良い値が見つかり、カイ二乗値は減少していきます。カイ二乗値の減少率が十分に小さくなった時点で、フィッティングは終了します。

反復カーブフィッティング中は、カーブフィッティングの進捗状況を示すウィンドウが表示されます。これは、適用された関数、更新された係数の値、カイ二乗値、パス数を示します。通常、Quit ボタンが無効になり、OK ボタンが有効になるまで、フィッティング処理が行われます。OK をクリックすると、フィッティングの結果が履歴領域に表示されます。

フィッティングが十分に進み、満足のいく結果が得られた場合は、Quit ボタンをクリックすると、進行中の反復が終了し、フィッティングが自動的に完了したかのように結果が履歴領域に表示されます。

カイ二乗値が減少しない場合や、係数の一部が非常に大きな無意味な値をとるなど、フィッティングがうまくいっていないことがわかる場合もあります。

User Abort Key Combinations のキーの組み合わせを押すと、フィッティングの結果を破棄してキャンセルできます。

この場合、フィッティング係数を調整して、再度試す必要があります。

## 初期推定

カイ二乗の最小値を探索するために、レベンバーグ・マルカート・アルゴリズムが使われます。

カイ二乗は、多次元誤差空間における曲面を定義します。

探索プロセスでは、最初に係数値の推定から開始します。

初期推定値から開始し、カイ二乗サーフェス上の出発点から丘を下るように移動しながら、最小値を探索します。

カイ二乗サーフェスで最も深い谷を見つけるのが目標です。

これは、実験データとフィッティングデータの差を最小二乗法の観点で最小化するフィッティング関数の係数の値のサーフェス上のポイントです。

適合関数のうち、谷が1つしかないものもあります。

この場合、谷底が見つかった時点で、最もフィットする値が得られたこととなります。

しかし、一部の関数では、複数の谷（周囲の値よりもフィッティングが良好な場所）が存在する可能性があるため、それが最適なフィットであるとは限りません。

フィッティング処理が谷底を見つけると、たとえサーフェスの他の場所にさらに深い谷があるとしても、フィッティングが完了したと判断します。

どの谷が最初に見つかるかは、初期推定によって決まります。

組み込みのフィッティング関数については、初期推定値を自動的に設定することができます。

これで満足のいく結果が得られない場合は、ユーザーが推定して手動で設定することになります。

ユーザー定義関数にフィットさせるには、手動で推定値を入力する必要があります。

## 終了基準

カーブフィッティングは、最適なフィットを探索するための40回のパスを行った後に終了しますが、9回連続でカイ二乗値の減少がみられない場合は、その時点で終了します。

これは、初期推定が非常に正確で、最小のカイ二乗値からフィッティングが始まる場合に起こりえます。

初期推定が大幅に外れている場合や、関数がデータに全くフィットしない場合にも、このようなことが起こります。

40回のパスの条件は変更できます。

マニュアル III-232 Special Variables for Curve Fitting の V\_FitMaxIters の説明を参照してください。

通常、40回以上のパスが必要な場合は、フィッティングに問題があることを示しています。

マニュアル III-226 Identifiability Problems を参照してください。

フィッティング関数とデータについてよく知らない限り、結果が良いものであると仮定するのは賢明ではありません。

ほとんどの場合、結果とデータを比較するために、結果のグラフを表示したいと思うでしょう。

また、残差、つまりフィッティングされたモデルとデータの差のグラフも見ることを勧めます。

Igor では、ほとんどの場合、どちらも簡単に実行できます。

## カーブフィッティングでのエラー

場合によっては、カイ二乗最小値を求めるために次にどこへ進むかを決定できない状況に遭遇することがあります。

これは「特異行列」エラーの原因となります。

これについては、マニュアル III-265 Singularities in Curve Fitting で説明しています。  
マニュアル III-266 Curve Fitting Troubleshooting を参照すると、問題の解決策を見つけることができます。

## カーブフィッティングのデータ

従属変数（通常「y」と呼ばれる）と独立変数（通常「x」と呼ばれる）の両方の値を測定している必要があります（特に、独立変数が1つしかない場合）。

これらは「応答変数」や「説明変数」と呼ばれることがあります。

ウェーブフォームデータまたはXY データに対してカーブフィッティングを行うことができます。

つまり、1つのウェーブに含まれるデータをフィッティングすることができ、そのウェーブのデータ値はY データを表し、X スケーリングは等間隔のX データを表します。

または、あるウェーブのデータ値がY 値を表し、別のウェーブのデータ値がX 値を表すような、2つの（またはそれ以上の）ウェーブのデータをフィッティングすることもできます。

この場合、データは等間隔である必要はありません。

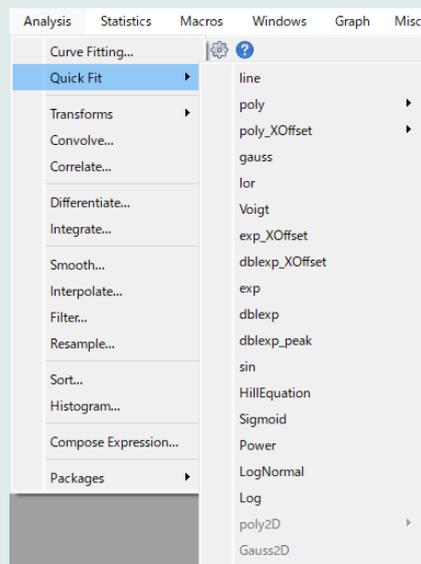
実際、X データはランダムな順序でも構いません。

ウェーブフォームとXY データの詳細については、マニュアル II-5 Waves を参照してください。

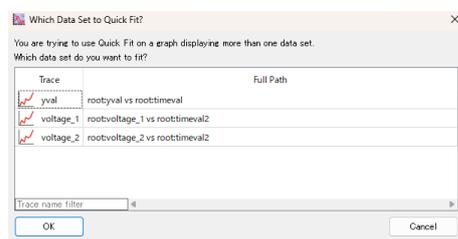
## Quick Fit メニューを使ったカーブフィッティング

カーブメニュー Analysis → Quick Fit が、カーブフィッティングを行うもっとも簡単な方法です。

Quick Fit メニューでは、ビルトインのフィッティング関数を使って、カーブフィッティングに素早くアクセスできます。フィッティングするデータは、最前面のグラフを調べることで決定されます。

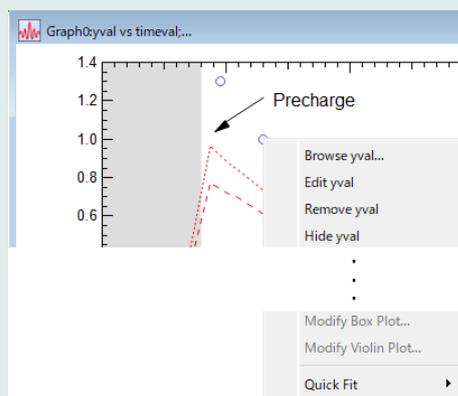


1つのトレースだけが見つかった場合、グラフ化されたデータは選択されたフィッティング関数でフィッティングされます。グラフに複数のトレースが含まれる場合、どのトレースをフィッティングするかを選択するダイアログが表示されます。



