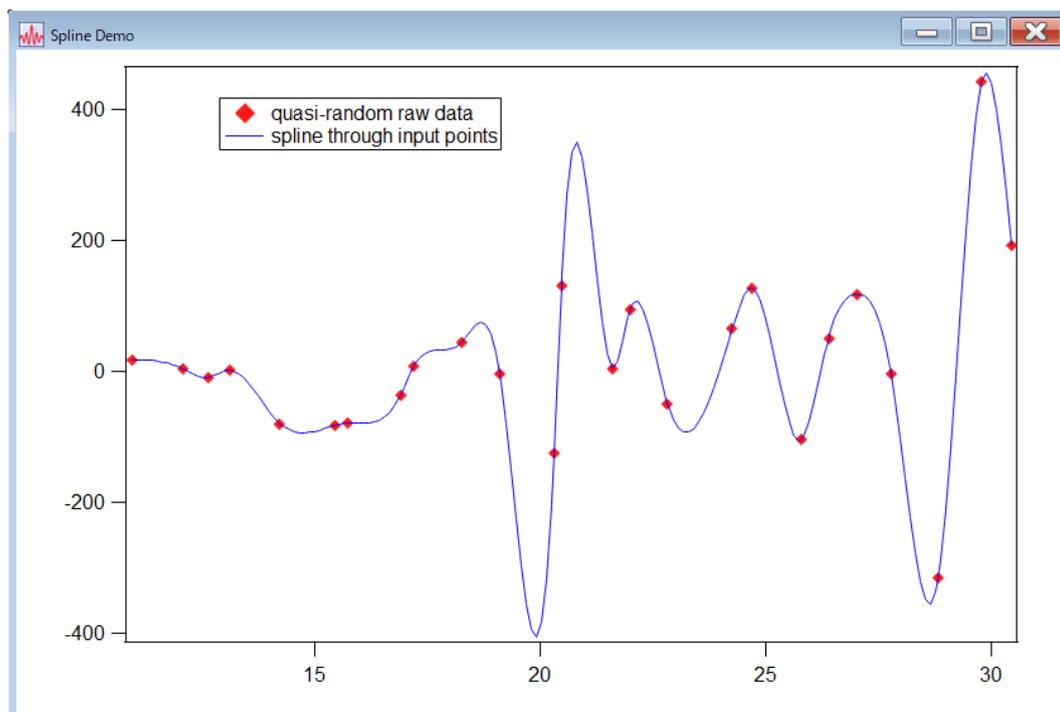


# CONTENTS

---

サンプルの Experiment – Spline Demo (+補間) .....	2
クイックノート .....	2
Macros メニュー .....	2
Interpolate2 コマンドのヘルプ .....	3
スプライン補間の例 .....	3
Interpolate ダイアログ .....	6
指数データの補間 .....	9
Smoothing Spline (平滑化スプライン) のアルゴリズム .....	10
Smoothing Spline (平滑化スプライン) のパラメーター .....	10
Interpolate2 の Pre-averaging (事前平均化) 機能 .....	11
Destination Wave から宛先 X 座標を取得 .....	11
微分と積分 .....	12
area 関数と mean 関数 .....	13
X 範囲と mean、faverage、area 関数 .....	14
ウェーブのセグメントの平均値を求める .....	14
XY データの面積 .....	15

# サンプルの Experiment – Spline Demo (+補間)



## クイックノート

### メニュー File → Example Experiments → Feature Demos → Spline Demo

この Experiment は、少数のデータポイントに滑らかな曲線を通すために3次スプラインを使う方法を示しています。

これは、Interpolate2 コマンド (Analysis メニュー) を使って3次スプラインを実行します。

サンプルの Experiment では、3次スプラインのみを示しています。

Interpolate2 コマンドでは、線形補間と平滑化スプラインも実行できます。

多数のノイズの多いデータポイントに曲線を通すには、通常、平滑化スプライン曲線が適しています。

## Macros メニュー

この Experiment の Macros メニューには次の項目が表示されます。

### Show/Hide Evenly Spaced Spline

等間隔スプラインを表示/非表示します。

### Show/Hide Spline Through Input Points

入力ポイントを通してスプラインを表示/非表示します。

Macros	Windows	Graph	Misc	Help
Display Documentation				Ctrl+0
Show Evenly Spaced Spline				Ctrl+1
Hide Spline Through Input Points				Ctrl+2
Demo Splines				Ctrl+3

## Demo Splines

疑似乱数の発生回数を入力します。

入力ウェーブを変更して描画を試すことができます。

## Interpolate2 コマンドのヘルプ

クイックノートの説明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 Interpolate2 コマンドのヘルプ (Analysis.ihf 内の The Interpolate2 Operation) の内容を記載します。

