

CONTENTS

Igor Pro ガイドツアー（1-3 - より複雑なグラフ2）	2
カーソルの使い方	2
トレースと軸の削除	4
積み重ねた軸を持つグラフの作成	5
2つのグラフのページレイアウト	8
コントロールの作成	10
依存関係の作成	16

Igor Pro ガイドツアー（1-3 – より複雑なグラフ2）

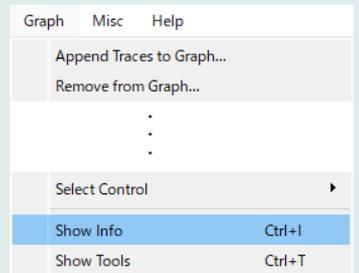
カーソルの使い方

前のセクションの続きから、カーソルの使い方を説明します。グラフウィンドウがアクティブになっているところからです。

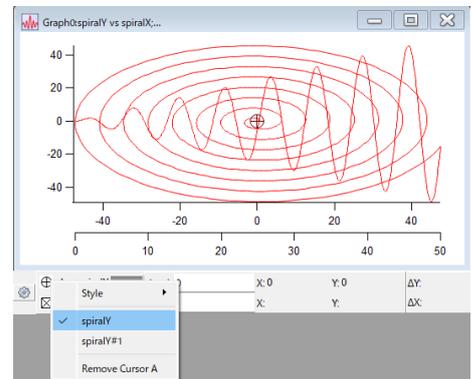
閉じてしまっている場合は、Tour1C.pxp を開きます。

1. メニュー Graph → Show Info を選択します。

グラフの下に、カーソル情報パネルが表示されます。



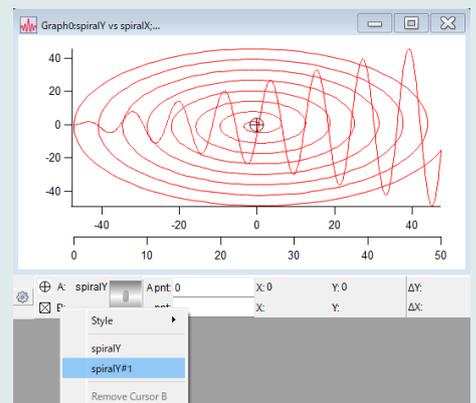
2. パネル内のカーソル A を右クリックして、spiralY を選択します。



3. パネル内のカーソル B を右クリックして、spiralY#1 を選択します。

ウェーブ spiralY は2回グラフ化されています。

2番目のインスタンスを1番目と区別するために #1 のサフィックスが使われています。Igor のインデックスはゼロから始まるため、#2 ではなく #1 となっています。

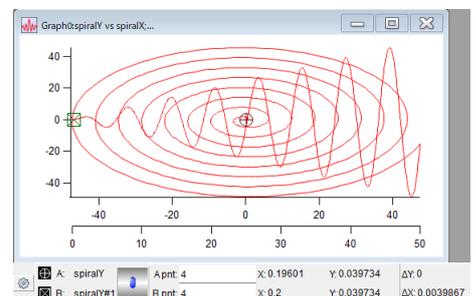


4. カーソル位置コントロールの中心 にマウスポインターを移動します。

青いスライダーを少しだけ右にドラッグします。

スライダーの移動に合わせて数値が変わり、グラフ内のカーソルも移動します。

キーボードの左右矢印キーでも移動させることができます。スライダーが中央から離れるほど、カーソルが高速で移動することに注目してください。



5. カーソル A のアイコンをクリックします。背景が白くなり、選択されていないことを示しています。



この状態でスライダーを左右に動かすと、カーソル B だけ変化していることがわかります。

6. グラフ内のカーソル B をマウスで別の場所に移動します。

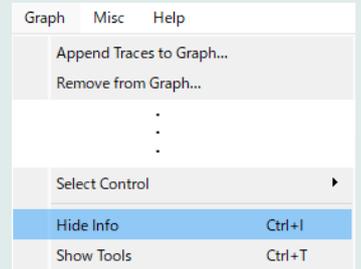
カーソルが示す数値が変わります。

⊕ A: spiralY	A pnt: 4	X: 0.19601	Y: 0.039734	ΔY: 1.4564
⊗ B: spiralY#1	B pnt: 50	X: 2.5	Y: 1.4962	ΔX: 2.304