グラフのコンテキストメニューからも Quick Fit メニューにアクセスできます。

グラフのトレースを右クリックすると、コンテキストメニューに下のほうに Quick Fit の項目があります。このようにして Quick Fit メニューにアクセスすると、クリックしたトレースを自動的にフィッティングします。これにより、グラフ上に複数のトレースがある場合に、どのトレースに Quick Fit を使うのかを指定するダイアログを回避することができます。



Quick Fit メニューを使うと、フィッティングを実行し、モデルカーブをグラフに自動的に追加するコマンドが生成されます。

デフォルトでは、グラフカーソルが存在する場合、カーソル間のデータのみがフィッティングされます。

Quick Fit メニューの Fit Between Cursors を選択して、そのチェックを外すことで、データセット全体にフィッティングすることができます。

チェックされていない場合、グラフのカーソルを無視してフィッティングが実行されます。

フィッティングするトレースにエラーバーがあり、エラーバーのデータがウェーブから取得されている場合、Quick Fit はフィッティングの重みづけウェーブとしてそのウェーブを使います。

これは、エラーバーが1標準偏差を表していることを前提としていることに注意してください。

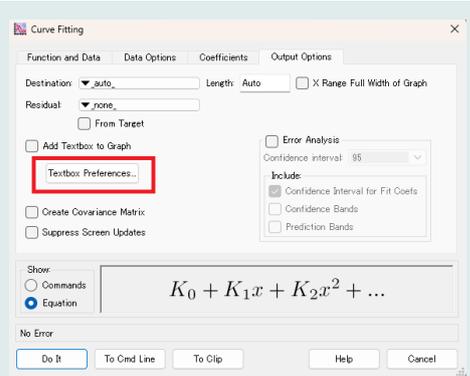
もし、エラーウェーブが1標準偏差以上を表している場合、または信頼区間を表している場合は、重み付けに使うべきではありません。

Quick Fit メニューの Weight from Error bar Wave を選択してチェックを外すと、エラーバーウェーブを積み付けに使うのを防ぐことができます。

デフォルトでは、カーブフィッティングの結果のレポートが履歴領域に表示されます。

Textbox Preferences ボタンをクリックすると、Curve Fit Textbox Preferences ダイアログが表示されます。

設定は、メニュー Analysis → Curve Fitting を選択して、Curve Fitting ダイアログで Textbox Preferences ボタンをクリックします。



Curve Fit Textbox Preferences ダイアログで、Display Curve Fit Info Textbox をチェックし、下のリストから表示したい項目を選択します。

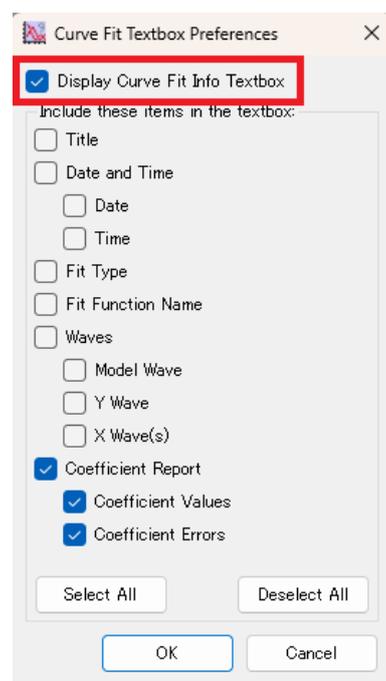
履歴領域に出力される情報のほとんどを含むテキストボックスをグラフに追加するように指定することができます。

Analysis メニューでは、poly2D と Gauss2D のフィッティング関数が使えなくなっていることがあります。

これは、グラフに等高線プロットまたは画像プロットが含まれていないためです。

これらのプロットが含まれている場合は、これらのフィッティング関数を使うことができます。

組み込みのフィッティング関数については、マニュアル III-206 Built-in Curve Fitting Functions を参照してください。



## Quick Fit メニューの制限

Quick Fit メニューでは、使うことができるすべてのカーブフィッティングオプションにアクセスすることはできません。

ユーザー定義のフィッティング関数、自動残差計算、マスキング、信頼区間分析にはアクセスできません。

Quick Fit では、常に自動推定が使われます。

自動推定がうまくいかない場合は、Curve Fitting ダイアログを使って手動で推定を入力する必要があります。

グラフに、画像ピクセルサイズを設定するために補助的な X および Y ウェーブを使う画像が表示されている場合、Quick Fit ではフィッティングすることができません。

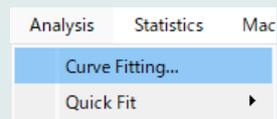
これは、画像プロット用のウェーブが余分なポイントを持つため、フィッティングに適していないためです。

等高線プロットでは、データの中心となる X と Y のウェーブを使っていて、これらはフィッティングに使うことができます。

Quick Fit は、このような等高線に対しては、適切な処理を行います。

## Curve Fitting ダイアログの使い方

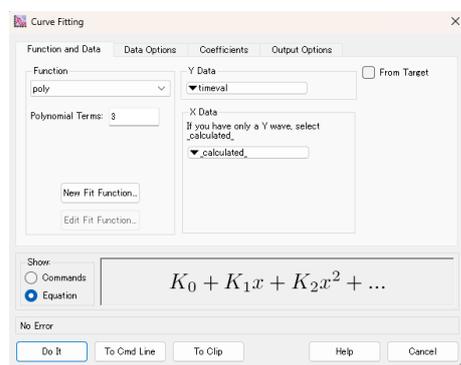
Quick Fit メニューで使うことができないオプションが必要な場合、その次に簡単なフィッティングの方法は、Analysis メニューから Curve Fitting を選択することです。



これは、Curve Fitting ダイアログを表示し、フィッティング関数とデータウェーブの選択、およびカーブフィッティングの各種オプションの設定を行うことができます。

必要に応じて、ダイアログで初期推定を入力することができます。

Curve Fitting ダイアログは、新しいユーザー定義フィッティング関数を作るためにも使えます。



ほとんどのカーブフィッティングは、Curve Fitting ダイアログを使って行うことができます。

同じフィッティング関数を使って、多数のデータセットに多数のフィッティングを行う必要がある場合、おそらく Igor のプログラミング言語でプロシージャを記述して行うことになるでしょう。

Curve Fitting ダイアログを使ってユーザー定義フィッティング関数を作る機能は、ほとんどの一般的なケースに対応できますが、非常に複雑なフィッティング関数を作るには最適な方法ではないかもしれません。

そのような場合、プロシージャウィンドウにフィッティング関数を記述する必要があります。

これは、マニュアル III-250 User-Defined Fitting Functions で説明しています。

非常に複雑なユーザー定義フィッティング関数は、Curve Fitting ダイアログではうまく動作しない場合があります。

場合によっては、プロシージャウィンドウにフィッティング関数を記述し、そのダイアログを使ってフィッティングの設定と実行を行う必要があります。

その他のケースでは、ユーザープロシージャを使うか、コマンドラインで入力して、手動で操作を行う必要があるかもしれません。

このようなケースはごく稀です。

### シンプルなケース：組み込み関数を使ってフィッティング (Line Fit)

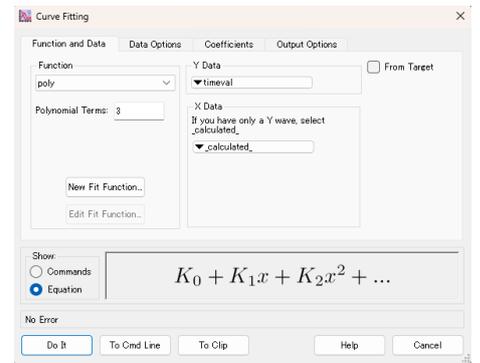
まず初めに、簡単な組み込みフィッティングである、Line Fit (直線フィッティング) について説明します。