クイックノート内のリンクをクリックするか、新しい Experiment を作成したときに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 "The Interpolate2 Operation"
```

Interpolate2 コマンドは、1D ウェーブフォームと XY データに対して、線形、3次スプライン、平滑化3次スプライン補間を実行します。

3次スプライン補間は、「Numerical Recipes in C」のルーチンに基づいています。

平滑化スプラインは「Smoothing by Spline Functions」(Christian H. Reinsch, Numerische Mathematic 10, 177-183 (1967)) に基づいています。

Igor Pro 7 以前は、Interpolate2 は Interpolate XOP の一部として実装されていました。

現在は、組み込み関数となっています。

Interpolate XOP では、若干異なる構文を使う Interpolate という古いコマンドも実装されていました。

過去の Experiment で Interpolate コマンドを使っている場合は、Interpolate2 を使うように変換することを推奨します。

線形補間は主に、XY ウェーブを等間隔の X 値における Y 値を含む単一のウェーブに変換するために使われます。これにより、等間隔のデータが必要な FFT などのコマンドを使えるようになります。

任意の XY データから滑らかな曲線を描くには、3次スプライン補間が最も便利です。

結果として得られる曲線には、元のデータとは何の関係もない特徴が含まれる可能性があるため、3次スプラインは、審美的な目的よりも分析的な目的で使う場合は注意が必要です。

線形および3次スプライン補間は、出力曲線をすべての入力ポイントを通るように制約されるため、少数の入力ポイントに対して最も効果的に機能します。

平滑化スプラインにはこの制約がないため、大きなノイズを含むデータセットでもうまく機能します。

Interpolate2 コマンドには、「事前平均」と呼ばれる機能があり、多数の入力ポイントがある場合に使うことができます。

平滑化スプラインをサポートする以前に、大きなノイズのあるデータセットに3次スプラインを適用する方法として、Interpolate2 に事前平均化が追加されました。

現在では、事前平均化ではなく、平滑化スプラインを使うことを推奨します。

## スプライン補間の例

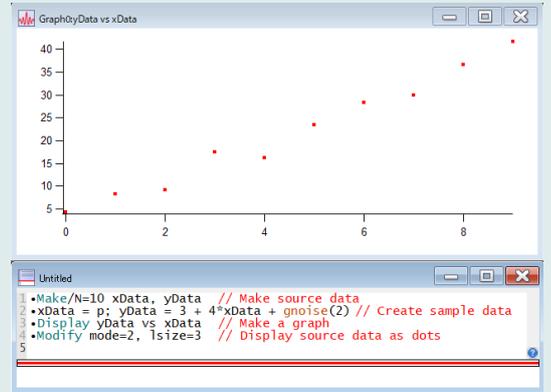
Interpolate2 について詳しく説明する前に、まず、簡単な例を見てください。

(ヘルプ内のコマンドは、選択して Ctrl+Enter キーで実行できます)

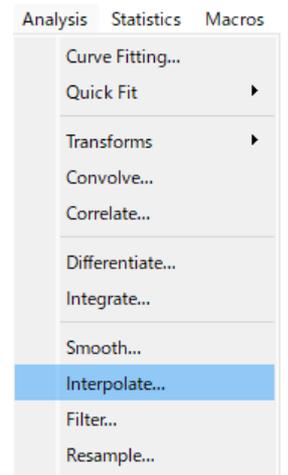
新しい Experiment を作成したところからの手順で確認します。

### 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。

```
// ソースデータの作成
Make/N=10 xData, yData
// サンプルデータの作成
xData = p; yData = 3 + 4*xData + gnoise(2)
// グラフを作成
Display yData vs xData
// データをドットで表示
Modify mode=2, lsize=3
```

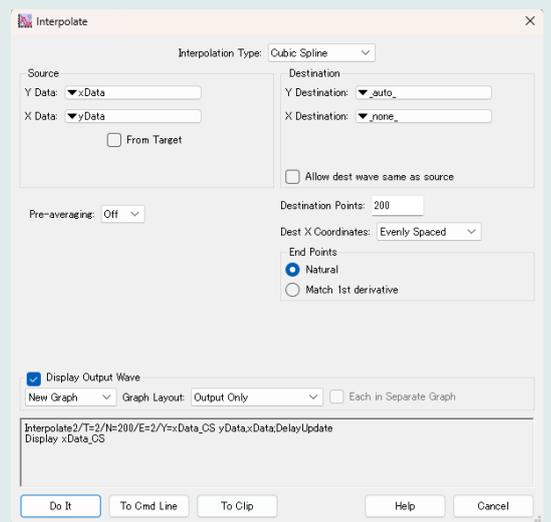


### 2. メニュー Analysis → Interpolate を選択して、Interpolate ダイアログを表示します。



### 3. 次のように設定します。

Interpolation Type: Cubic Spline  
X Data: xData  
Y Data: yData  
X Destination: \_none\_  
Y Destination: \_auto\_  
Destination Points: 200  
Pre-averaging: Off  
Dest X Coords: Evenly Spaced  
End Points: Natural



### 4. ダイアログ下部を見ると、次のようなコマンドが生成されることがわかります。

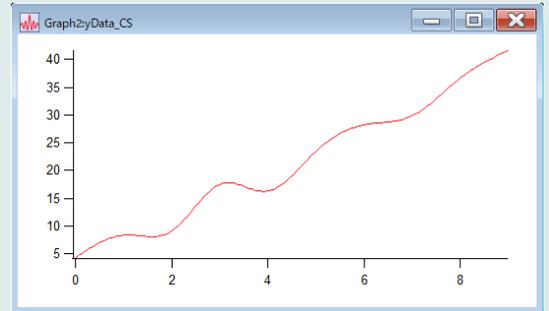
```
Interpolate2/T=2/N=200/E=2/Y=yData_CS yData, xData
```

これは、Interpolate2 が yData を Y ソースウェーブ、xData を X ソースウェーブとして使い、yData\_CS を宛先ウェーブとして作成することを意味します。

「\_CS」は「3次スプライン」を意味します。

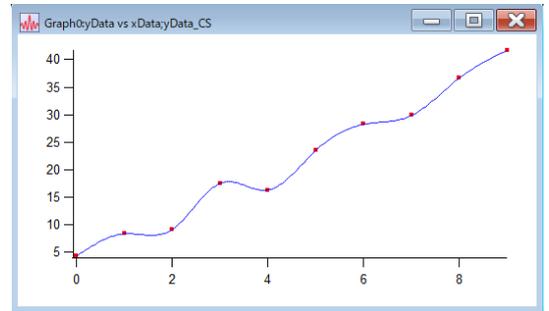
## 5. Do It ボタンをクリックして、補間を実行します。

新しいグラフウィンドウにグラフが表示されます。



## 6. 最初に作成したグラフにこの曲線を表示したいため、最初のグラフを最前面に表示し、コマンドウィンドウで次を実行します。

```
AppendToGraph yData_CS;  
Modify rgb(yData_CS)=(0,0,65535)
```

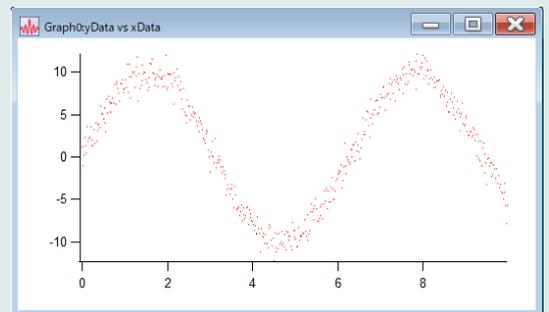


では、より多くの入力データポイントで試してみましょう。  
前の手順の続きから行います。

## 7. コマンドラインで次を実行します。