カーソル A を完全にグラフの外に移動します。

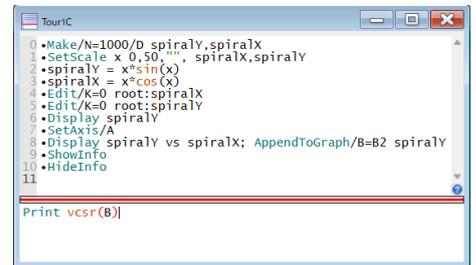
カーソルはグラフから消え、数値も表示されなくなります。

7. メニュー Graph → Hide Info を選択し、パネルを隠します。



8. コマンドウィンドウをクリックし、次を入力して Enter キーを押します。

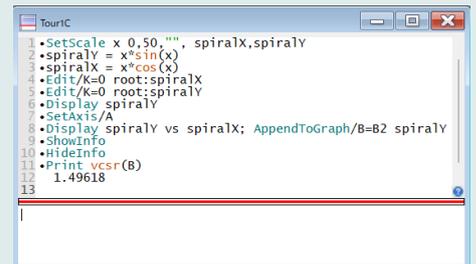
```
Print vcsr(B)
```



9. カーソル B の Y 値が履歴領域に表示されます。

カーソルに関する情報を取得するための多くの機能が実装されています。

カーソル B を完全にグラフの外まで移動し、非表示にします。



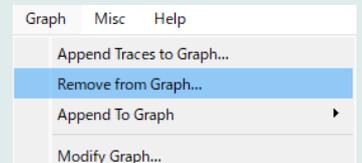
トレースと軸の削除

複数の軸を持つグラフから、トレースや軸を削除します。
前のセクションのグラフウィンドウがアクティブになっているところからです。

1. メニュー Graph → Remove from Graph を選択します。

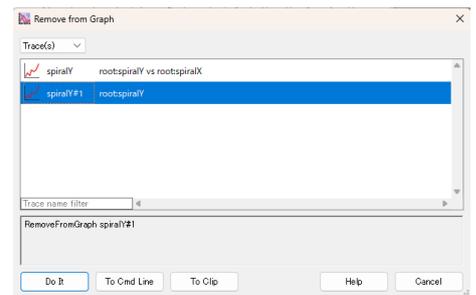
Remove From Graph ダイアログに spiralY が 2 回表示されます。

グラフを作成したときに、スパイラルを作るために spiralY を 1 回、正弦ウェーブを表示するために spiralY と計算された X 値の組み合わせで、2 回使用しました。



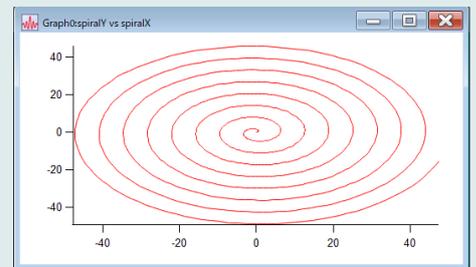
2. spiralY の 2 番目のインスタンス (spiralY#1) をクリックして、Do It をクリックします。

そのトレースに対する軸も併せて削除されます。



3. X 軸を下にドラッグして、ウィンドウの外までドラッグします。

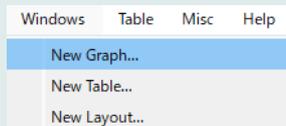
これにより、余白設定が auto に戻ります。
以前のステップで Alt キーを押しながら余白をドラッグしたときに、これを固定位置に設定していたためです。