理論的に、データは  $y=ax+b$  という関数で表されるはずだと考える理由があるかもしれません。

単に、データが線に沿って並んでいるように見える、という経験則的な観察があり、その線の特徴づけただけなのかもしれません。

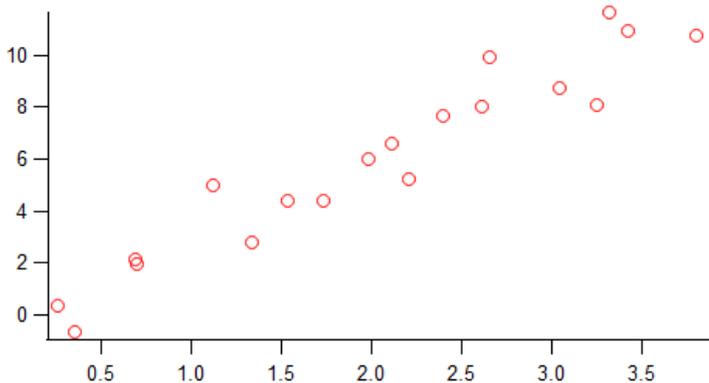
理論的な根拠があればよいのですが、そううまくはいきません。

Curve Fitting ダイアログは、4つのタブで構成されています。各タブには、フィッティング処理のいくつかの側面のコントロールが含まれています。デフォルトのオプションを使って組み込み関数にフィッティングするだけの場合は、Function タブと Data タブのみが必要となります。



グラフに表示されたデータに直線を当てはめるために必要な手順を説明します。その他の組み込み関数も、ほぼ同様の方法で動作します。

次のようなグラフでデータが表示されているとします。今、このデータに最もフィットした線を見つけたいと考えています。

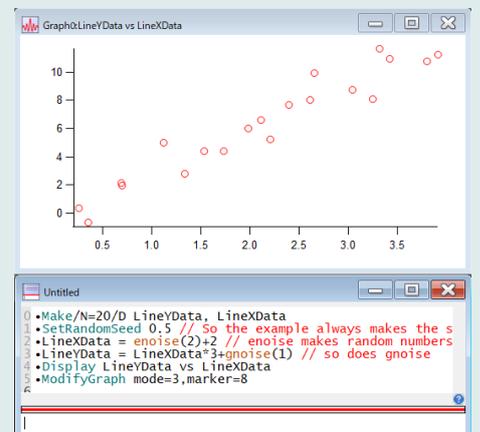


## 1. 新しい Experiment を作成して、コマンドウィンドウで次を実行して、ウェーブとグラフを作成します。

noise 関数で生成された「ランダム」は散布データが上記と同じになるように、SetRandomSeed コマンドを使います（これを読みながら自分で操作する場合は、ここに表示されている同じデータとグラフを作成できます）。

```
Make/N=20/D LineYData, LineXData
SetRandomSeed 0.5
LineXData = noise(2)+2
LineYData = LineXData*3+gnoise(1)
Display LineYData vs LineXData
ModifyGraph mode=3,marker=8
```

最初の行では、2つのウェーブがデータを読み込みます。2行目では、疑似乱数ジェネレーターのシードを設定し、再現可能なノイズを生成します。3行目は、ゼロから4までの範囲で X ウェーブに一様に分布する乱数を埋めています。4行目では、原点を通り、傾きが3の直線に乗ったデータで Y ウェーブを埋め、正規分布するノイズを追加します。

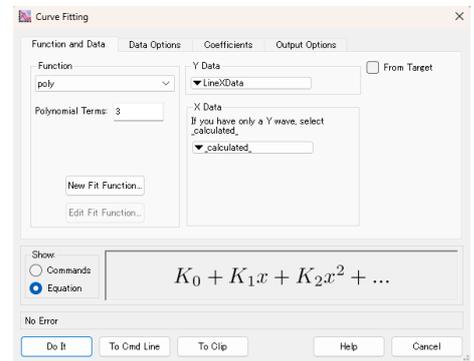


最後の2行でグラフを作成し、マーカーモードで表示し、マーカーとして白抜きのコピーを設定しています。

## 関数とデータの選択

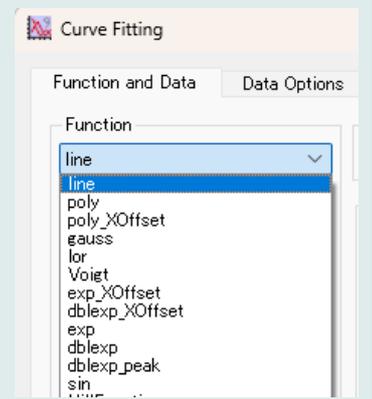
### 2. Analysis メニューから Curve Fitting を選択すると、Curve Fitting ダイアログが表示されます。

まだこのダイアログを使用したことがなければ、Function and Data タブが表示された状態で画面が表示されます。



### 3. カーブフィッティングを行う最初のステップは、フィッティング関数を選択することです。

今回は単純な直線フィッティングを行うので、Function メニューから「line」を選択します。



### 4. Y Data メニューから Y データを選択します。

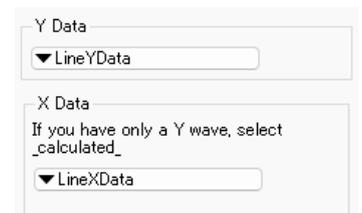
ウェブフォームデータがある場合は、X Data メニューで「\_calculated\_」が選択されていることを確認してください。

X と Y のデータウェーブが別々になっている場合は、X Data メニューで X ウェーブを選択する必要があります。

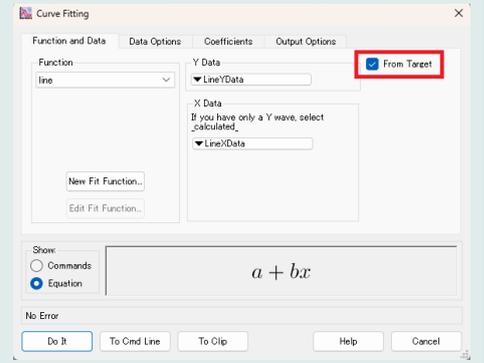
このメニューでは、Y ウェーブと同じデータポイント数を持つウェーブのみが表示されます。

メニューに X ウェーブが表示されない場合、通常はポイント数の不一致が原因です。

今回の直線フィッティングの例では、Y Data メニューから LineYData を、X Data メニューから LineXData を選択します。



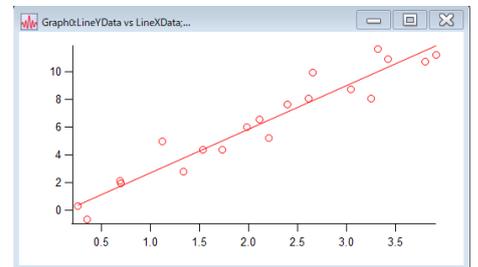
5. Experiment に多数のウェーブがある場合は、From Target チェックボックスを選択すると簡単になる場合があります。最前面のグラフまたはテーブルからウェーブのみが選択されると、Y ウェーブと X ウェーブのメニューに表示され、グラフ上のトレースで使われているウェーブのペアを選択しようと試みます。