```
// データポイントの数を増やす  
Redimension/N=500 xData, yData  
// サンプルデータの作成  
xData = p/50; yData = 10*sin(xData) + gnoise(1.0)  
// ドットを小さくする  
Modify lsize(yData)=1
```

yData\_CS のトレースは削除しておきます。



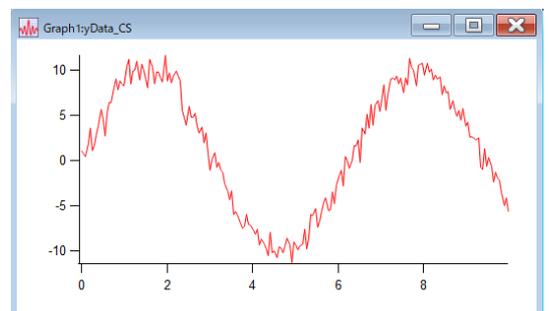
## 8. メニュー Analysis → Interpolate を選択して、Interpolate ダイアログを表示します。

すべての設定は、前のものを変更しないでください。

Do It をクリックします。

結果の3次スプラインは、入力データポイントのすべてを通過しようとしていることに注目してください。

これは通常、私たちが望むことではありません。



## 9. もう一度、メニュー Analysis → Interpolate を選択して、Interpolate ダイアログで次を変更します：

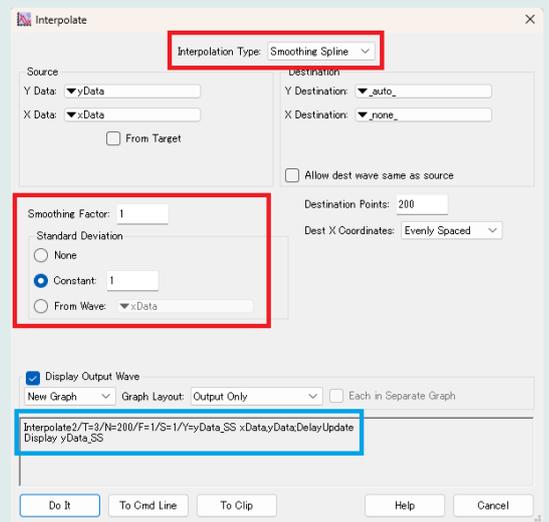
Interpolation Type: Smoothing Spline

生成されるコマンドが yData\_SS という名前のウェーブを参照していることに注意してください。  
これが出力ウェーブとなります。

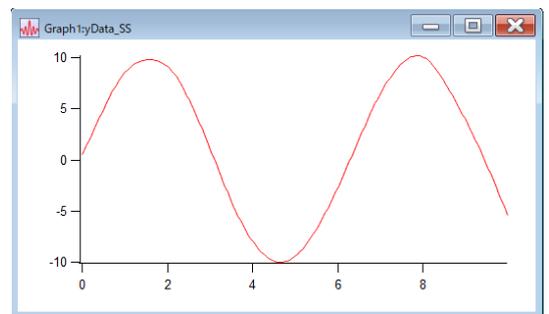
**Smoothing Factor (平滑化係数) には、1.0 を入れます。**

通常、これが良い開始点です。

Standard Deviation セクションで、Constant ラジオボタンをクリックし、Standard Deviation の値として 1.0 を入力します。  
1.0 が正しいのは、上記の「gnoise(1.0)」という関数の結果として、標準偏差 1.0 のノイズがデータに含まれていることが分かっているからです。



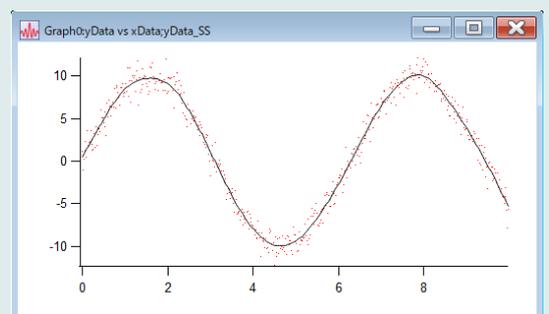
## 10. Do It をクリックして、補間を実行します。



## 11. 前に行ったのと同様に、元のドットのグラフウィンドウを前面に移動し、コマンドウィンドウで次を実行します：