積み重ねた軸を持つグラフの作成

グラフを縦に積み重ねたグラフを作成します。データはこれまでのものを使います。あとでレイアウトに使うため、これまでに作成したグラフも表示しておきます。

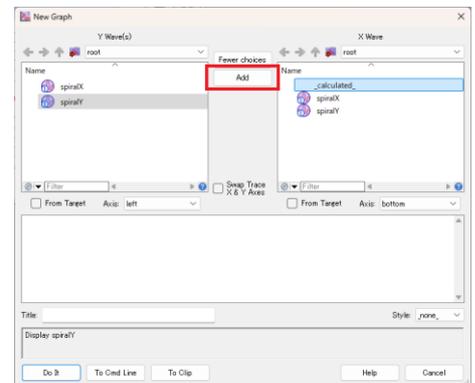
1. メニュー Windows → New Graph を選択します。



2. ボタンのラベルは More Choices になっている場合は、これをクリックします。

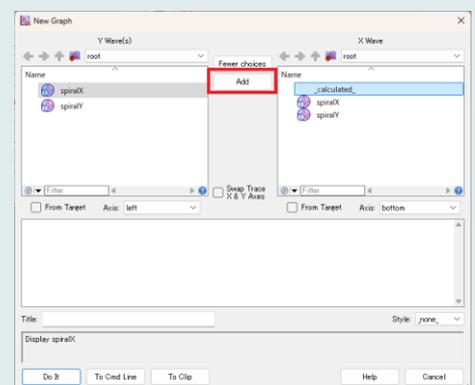
Y Waves のリストで spiralY を選択します。

X Waves のリストで _calculated_ を選択し、Add をクリックします。

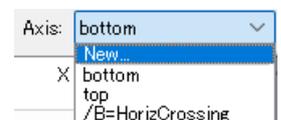


3. Y Waves のリストで spiralX を選択します。

X Waves のリストで _calculated_ を選択し、Add をクリックします。

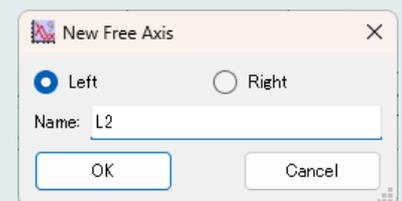


4. Y Waves リストの下の Axis のポップアップメニューで New を選択します。



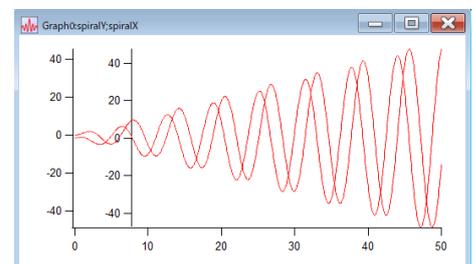
5. Name のボックスに L2 と入力します。

OK をクリックします。



6. Do It をクリックすると、次のグラフが作成されます。

以下のステップでは、左の軸の上に L2 軸を積み重ねます。

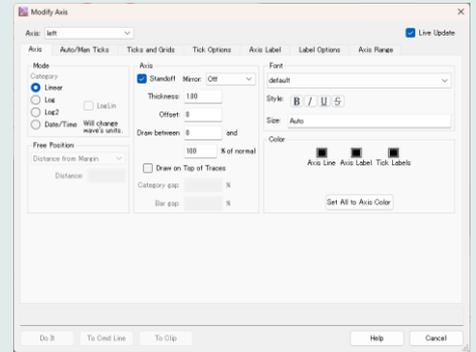


7. 一番左の Y 軸をダブルクリックします。

Modify Axis ダイアログが表示されます。

他のダイアログが表示されたときは、キャンセルして、カーソルが軸の上にあることを確認してからもう一度試してください。

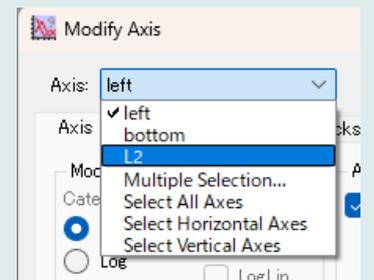
ダイアログ左上にある Axis ポップアップメニューで Left 軸を選択してください。



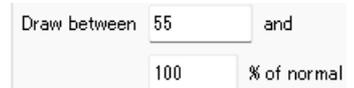
8. Axis タブをクリックし、中央の Draw between を 0 and 45% of normal に設定します。