6. これで、フィッティングを行うための準備が整いました。今回の単純なケースでは、ダイアログ内の他のタブを設定する必要はありません。

**Do It** をクリックすると、フィッティングが実行されます。

回帰フィットのグラフはこのような形になります。



グラフに表示されているモデルラインに加えて、履歴領域にはさまざまな情報が表示されます。

```

•CurveFit line LineYData /X=LineXData /D ← ダイアログが生成したコマンドライン
  fit_LineYData= W_coef[0]+W_coef[1]*x ← コピーされ、モデルカーブを再評価するのに使用される
  W_coef={-0.39815,3.136} ← (ウェーブとして) フィッティング係数
  V_chisq= 20.3201;V_npnts= 20;V_numNaNs= 0;V_numINFs= 0;
  V_startRow= 0;V_endRow= 19;V_q= 1;V_Rab= -0.885758;
  V_Pr= 0.960731;V_r2= 0.923004;
  W_sigma={0.512,0.213} ← (ウェーブとして) フィッティング係数の標準偏差
  Coefficient values ± one standard deviation
    a      = -0.39815 ± 0.512 ← ダイアログに表示される名前を使って、
    b      =  3.136 ± 0.213 ← 係数値をリスト
  
```

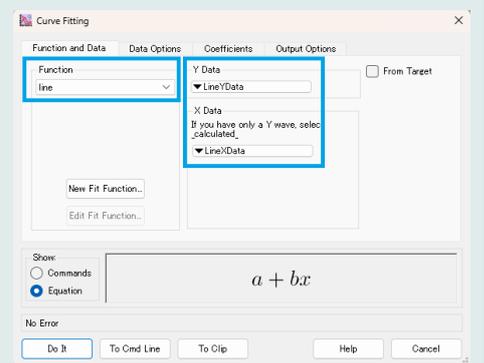
## 2つの便利な追加機能：係数の保持と残差の生成

フィッティングを行い、グラフを確認した結果、原点を通るべきだと考える理由があることに気付いたとします。測定された Y 値にばらつきがあるため、フィッティングされた直線は原点を外れています。解決策は、Y 切片係数をゼロに固定した状態で、フィッティングを再度行うことです。

また、フィッティングの視覚的な確認として残差を表示したい場合もあるでしょう。

## 7. Curve Fitting ダイアログを再度表示します。

ダイアログは前回使用した設定を記憶しているため、Function メニューでは line フィット関数が既に選択され、Y Data メニューと X Data メニューではデータウェーブが選択されています。



## 8. Coefficients タブを選択します。

各係数は係数リストの行に対応しています。

「Hold?」と表示された列のチェックボックスをクリックすると、その係数の値が固定されます。

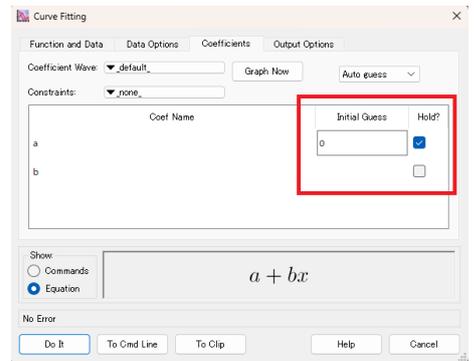
係数の値を指定するには、Initial Guess 列にあるボックスに数値を入力します。

Hold ボックスにチェックを入れるまでは、Initial Guess ボックスは表示されません。

組み込みのフィッティングには初期推定値が不要なためです。

値を入力するにはボックスをクリックします。

入力が完了したら、Enter キーを押してボックスの編集モードを終了します。



## 9. 次に、フィッティングの残差を計算し、グラフに追加します。

Output Options タブをクリックし、Residual メニューから \_auto trace\_ を選択します。

残差にはいくつかのオプションがあります。

ここでは、残差を計算してグラフに追加するため、\_auto trace\_ を選択しました。

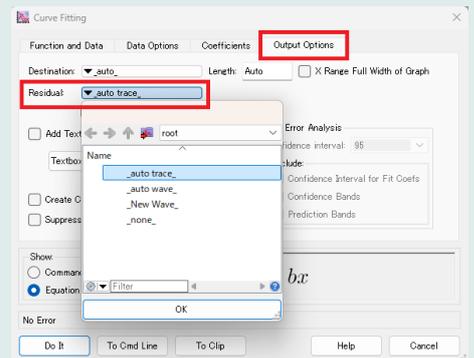
常にグラフに残差を追加したいわけではないかもしれません。

\_auto wave\_ を選択すると、残差は自動的に計算されますが、グラフには表示されません。

\_auto trace\_ と \_auto wave\_ のどちらも、Y ウェーブと同じ名前のウェーブを作成しますが、名前には「Res\_」がプリフィックスとして付けられます。

\_New Wave\_ を選択すると、新しいウェーブを作成するコマンドが生成され、任意の名前を付けて残差で満たすことができます。

グラフには追加されません。

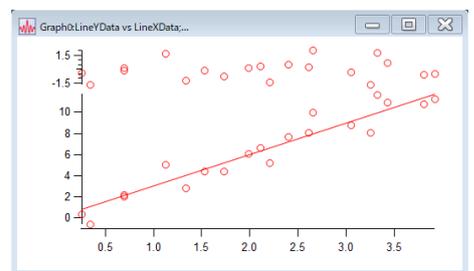


## 10. Do It をクリックすると、直線が原点を通るようにフィッティングが再計算され、ゼロに固定されます。

残差も計算され、グラフに追加されます。

グラフの線がゼロで垂直軸と交差していないことに注意してください。

これは、横軸がゼロまで伸びていないためです。



a をゼロに固定すると、履歴領域に表示されるフィッティングの結果は次のようになります。

CurveFit コマンドの /H フラグは 1 つ以上の係数が固定されていることを示します。

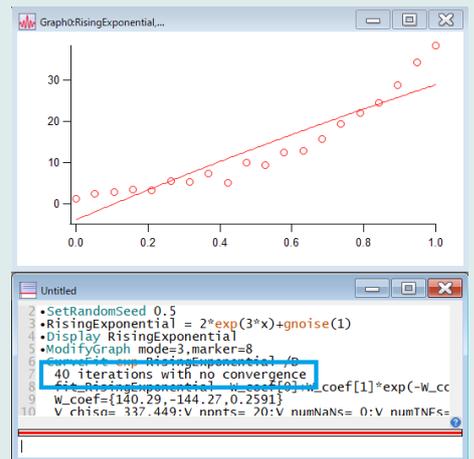
- K0 = 0;
- CurveFit/H="10" line LineYData /X=LineXData /D /R  
fit\_LineYData= W\_coef[0]+W\_coef[1]\*x  
Res\_LineYData= LineYData[p] - (W\_coef[0]+W\_coef[1]\*LineXData[p])  
W\_coef={0,2.989}