```
AppendToGraph yData_SS;  
Modify rgb(yData_SS)=(0,0,0)
```

平滑化スプラインが大きなノイズのあるデータセットにちょうど良い曲線を加えていることに注目してください。



平滑化スプラインは、平滑化係数パラメータまたは標準偏差パラメータを使って調整することができます。  
ノイズの標準偏差が不明な場合は、平滑化係数を 1.0 のままにして、異なる標準偏差の値を試してみてください。  
通常、妥当な値を見つけるのはそれほど難しくはありません。

## Interpolate ダイアログ

メニュー Analysis → Interpolate を選択すると、Interpolate ダイアログが表示され、希望する補間タイプ、ソースウェーブ、宛先ウェーブ、宛先ウェーブのポイント数を選択できます。

このダイアログは、実行、クリップボードへのコピー、コマンドラインへのコピーが可能な Interpolate2 コマンドを生成します。

Interpolation Type ポップアップメニューから、Linear（線形）、Cubic Spline（3次スプライン）、Smoothing Spline（平滑化スプライン）を選択します。

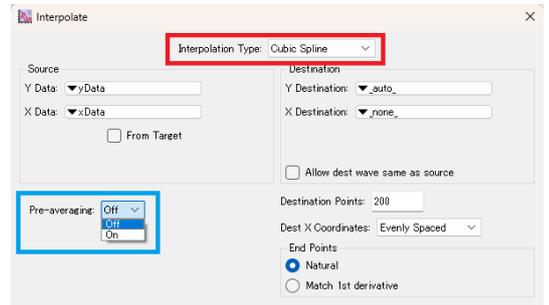
3次スプラインは、入力データセットが小さい場合に適しています。

平滑化スプラインは、入力データセットが大きく、ノイズが多い場合に適しています。

Cubic Spline を選択すると、Pre-averaging（事前平均化）ポップアップメニューが表示されます。

事前平均化機能は、もはやほとんど必要がないため、推奨されません。

事前平均化機能を使う場合は、Cubic Spline ではなく、Smoothing Spline を使ってください。

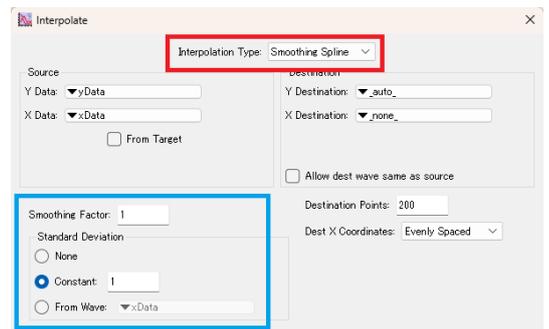


Smoothing Spline を選択すると、Smoothing Factor（平滑化係数）と Standard Deviation（標準偏差）コントロールが表示されます。

通常は、平滑化係数を 1.0 に設定し、標準偏差の設定には Constant モードを使うのが最適です。

次に、Y データのノイズの標準偏差の推定値を入力する必要があります。

そして、満足のいく平滑化スプラインが得られるまで、標準偏差の値を変えて試みます。

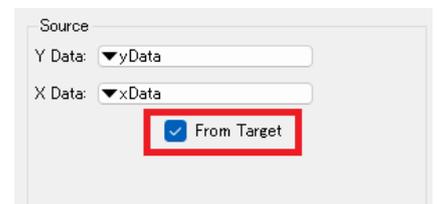


Y Data と X Data のポップアップメニューから、補間したいデータに定義するソースウェーブを選択します。

X Data ポップアップからウェーブを選択すると、Interpolate2 は、Y データウェーブの内容と X データウェーブの内容を比較して定義された XY 曲線をソースデータとして使います。

X Data ポップアップメニューから `_calculated_` を選択すると、Y データウェーブの X 値と Y 値がソースデータとして使われます。

From Target チェックボックスをクリックすると、Source と Destination セクションのポップアップメニューには、ターゲットのグラフまたはテーブル内のウェーブのみが表示されます。



X と Y のデータウェーブは、同じ数のデータポイントを持つ必要があります。同じデータタイプである必要はありません。

Interpolate2 を使う前に、X と Y のソースデータをソートする必要はありません。必要に応じて、Interpolate2 は補間を行う前に、入力データのコピーをソートします。

ソースデータ内の NaN（欠損値）と INF（無限大）は無視されます。

X または Y の値が NaN または INF であるデータポイントは、存在しないものとして扱われます。

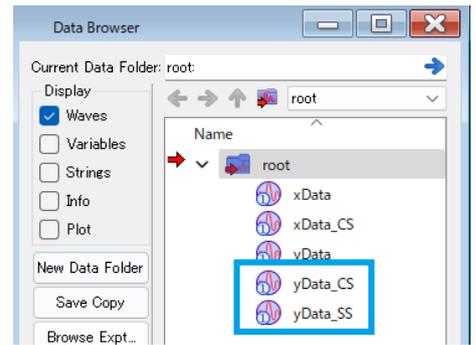
Destination Points ボックスに、宛先ウェーブに設定したいポイント数を入力します。通常は、200 ポイントでよいでしょう。

Y Destination と X Destination ポップアップメニューから、補間結果を格納するウェーブを選択します。ほとんどの場合、Y Destination ウェーブには `_auto_` を、X Destination ウェーブには `_none_` を選択します。これにより、出力ウェーブフォームが得られます。

Y Destination ポップアップメニューから `_auto_` を選択した場合、Interpolate2 は Y 出力データを、Y データウェーブの名前にサフィックスを追加した名前のウェーブに配置します。

サフィックスは、線形補間が「`_L`」、3次スプライン補間が「`_CS`」、平滑化スプライン補間が「`_SS`」です。

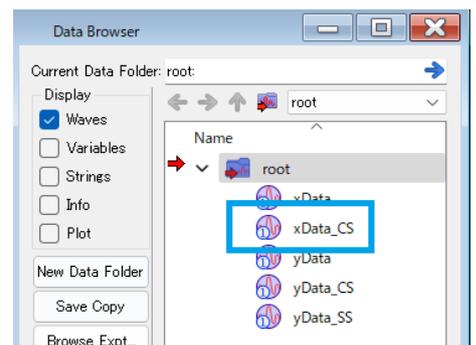
例えば、Y データウェーブが `yData` という名前の場合、デフォルトの宛先ウェーブは `yData_L`、または `yData_CS`、`yData_SS` と名付けられます。