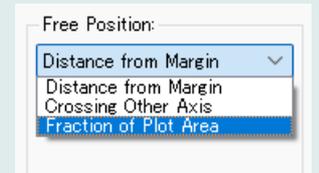
9. Axis ポップアップメニューから L2 を選択します。



10. Draw between を 55% and 100% of normal に設定します。



11. Free Axis Position でポップアップメニューから Fraction of Plot Area を選択します。



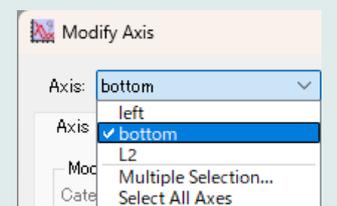
12. % of Plot Area と表示されたボックスがゼロに設定されていることを確認してください。

直前の操作では、L2 軸を左軸と一直線になるように移動しています。

これをデフォルトに設定していないのはなぜですか？

良い質問です。プロット領域に対するパーセント表示は、Igor Pro 6 で追加されました。デフォルトの動作は広報互換性を維持しているためです。

13. Axis ポップアップメニューから bottom を選択します。

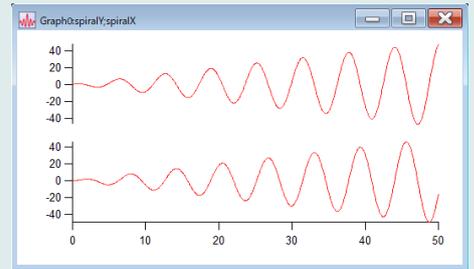


14. 中央の Standoff のチェックボックスをオフにします。



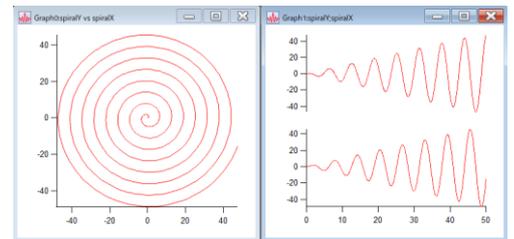
15. Do It をクリックします。

次のグラフが表示されます。



16. 以前に作成した Graph0 と今回作成した Graph1 の大きさをそろえて横に並べます。

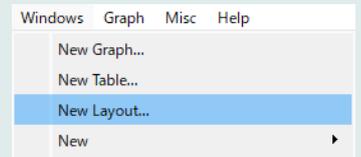
次のようになったと思います。



2つのグラフのページレイアウト

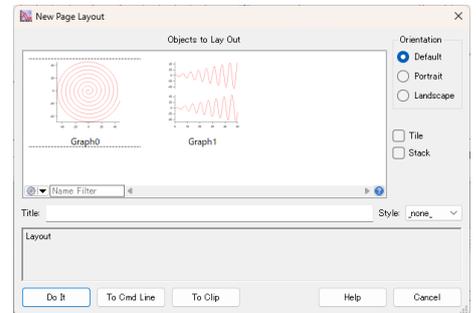
ここまでに作成した2つのグラフをページにレイアウトします。

1. メニュー Windows → New Layout を選択します。



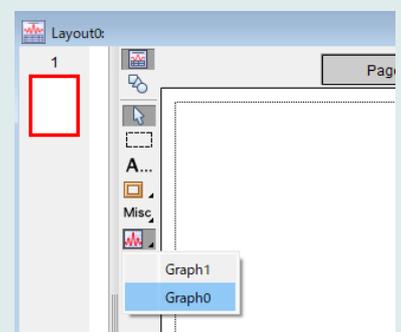
2. Do It をクリックします。

新しい空白のページレイアウトウィンドウが作成されます。



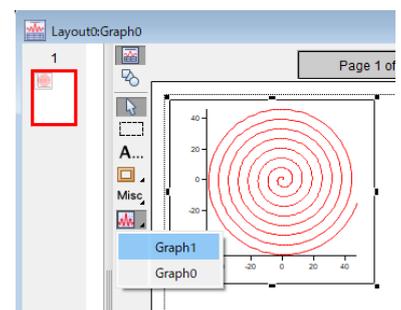
3. グラフアイコンをクリックし、メニューから Graph0 を選択します。

Graph0 がレイアウトに追加されます。



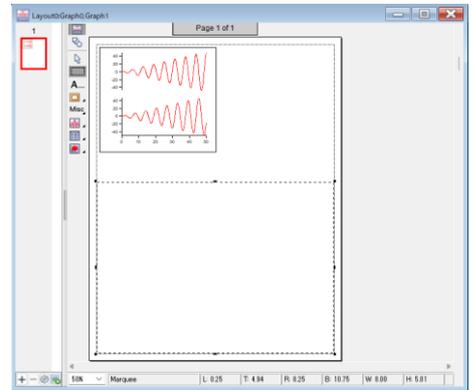
4. もう一度、グラフアイコンをクリックし、メニューから Graph1 を選択します。