Fit Progress ダイアログが表示されるので、OK をクリックします。

3. 自動推定は、指数データが指数の負の係数によってうまく説明されると仮定しているため、フィッティングはうまくいきません。

```
•CurveFit exp RisingExponential /D
40 iterations with no convergence
fit_RisingExponential= W_coef[0]+W_coef[1]*exp(-
W_coef[2]*x)
W_coef={140.29,-144.27,0.2591}
V_chisq= 337.449;V_npnts= 20;V_numNaNs=
0;V_numINFs= 0;
V_startRow= 0;V_endRow= 19;
W_sigma={362,360,0.739}
Coefficient values ± one standard deviation
y0      = 140.29 ± 362
A       = -144.27 ± 360
invTau  = 0.2591 ± 0.739
```



グラフ化されたフィッティング曲線がデータポイントに沿っていないという事実に加えて、フィッティング係数の推定不確実性が異常に大きいです。

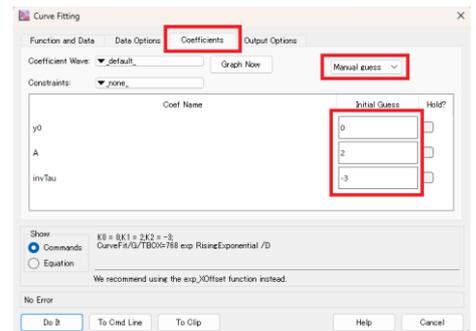
4. 解決策は、独自の初期推定値を入力することです。

Curve Fitting ダイアログの Coefficients タブをクリックし、右上のメニューで Manual Guesses を選択します。

係数リストの Initial Guess 欄に、invTau の負の値を含む初期推定を入力できるようになりました。

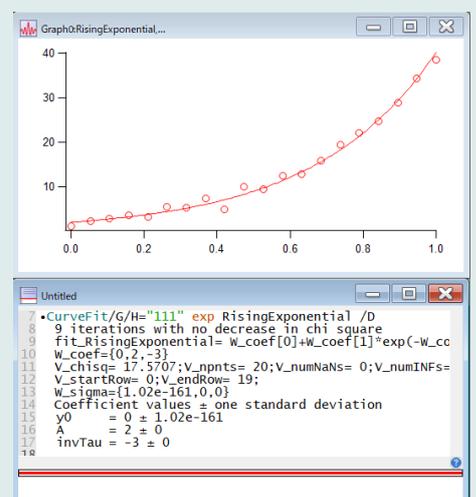
このケースでは、次のような初期推定値を入力します：

```
y0      0
A       2
invTau  -3
```



5. Do It をクリックすると、初期推定値を設定するための追加のコマンドを生成し、今後はフィッティングが正しく機能します。

```
•K0 = 0;K1 = 2;K2 = -3;
•CurveFit/G/TBOX=768 exp RisingExponential /D
Fit converged properly
fit_RisingExponential= W_coef[0]+W_coef[1]*exp(-
W_coef[2]*x)
W_coef={-0.95257,2.6625,-2.707}
V_chisq= 14.1036;V_npnts= 20;V_numNaNs= 0;V_numINFs=
0;
V_startRow= 0;V_endRow= 19;
W_sigma={0.832,0.461,0.164}
Coefficient values ± one standard deviation
y0      = -0.95257 ± 0.832
```



A = 2.6625 ± 0.461  
invTau = -2.707 ± 0.164

ゼロから初期推定値のセットを見つけるのは難しいかもしれません。

自動推定は、修正することで適切な初期推定値を提供する良い出発点となる可能性があります。

そのため、ダイアログには Only Guess モードが用意されています。

Only Guess が選択されている場合、自動初期推定値を作成するために Do It をクリックするとフィッティングを行わずに終了します。

もう一度、Curve Fitting ダイアログを開くと、自動推定によって作成された係数ウェーブ (Coefficient Wave メニューで `_default_` を選択した場合は `W_coef`) を選択できるようになります。

このウェーブを選択すると、初期推定値が自動推定値として設定されます。

次に、Manual Guess を選択し、初期推定を修正します。

Graph Now ボタンは、良い初期推定値を見つけるのに役立つかもしれません (マニュアル III-192 Coefficients Tab for a User-Defined Function を参照)。

## 定数でフィッティング

組み込みフィッティング関数のうちのいくつかは、フィッティング中に変化しない定数を含んでいます。

これらは、例えば、数値の安定性を向上させるための一定の X オフセットを提供するためにのみ入力します。

このような組み込みフィッティング関数の 1 つに `exp_XOffset` フィッティング関数があります。

これは次の式です。

$$y_0 + A \exp\left(\frac{x - x_0}{\tau}\right)$$

ここで、 $y_0$ 、 $A$ 、 $\tau$  はフィッティング係数であり、反復的なフィッティングの過程で変化し、最終的な値がフィッティングの解となります。

一方、 $x_0$  は定数であり、変化することはありません。

むしろ、フィッティングの設定の一部として任意の値を設定します。

`exp_XOffset` フィッティング関数の場合は、ユーザーが設定しないと、デフォルトで入力データの最小の X 値に設定されます。

原点から離れたデータにフィッティングする場合、数値安定性が向上します。

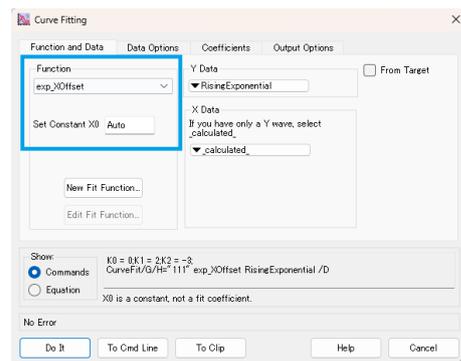
当然、最終的な解の  $A$  の値に影響します。

Curve Fitting ダイアログで、定数を使う組み込みフィッティング関数を選択すると、フィッティング関数メニューの下に追加の編集ボックスが表示され、定数の値を設定できるようになります。デフォルト値の `Auto` に設定すると、入力データに基づいて定数を妥当な値に設定します。

定数を使う組み込みフィッティング関数を使用してフィッティングを行うと、出力に `W_fitConstants` というウェーブが含まれます。

このウェーブの各要素は、方程式の定数の値を保持しています。

複数の定数を使う組み込みフィッティング関数ではない場合、ウェーブには要素が 1 つだけ含まれます。



特定のフィッティング関数で使われる定数については、マニュアル III-206 Built-in Curve Fitting Functions を参照してください。