X Destination ウェーブに `_none_` を選択した場合、Interpolate2 は、Y 出力データを Y Destination ウェーブに入れ、X 出力データを表現するために Y Destination ウェーブの X スケーリングを設定します。

X Destination ポップアップメニューから `_auto_` を選択した場合、Interpolate2 は X 出力データを、X データウェーブの名前に適切なサフィックスを追加して生成された名前のウェーブに配置します。

例えば、X データウェーブが `xData` という名前の場合、デフォルトの宛先ウェーブは `xData_L`、または `xData_CS`、`xData_SS` と名付けられます。

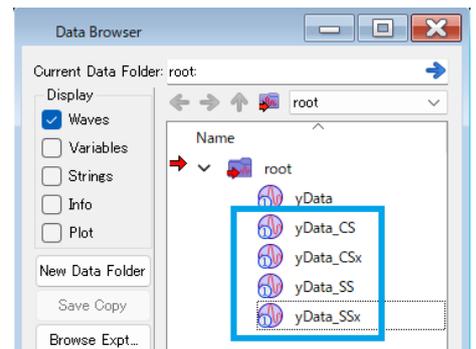


X データウェーブが存在しない場合、X Destination ウェーブ名は、Y Destination ウェーブ名の後に「`x`」を追加することで生成されます。

例えば、Y の宛先が `yData_CS` の場合、X の宛先ウェーブは `yData_CSx` となります。

X と Y の宛先ウェーブの両方で、ウェーブが既に存在する場合は、Interpolate2 はそれを上書きします。

まだ存在しない場合は、新規に作成します。



Interpolate2 が呼び出されたときに、すでに存在している場合を除き、宛先ウェーブは倍精度になります。

宛先ウェーブが単精度の場合、Interpolate2 は単精度のままにします。

その他の精度については、Interpolate2 は倍精度に変換します。

Dest X Coordinates ポップアップメニューでは、補間を実行する X 座標をコントロールできます。

通常は、Evenly Spaced を選択します。

これにより、X の入力値の範囲にわたって等間隔で補間値が生成されます。



Evenly Spaced + (Evenly Spaced Plus Input X Coordinates) の設定は Evenly Spaced と同じですが、Interpolate2 は出力 X 値が入力 X 値のすべてを含むようにします。

これは通常必要ありません。

X Destination ウェーブとして `_none_` を選択した場合は、このモードは使うことができません。

Log Spaced は、対数軸上に均等に間隔をあけて出力します。

このモードでは、入力 X データの正ではない値は無視されます。

これは、X Destination ウェーブに `_none_` を選択した場合は使うことができません。  
代わりの方法については、次のセクション「指数データの補間」を参照してください。

From Dest Wave 設定では、出力 X 値を宛先ウェーブの X 座標から取得します。

Destination Points 設定は無視されます。

例えば、入力データのサブセットにスプラインを適用するときこれを使うことができます。

このモードでは、補間を行う前に、宛先ウェーブを作成する必要があります。

宛先がウェーブフォームの場合、SetScale コマンドを使って、ウェーブフォームの X 値を定義します。

Interpolate2 は、これらの X 値で出力を計算します。

宛先が XY ペアの場合、X の宛先ウェーブの値を設定します。

Interpolate2 はこれらの値のソートされたバージョンを作成し、その値で出力を計算します。

X 宛先ウェーブがもともと逆ソートされていた場合、Interpolate2 は出力を逆ソートします。

End Points のラジオボタンは、3 次スプラインのみに適用されます。

これらは、ソースウェーブの最初の区間と最後の区間での宛先ウェーブをコントロールします。

スプラインの 2 階微分は、宛先ウェーブの最初と最後のポイントで、自然にゼロになります。

1 次導関数に一致させることで、スプラインの傾きを、最初の入力ポイントと 2 番目の入力ポイントの間、そして最後の入力ポイントと最後から 2 番目の入力ポイントの間に引かれた直線に一致させます。

ほとんどの場合、どちらを選択しても、あまり違いはありません。

## 指数データの補間

指数関数的なプロセスから生じるデータなど、桁が変わるデータは対数軸に対してプロットするのが一般的です。

このようなデータから補間データセットを作成するには、多くの場合、元のデータそのものではなく、元のデータの対数に対して補間を行うのが最善です。

Interpolate2 Log Demo (¥Igor Pro 9 Folder¥Examples¥Feature Demos¥ Interpolate2 Log Demo.pxp) では、このような補間を行う方法を示しています。

この Experiment は File メニューから選択することはできません。

クイックノート内のリンクをクリックするか、上記のパスの pxp ファイルを開いてください。

以下は、その Experiment のスニペットです：