Graph1 がレイアウトに追加されます。



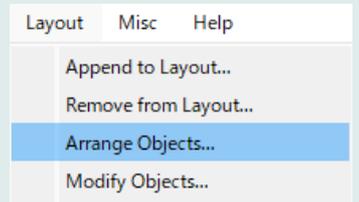
5. 矩形選択アイコン をクリックします。

6. ドラッグして、ページレイアウトの下半分を囲みます。

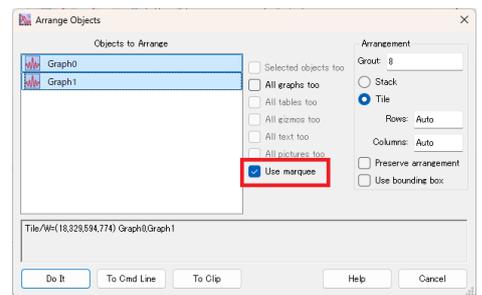


7. メニュー Layout → Arrange Objects を選択します。

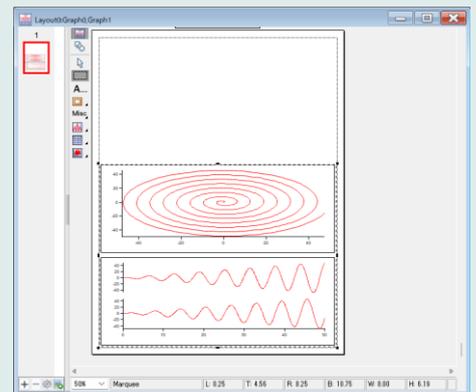
Arrange Objects ダイアログが表示されます。



8. Graph0 と Graph1 の両方を選択します。Use marquee チェックボックスをオンにします。

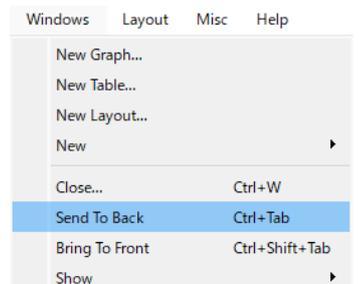


9. Do It をクリックすると、次のように2つのグラフがページの矩形枠内に配置されます。



10. ページ内の矩形範囲の外でクリックすると、選択枠が解除されます。

メニュー Windows → Send to Back を選択して、レイアウトウィンドウを後ろにまわします。



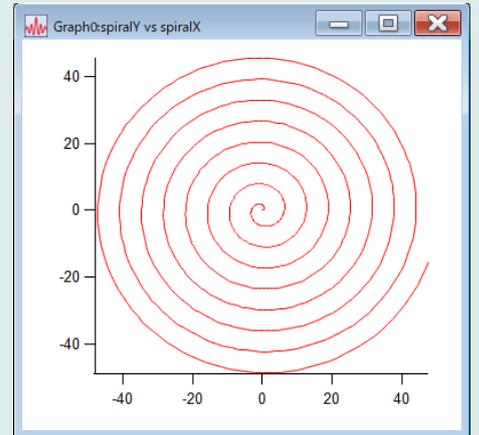
11. ここまでの作業を別名で保存します。

メニュー Files → Save Experiment As を選択して、"Tour1D.pxp" として保存します。

コントロールの作成

ここでは Igor のグラフにコントロールを追加します。
Tour1D.pxp が開いているところから始めます。

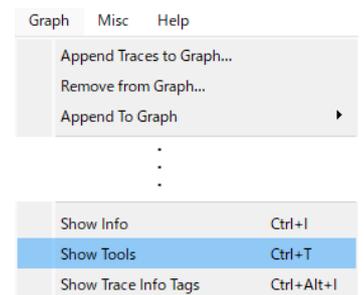
1. Graph0 (渦巻) のウィンドウをフロントにもってきます。