## ユーザー定義関数でフィッティング

ユーザー定義関数へのフィッティングは、組み込み関数へのフィッティングとほぼ同じですが、主に 2 つの違いがあります：

- フィッティング関数を定義する必要がある
- 初期推定値を指定する必要がある

ユーザー定義フィッティング関数の作成を説明するために、対数関数をフィッティングする関数を作成します： $y = C_1 + C_2 \ln(x)$ 。

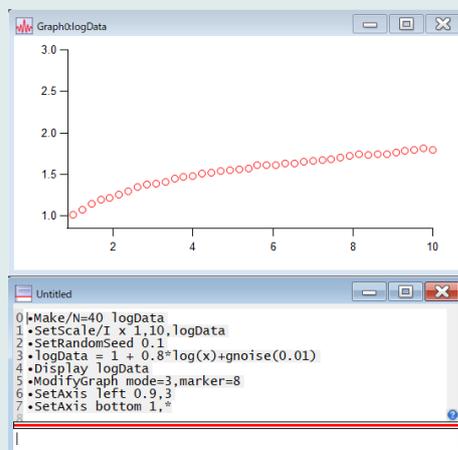
## 関数の作成

### 1. 新しい Experiment を作成して、コマンドウィンドウで次を実行して、ウェーブとグラフを作成します。

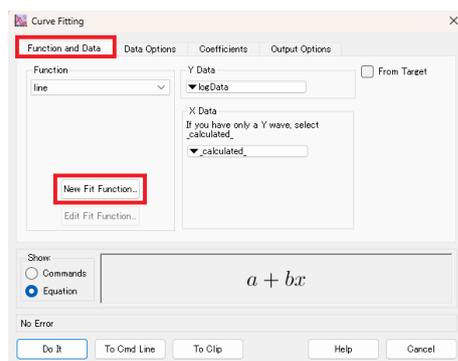
(ここでは、対数データの生成→グラフの作成→マーカーと軸の設定の順で行いました)

```
Make/N=40 logData
SetScale/I x 1,10,logData
SetRandomSeed 0.1
logData = 1 + 0.8*log(x)+gnoise(0.01)
Display logData
ModifyGraph mode=3,marker=8
SetAxis left 0.9,3
SetAxis bottom 1,*
```

これは、40 ポイントのウェーブを作成し、その X スケーリングを 1 から 10 の範囲をカバーするように設定し、対数値とノイズを少し加えてウェーブを満たし、グラフを作成します。

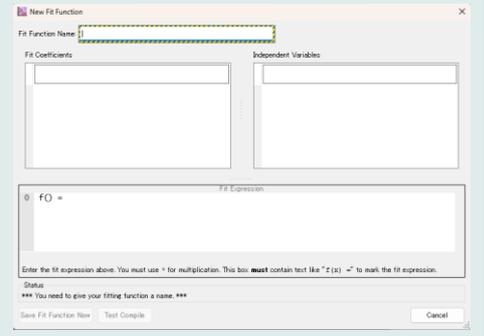


### 2. ユーザー定義フィッティング関数を作成するには、Curve Fitting ダイアログ (メニュー Analysis → Curve Fitting) の Function and Data タブにある New Fit Function ボタンをクリックします。



### 3. New Fit Function ダイアログが表示されます。

関数名を入力し、Fit Coefficients リストに係数名を入力し、Independent Variables リストに独立変数名を入力し、Fit Expression ウィンドウにフィッティング式を入力する必要があります。

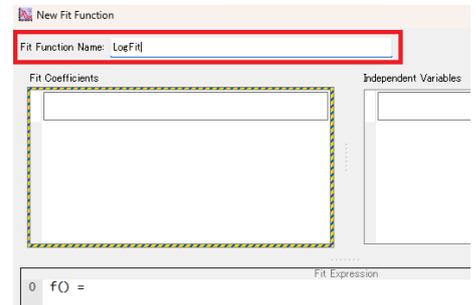


### 4. 関数名は、非自由形式の命名規則に従う必要があります。

つまり、文字で始まり、文字、数字、アンダースコア文字のみを含み、255 バイト以下でなければなりません（マニュアル III-501 Object Names を参照）。

組み込み関数、ユーザープロシージャ、ウェーブ名など、他のオブジェクトの名前と同じであってははいけません。

**この例でのログ関数には、「LogFit」と名付けます。**



### 5. Tab キーを押すと、フィット係数のリストの最初の項目に移動します。

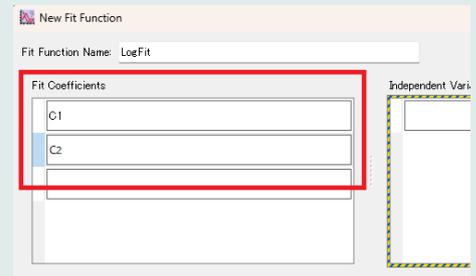
リストには常に 1 つの空欄があり、ここに新しい係数名を追加することができます。

まだ何も入力していないため、空欄は 1 つだけです。

名前を入力するたびに、Enter キーを押します。

その名前が確定すると、次の名前を入力できる新しい空欄を作成します。

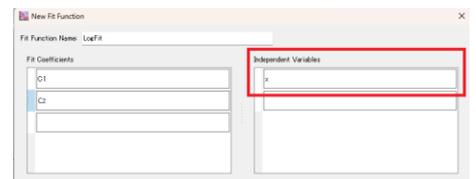
**ここでは、C1 と C2 を名前として入力します。**



### 6. Independent Variables リストの最初の空欄をクリックします。

ほとんどのフィッティング関数は、独立変数を 1 つだけ必要とします。

**ここでは、独立変数に x と名前を付けることにします。**



### 7. フィッティングの式を入力します。

Independent Variables リストに名前を入力すると、下の Fit Expression のテキストに入力されることに気づくと思います。

フィッティング関数の戻り値（ほとんどの場合 Y 値）は、

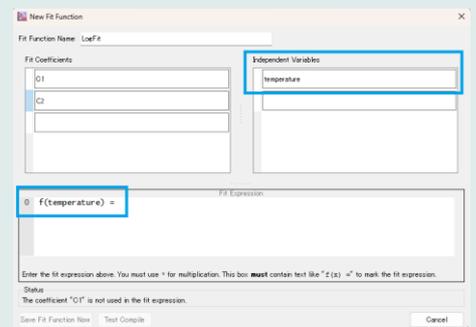
「f(x)=」 のように表示されます。

独立変数として「temperature」と入力した場合は、

「f(temperature)=」と表示されます。

この「f(x)=」というテキストは必須です。

そうしないと、関数の戻り値が不明となります。



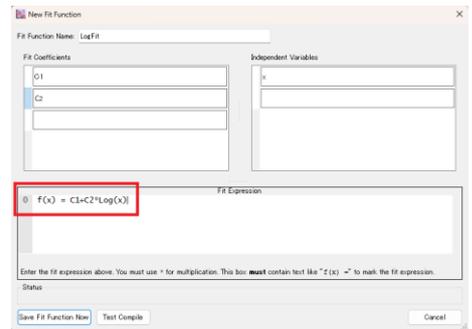
## 8. Fit Expression は代数式ではありません。

コマンドラインのコマンドと同じ形式で入力する必要があります。  
正しい記述の作成にヘルプが必要な場合は、マニュアル IV-4  
Assignment Statements を参照してください。

**入力する式は、代入文の右辺です。今回の例の対数の式は次のようになります：**

$$f(x) = C1+C2*Log(x)$$

乗算には明示的な「\*」が必要です。



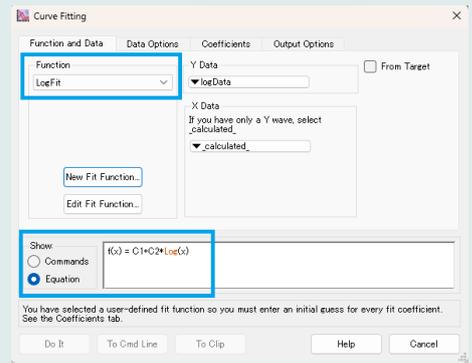
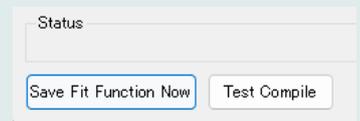
**9. ダイアログは、Fit Expression のエラーをチェックします。**  
例えば、係数または独立変数のどちらかが式から欠落している場合、Save Fit Function Now ボタンは使えません。