```
Function DoLogYInterpolate2(xInput, yInput, numOutputPoints, xOutput, yOutput)
    Wave xInput, yInput
    int numOutputPoints
    Wave& xOutput          // 出力ウェーブの参照
    Wave& yOutput          // 出力ウェーブの参照

    // 補間で使う log(yInput) を作成
    Duplicate/FREE yInput, logYInput          // 関数が終了する時フリーウェーブを自動的に Kill
    logYInput = log(yInput)

    String xOutName = NameOfWave(xInput) + "_interp"
    String yOutName = NameOfWave(yInput) + "_interp"
    Interpolate2 /T=1 /I=0 /N=(numOutputPoints) /X=$xOutName /Y=$yOutName xInput, logYInput

    WAVE xOutput = $xOutName
    WAVE yOutput = $yOutName

    // yOutput は log(yInput) をもとに設定されたため、その変換を元のスケールに戻す
    yOutput = 10^yOutput
```

End

実際の動きは、Interpolate2 Log Demo で確認してください。

## Smoothing Spline (平滑化スプライン) のアルゴリズム

平滑化スプラインのアルゴリズムは、"Smoothing by Spline Functions", Christian H. Reinsch, Numerische Mathematik 10 に基づいています。

これは、与えられたポイントにおける平滑化スプラインの値を  $g(x_i)$ 、そのポイントにおける Y データを  $y_i$ 、そのポイントの標準偏差を  $\sigma_i$ 、平滑化係数を  $S$  とすると、すべての関数  $g(x)$  のうち、

$$\int_{x_0}^{x_n} g''(x)^2 dx$$

を満たす

$$\sum_{i=0}^n \left( \frac{g(x_i) - y_i}{\sigma_i} \right)^2 \leq S$$

を最小化します。

## Smoothing Spline (平滑化スプライン) のパラメーター

平滑化スプラインコマンドには、標準偏差パラメーターと平滑化係数パラメーターが必要です。標準偏差パラメーターは、Y データのノイズの標準偏差を適切に推定するものであるべきです。正確な標準偏差の推定値がある場合、平滑化係数は通常、1.0 に近い値であるべきです。

Interpolate ダイアログの Standard Deviation セクションでは、標準偏差パラメーターを3つのオプション (None、Constant、From Wave) から1つを選択できます。

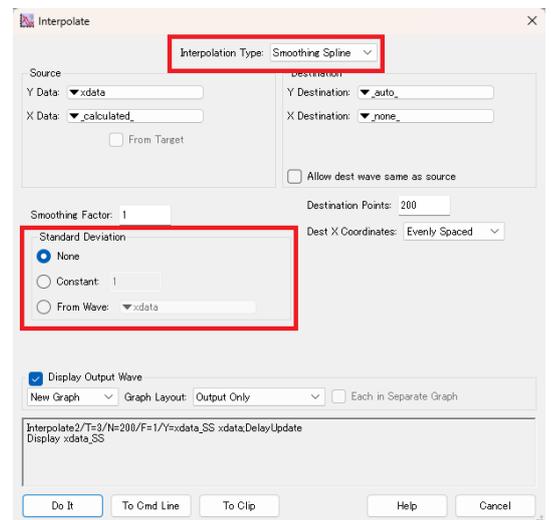
None を選択した場合、Interpolate2 は、Y データの振幅の 0.05 倍という任意の標準偏差推定値を使います。

その後、平滑化パラメーターを調整して、満足のいく平滑化スプラインが得られるまで試すことができます。

平滑化係数を 1.0 から開始します。

ただ、この方法は推奨されません。

Constant を選択した場合、ノイズの標準偏差の推定値を入力できるようになり、Interpolate2 は、この値を Y データの各ポイントの標準偏差として使います。



推定が正確であれば、1.0 前後の平滑化係数を使うことで、データに滑らかな曲線を描くことができます。

最初の試行がうまくいかなかった場合は、平滑化係数を 1.0 のままにして、標準偏差の別の推定値を試してみます。

ほとんどの種類のデータでは、この方法が推奨されます。

From Wave を選択した場合、Interpolate2 は、指定したウェーブ内の各ポイントに、Y データの対応するポイントの推定標準偏差が含まれていることを想定します。

適切なウェーブがある場合は、この方法を使ってください。

## Interpolate2 の Pre-averaging (事前平均化) 機能

線形または3次スプライン補間は、すべての入力データポイントを通過します。

もし、サイズが大きく、ノイズの多いデータセットに対して使うのであれば、これはおそらく目的としている結果とは異なると思います。

その場合は、平滑化スプラインを使ってください。

Interpolate2 に平滑化スプラインが実装される前は、入力データの縮小版を補間するために3次スプラインを使用することを推奨していました。

事前平均化機能は、これを簡単に行うために設計されました。

Interpolate2 が平滑化スプラインをサポートするようになったため、事前平均化機能は不要になりました。ただし、後方互換性を維持するために、引き続きサポートしています。

Pre-averaging をオンにすると、Interpolate2 は入力データの仮のコピーを作成し、縮小化によってノードと呼ばれるより少ない数のデータポイントに削減します。

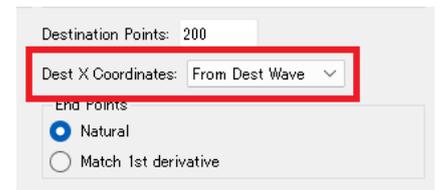
Interpolate2 は通常、データの最初と最後の部分にノードを追加します。

最後に、これらのノードで補間を行います。

## Destination Wave から宛先 X 座標を取得

このモード (X From Dest モードと呼びます) は、Interpolate ダイアログの Destination X Coordinates ポップアップメニューから From Dest Wave を選択するか、Interpolate2 /I=3 フラグを使うと有効になります。

このモードでは、出力ポイントの数は宛先ウェーブによって決定され、/N フラグは無視されます。



X From Dest モードでは、補間が行われるポイントは宛先ウェーブによって決定されます。

宛先がウェーブフォームの場合、補間はその X 値で行われます。

あるいは、宛先が XY ペアの場合、補間は X の宛先ウェーブに格納されたデータ値で行われます。

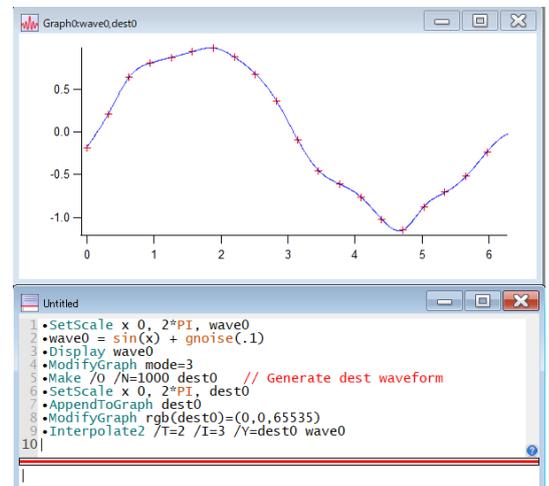
ウェーブフォームを宛先として使用する例を以下に示します。新しい Experiment で次のコマンドを実行します。

```
Make /O /N=20 wave0 // ソースデータを生成
SetScale x 0, 2*PI, wave0
wave0 = sin(x) + gnoise(.1)
Display wave0
ModifyGraph mode=3
```

```
Make /O /N=1000 dest0 // 宛先ウェーブフォームを生成
SetScale x 0, 2*PI, dest0
AppendToGraph dest0
ModifyGraph rgb(dest0)=(0,0,65535)
```

```
// 3次スプライン補間を実行
```

```
Interpolate2 /T=2 /I=3 /Y=dest0 wave0
```



宛先が XY ペアの場合、Interpolate2 はこれらの値のソートされたバージョンを作成し、その出力を X 宛先ウェーブの値で計算します。

X 宛先ウェーブが、その最初の値と最後の値を調べることによって決定されるように、もともと逆ソートされていた場合、Interpolate2 は補間処理後に出力を反転させて、元の順序を復元します。

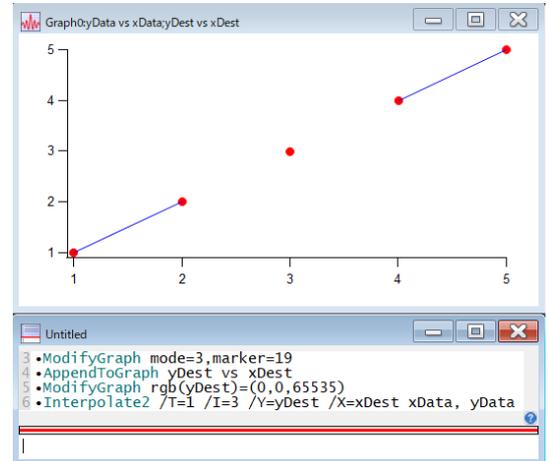
X From Dest モードで XY ペアを宛先として使う場合、X ウェーブには NaN が含まれる場合があります。この場合、Interpolate2 は XY ペアの内部コピーを作成し、X 宛先が NaN のポイントを削除し、補間を実行し、その結果を宛先 XY ペアにコピーします。この最後のステップでは、Interpolate2 が NaN を X の宛先ウェーブの元の位置に戻します。

以下は X 宛先ウェーブに NaN がある XY 宛先の例です。新しい Experiment で次のコマンドを実行します。