ダイアログでは、正しい式の構文をチェックすることはできません。

簡易的にチェックできる項目がすべて正しい場合、Save Fit Function Now と Test Compile ボタンが有効になります。

どちらかをクリックすると、プロシージャウィンドウに新しい関数が入力され、プロシージャのコンパイルが試行されます。

**Save Fit Function Now ボタンをクリックしてコンパイルが成功すると、Function メニューで新しい関数が選択された状態で、Curve Fitting ダイアログに戻ります。**



**10. コンパイルエラーが発生した場合は、コンパイラのエラーメッセージがステータスボックスに表示され、問題のある部分がハイライト表示されます。**

よくある間違いは、式のどこかで係数名をスペルミスしてしまうことです。

例えば、C1 の代わりに CC1 とタイプしてしまった場合、右のような結果になる可能性があります。

C1 が式に表示されていることに注意してください。

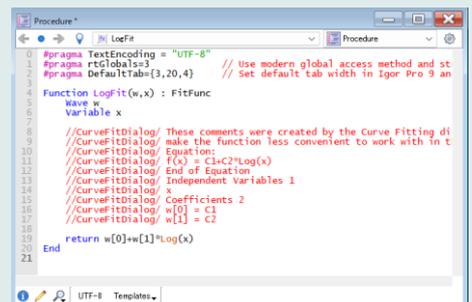
そうでなければ、ダイアログに C1 が欠落していることが表示されます。



**11. 準備がすべて整ったら、Save Fit Function Now ボタンをクリックして、Procedure ウィンドウに関数を構築します。Curve Fitting ダイアログは、Cancel していったん閉じます。**

**メニュー Windows → Procedure Windows → Procedure Window を選択します。**

これには、ダイアログのさまざまな情報を識別する機能コード内のコメントも含まれています。



今回の例では、関数は次のようになります：

```
Function LogFit(w,x) : FitFunc
    Wave w
    Variable x

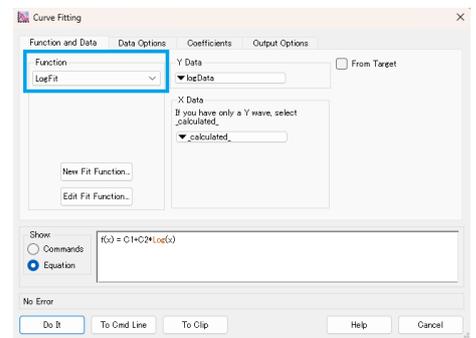
    //CurveFitDialog/ These comments were created by the Curve Fitting dialog. Altering
them will
    //CurveFitDialog/ make the function less convenient to work with in the Curve Fitting
dialog.
    //CurveFitDialog/ Equation:
    //CurveFitDialog/ f(x) = C1+C2*Log(x)
    //CurveFitDialog/ End of Equation
    //CurveFitDialog/ Independent Variables 1
    //CurveFitDialog/ x
    //CurveFitDialog/ Coefficients 2
    //CurveFitDialog/ w[0] = C1
    //CurveFitDialog/ w[1] = C2

    return w[0]+w[1]*Log(x)
End
```

関数が非常に複雑でダイアログで処理できない場合を除き、Procedure ウィンドウのコードを修正する必要はありません。

Procedure ウィンドウでのフィッティング関数の記述方法の詳細については、マニュアル III-250 User-Defined Fitting Functions を参照してください。

**12.** もう一度 Curve Fitting ダイアログを開いて、Function メニューを確認すると、LogFit がフィッティング関数として選択できるようになっています。



## ユーザー定義関数の Coefficients タブ

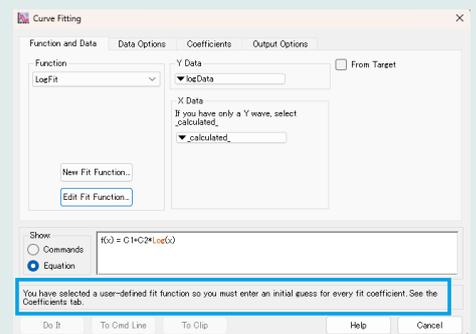
ユーザー定義関数にフィッティングするには、Coefficients タブで初期推定値を入力する必要があります。

ユーザー定義のフィッティング関数を作成した（または既存の関数を選択した）場合、ダイアログの下部のエラーメッセージウィンドウに次のように表示されます：

You have selected a user-defined fit function so you must enter an initial guess for every fit coefficient. See the Coefficients tab.

（ユーザー定義のフィッティング関数を選択したため、すべてのフィッティング係数の初期推定値を入力する必要があります。

Coefficients タブを見てください）

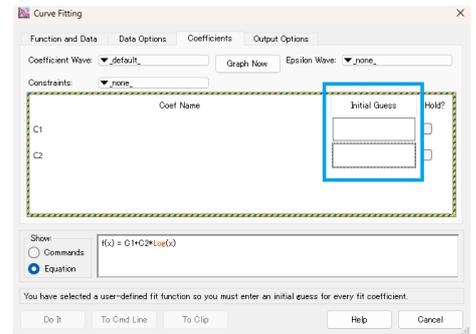


ユーザー定義のフィッティング関数を選択すると、係数リストの Initial Guess 列が有効になります。

各行には数値を入力する必要があります。

一部の関数はフィッティングするのが難しい場合があります。

そのような場合、初期推定値が最終的な解にかなり近い値になることがあります。

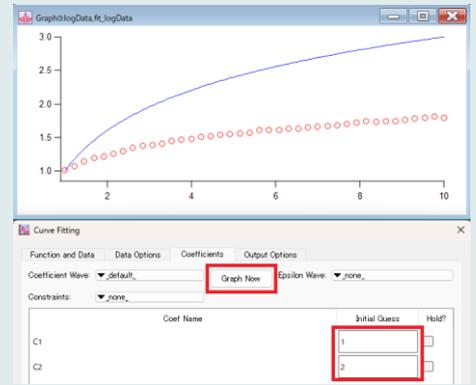


初期推定値を適切に設定できるように、Coefficients タブには Graph Now ボタンが用意されています。

このボタンをクリックすると、入力した初期推定値を使って、フィッティング関数を表すトレースが前面のグラフに追加されます。

初期推測の列の値を変更し、Graph Now ボタンを何度でもクリックすることができます。

変更が加えられるたびにトレースが更新されます。

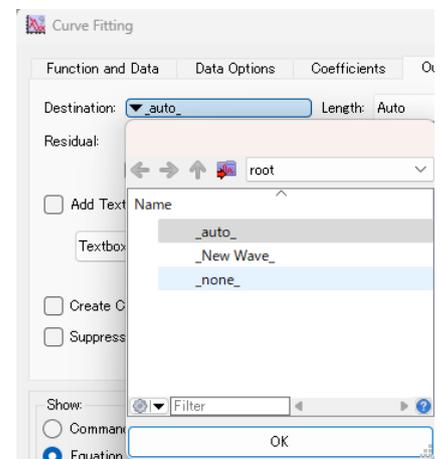


Graph Now ボタンは、マニュアル III-196 The Destination Wave で説明されているように動作しますが、1つの例外があります。

Output Options タブの Destination ポップアップメニューで \_none\_ を選択した場合、Graph Now ボタンは \_auto\_ を選択した場合と同じように動作します。