```
// ソースデータを生成
Make/O xData={1,2,3,4,5}, yData={1,2,3,4,5}
Display yData vs xData

// 宛先 XY ペアを生成
Make/O xDest={1,2,NaN,4,5}, yDest={0,0,0,0,0}
ModifyGraph mode=3,marker=19
AppendToGraph yDest vs xDest
ModifyGraph rgb(yDest)=(0,0,65535)

// 線形補間を実行
Interpolate2 /T=1 /I=3 /Y=yDest /X=xDest xData, yData
```



## 微分と積分

微分 (Differentiate) と積分 (Integrate) のコマンドでは、1次元ウェーブフォームと XY データの操作に利用できる多数のアルゴリズムが提供されています。

これらのコマンドは、元のデータを置き換えるか、結果を新しいウェーブとして作成することができます。

これらの操作を最も簡単に実行する方法は、Analysis メニューから開くことができるダイアログを使うことです。

ほとんどの用途では、台形積分と中心差分が適切な方法です。しかし、XY データで操作を行う場合、アルゴリズムによって X ウェーブのポイント数の要件が異なります。ダイアログに X ウェーブが表示されない場合は、別のアルゴリズムを選択するか、ヘルプボタンをクリックして要件を確認してください。

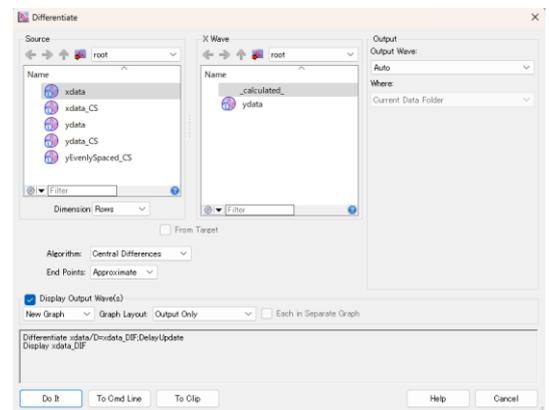
ウェーブフォームデータで操作を行う場合、X スケーリングが考慮されます。

この機能を無効にするには、/P フラグを使います。

SetScale コマンドを使って、Y データウェーブの X スケーリングを定義することができます。

これらのコマンドは 1次元に沿ってのみ機能しますが、/DIM フラグを使うことで、行列の行または列（あるいはさらに高次元）に沿って動作するようにターゲットを絞ることができます。

積分コマンドは、数値積分でウェーブを置き換えたり作成したりします。曲線下の面積を求めるには、次の「area 関数と mean 関数」を参照してください。



## area 関数と mean 関数

Igor を使うと、ウェーブの面積と平均値をいくつかの方法で計算することができます。

おそらく、平均値を計算する最も簡単な方法は、Statistics メニューの Wave Stats ダイアログを使うことです。

ウェーブを選択し、X の範囲を入力（または現在のカーソル位置を使用）し、Do It をクリックすると、履歴エリアにいくつかの統計結果を出力します。

その中に V\_avg（平均値または中間値）があります。

これは、Igor の平均関数によって返される値と同じであり、他の統計値を計算しないため、より高速です。

指定範囲内のデータに NaN が含まれる場合、平均関数は NaN を返します。

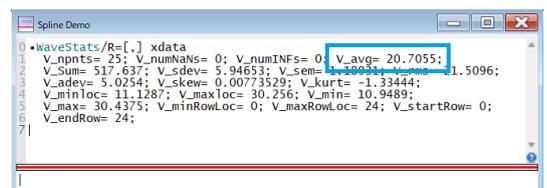
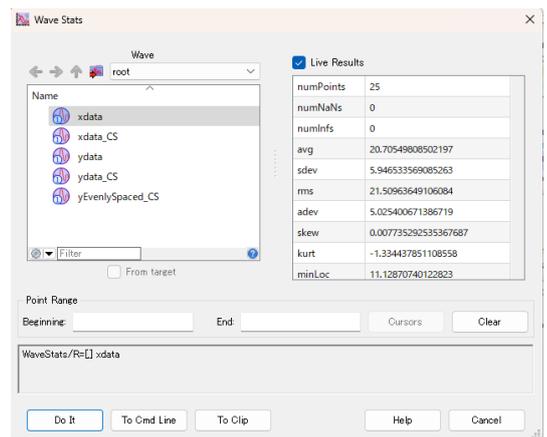
一方、WaveStats の処理では、このような欠損値は無視されます。

WaveStats と mean 関数は、平均値を計算する際に同じ方法を使います。

指定された X の範囲内のウェーブフォーム値を見つけ、それらを合計し、値の数で割ります。

X 範囲は、組み合わせる値を選択するだけです。

範囲は、最も近いポイント番号に丸められます。



イベントのカウントなど、離散的な値を記述するデータであると考えられる場合は、平均値を計算するには WaveStats または mean 関数を使うべきです。

合計イベント数は、ある種の面積であり、sum 関数を使うことで最も簡単に計算できます。

また、WaveStats の出力 V\_avg と V\_npnts を乗算することでも簡単に計算できます。

データが、サンプリングされたオーディオ信号のような連続的なプロセスのサンプリングされたものである場合、平均値を計算するには favarage 関数、面積を計算するには area 関数を使います。

この2つの関数は、データポイント間のウェーブフォーム値を推定するために、台形積分と同じ線形補間スキームを使っています。

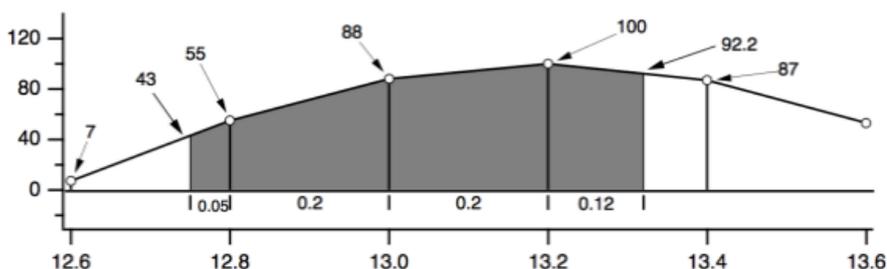
X の範囲は、最も近いポイントに丸められていません。

この線形補間により、X の範囲の一部が計算に含まれます。

下記の図は、表示されているデータに対して各関数が実行する計算を示しています。

43 と 92.2 の2つの値は、線形補間による推定値です。

この図は、区間 (12.75,13.32) における area、favarage、mean 関数を比較しています。



$$V\_avg = (55+88+100+87)/4 = 82.5$$

$$\text{mean}(\text{wave},12.75,13.32) = (55+88+100+87)/4 = 82.5$$

$$\begin{aligned} \text{area}(\text{wave},12.75,13.32) &= 0.05 \cdot (43+55)/2 && // \text{最初の台形} \\ &+ 0.20 \cdot (55+88)/2 && // \text{2番目の台形} \\ &+ 0.20 \cdot (88+100)/2 && // \text{3番目の台形} \\ &+ 0.12 \cdot (100+92.2)/2 && // \text{4番目の台形} \\ &= 47.082 \end{aligned}$$

$$\text{faverage}(\text{wave}, 12.75, 13.32) = \text{area}(\text{wave}, 12.75, 13.32) / (13.32 - 12.75) \\ = 47.082 / 0.57 = 82.6$$

ウェーブの X スケーリングの影響を受けるのは area 関数のみであることを注意してください。

faverage は、面積を、その面積が乗算される X 範囲と同じ X 範囲で割ることにより、X スケーリングの影響を排除します。

これらの関数には、指定されたデータ範囲に欠損値 (NaN) が存在する場合に使用できないという問題があります。

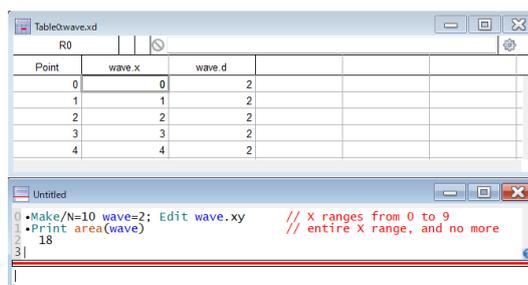
## X 範囲と mean、faverage、area 関数

mean、faverage、および area 関数の X 範囲入力はオプションです。

したがって、ウェーブ全体を含めるには、範囲を指定する必要はありません。

```
// 0 から 9 までの X 範囲
Make/N=10 wave=2; Edit wave.xy
// 全範囲
Print area(wave)
```

出力 : 18



プログラミングでは、ある範囲がウェーブの端を越えているかどうかを判断することが不便な場合があります。幸い、これらの関数は、ウェーブの端を越えている X の範囲も受け入れます。

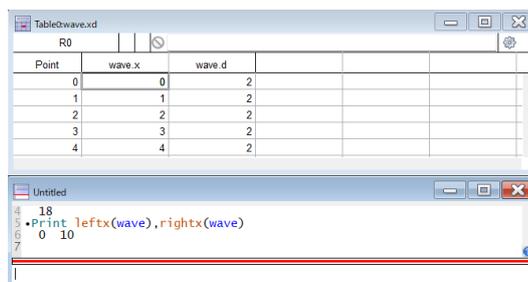
ウェーブの両端を越える範囲を評価する式を使うこともできます。

```
Print leftx(wave), rightx(wave)
```

出力 : 0 10

```
// 全範囲
Print area(wave, leftx(wave), rightx(wave))
```

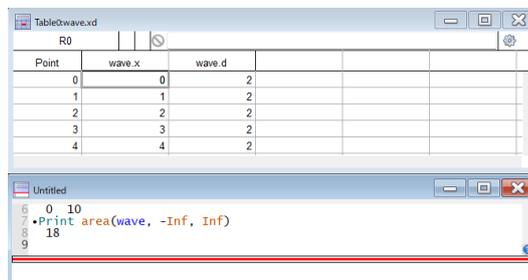
出力 : 18



$\pm\infty$  の X 範囲を指定することもできます。

```
// 全範囲
Print area(wave, -Inf, Inf)
```

出力 : 18



## ウェーブのセグメントの平均値を求める

解析プログラミング (Analysis Programming) は、指定した長さのセグメントのウェーブの平均値を求める機能です。

これは各セグメントの平均値を含む新しいウェーブを作成します。

ヘルプ Analysis.ihf の Analysis Programming を参照してください。

## XY データの面積

XY ペアのウェーブに含まれるデータ領域の面積を計算するには、areaXY 関数を使います。

faverage 関数の XY バージョンもあります。

詳細は faverageXY の説明を参照してください。

さらに、WaveMetrics のプロシージャファイル「AreaXYBetweenCursors」には、「AreaXYBetweenCursors」と「AreaXYBetweenCursorsLessBase」のプロシージャが含まれています。

プロシージャファイルのロード方法については、WaveMetrics プロシージャフォルダーについての説明を参照してください。

Info Panel と Cursor を使って、面積を計算する X の範囲を区切ることができます。

AreaXYBetweenCursorsLessBase は、カーソル間の直線、単純な台形ベースラインを削除します。