Graph Now ボタンは、選択した Destination のウェーブスタイルを反映し、\_New Wave\_ を選択した場合は新しいウェーブを作成します。

Do It をクリックする前に Destination のウェーブ設定を変更しない限り、Graph Now で設定したウェーブはフィッティングによって上書きされます。



Coefficient タブでは、「イプシロン」ウェーブを選択するオプションがあります。

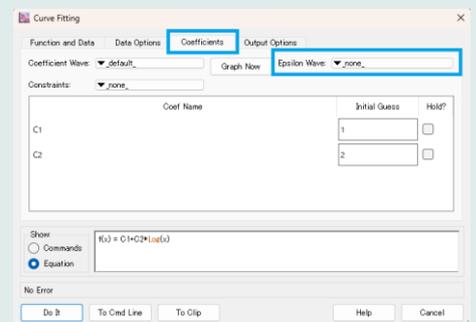
イプシロンウェーブには、係数ウェーブの各ポイントのイプシロン値が1つずつ含まれます。

デフォルトでは、Epsilon Wave メニューは \_none\_ に設定されていて、イプシロン値はデフォルト値に設定されていることを示しています。

各イプシロン値は、フィッティング係数に関する偏微分を計算するために使われます。

偏微分は、カイ二乗値が最小となる係数の探索方向を決めるために使われます。

ほとんどの場合、イプシロン値はクリティカルではありません。ただし、デフォルトのイプシロン値が許容できる偏微分を提供していないと判断する理由がある場合は、独自の値を指定することができます（特異行列エラーを回避できる場合もあります）。



イプシロン値を指定するには、Epsilon Wave メニューから既存のウェーブを選択するか、\_New Wave\_ を選択します。メニューには、フィッティング係数と同じポイント数を持つウェーブのみが表示されます。どちらを選択しても、係数リストにイプシロン列が表示され、イプシロン用の値を入力することができます。

Epsilon Wave メニューからウェーブを選択すると、そのウェーブの値がリストに入力されます。\_New Wave\_ を選択すると、ダイアログがイプシロンウェーブを作成し、Epsilon 列の値でそれを埋めるためのコマンドを生成します。

イプシロンウェーブの詳細とその機能については、マニュアル III-267 The Epsilon Wave を参照してください。

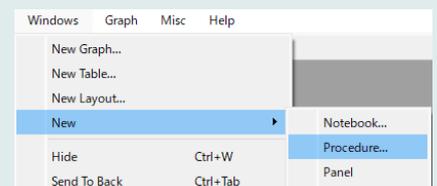
## ユーザー定義関数が常に使えるようにする

フィッティング関数は Procedure ウィンドウで作成されるため、Experiment ファイルの一部として保存されることに注意してください。

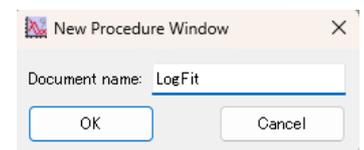
つまり、作成した Experiment で作業している間は使うことができますが、他の Experiment ファイルで作業するときは使うことができません。

Igor Pro を起動するたびに作成したフィッティング関数を使うことができるようになります。前のセクションからの続きで説明します。

### 1. Windows メニューの New から Procedure を選択して、New Procedure Window を開きます。

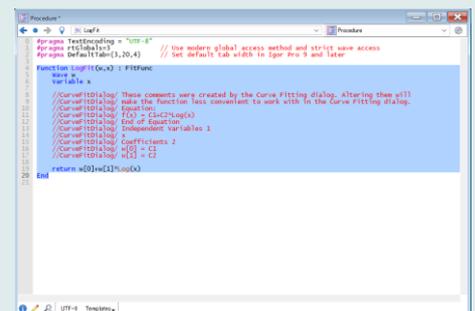


### 2. 前のセクションでは、フィッティング関数に LogFit と名前を付けたので、Document name には LogFit と入力して、OK をクリックします。

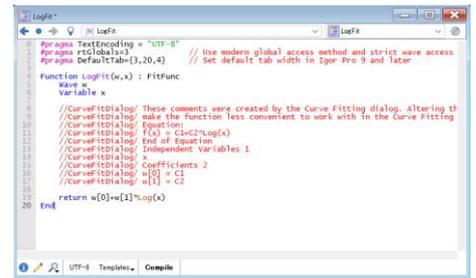


### 3. メニュー Windows → Procedure Windows → Procedure Window を選択します。

Procedure ウィンドウでフィッティング関数を見つけ、Function から End までのテキストをすべて選択し、Edit メニューから Cut を選択します。



#### 4. コードを新しいプロシージャウィンドウに貼り付けます。



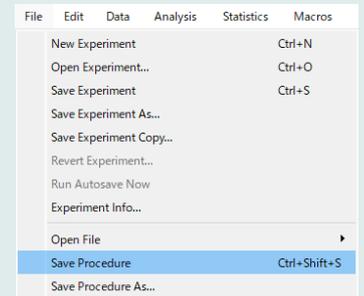
```
LogFit
-----
#pragma TextEncoding = "UTF-8" // Use modern global access method and strict wave access
#pragma rscLibTab(3,20,4) // Set default tab width in Igor Pro 9 and later

Function LogFit(w,x) : FitFunc
Wave w
Variable x

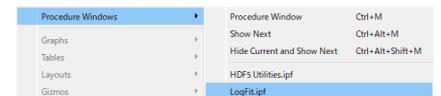
//CurveFitDialog/ These comments were created by the Curve Fitting dialog. Altering th
//CurveFitDialog/ make the function less convenient to work with in the Curve Fitting
//CurveFitDialog/ Equation
//CurveFitDialog/ F(x) = C1+C2*Log(x)
//CurveFitDialog/ End of Equation
//CurveFitDialog/ Independent Variables 1
//CurveFitDialog/ Coefficients 2
//CurveFitDialog/ w[0] = C1
//CurveFitDialog/ w[1] = C2

return w[0]+w[1]*Log(x)
End
```

5. 最後に、File メニューから **Save Procedure Window** を選択し、Igor Pro User Files フォルダの Igor Procedure に保存します（詳細はマニュアル II-31 Igor Pro User Files を参照）。



6. これで次に Igor Pro を起動すると、Windows メニューの Procedure Windows でそのプロシージャを選択することができます。



### ユーザー定義関数を削除する

不要になったユーザー定義フィッティング関数を削除するには、Windows メニューから Procedure Windows を選択します。

Procedure ウィンドウで関数を探します（Edit メニューから Find を選択し、関数名を検索します）。

Function という語から End という語までの関数を定義するテキストをすべて選択し、テキストを削除します。

前のセクションで説明した、常に使えるように設定した場合は、Igor Pro User Files フォルダの Igor Procedures フォルダ内の対応するプロシージャファイルを見つけ、フォルダから削除して、Igor を再起動してください。

### ユーザー定義フィッティング関数の詳細

ユーザー定義フィッティング関数を入力するもっとも簡単な方法は、New Fit Function ダイアログを使うことです。

しかし、フィッティングする式が非常に長い場合や、ローカル変数や条件分岐を含む複数の行が必要な場合、ダイアログが煩雑になる可能性があります。

また、特定の特殊な状況では、ダイアログでサポートされていない書式が必要になる場合があります。

ユーザー定義フィッティング関数の形式と、異なる形式の使い方についての詳細は、マニュアル III-250 User-Defined Fitting Functions を参照してください。