2. メニュー Graph → Show Tools を選択するか、Ctrl+T キーを押します。

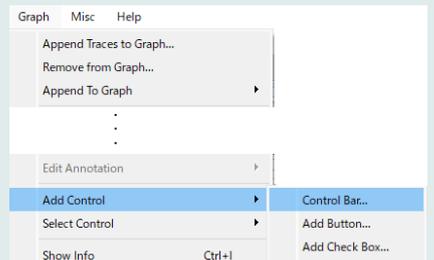
グラフの左側にツールパレットが表示されます。2 番目のアイコンが選択されているため、グラフは通常モードではなく、描画モードになっています。

セレクトーツール (矢印) がアクティブになっています。
これはコントロールの作成、選択、移動、サイズ変更に使います。



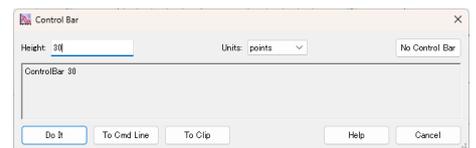
3. メニュー Graph → Add Control → Control Bar を選択します。

Control Bar ダイアログが表示されます。



4. Height を 30 points に設定し、Do It をクリックします。

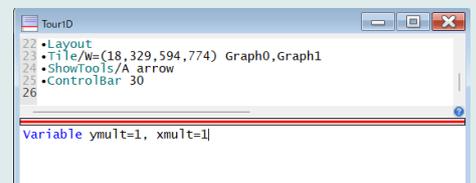
これにより、グラフの一番上にコントロール用のスペースが確保されます。



6. コマンドラインに次のコマンドを入力して Enter キーを押してください。

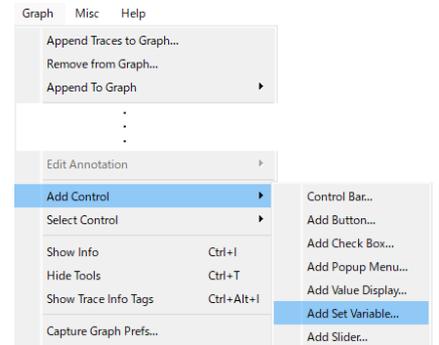
```
Variable ymult=1, xmult=1
```

これは2つの変数 (numeric) を作成し、両方を 1.0 に設定します。



6. Graph0 のウィンドウをクリックして、メニュー Graph → Add Control → Add Set Variable を選択します。

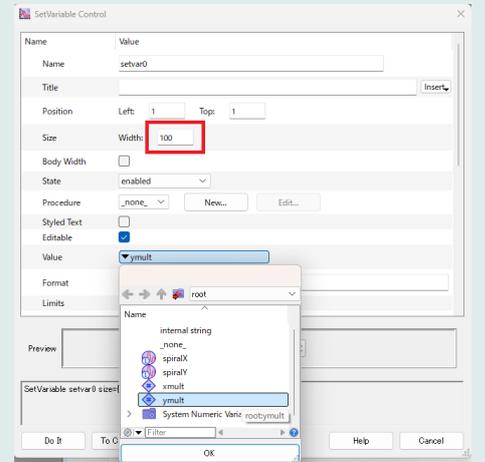
SetVariable Control ダイアログは表示されます。SetVariable コントロールは、変数の値を表示および変更する方法を提供します。



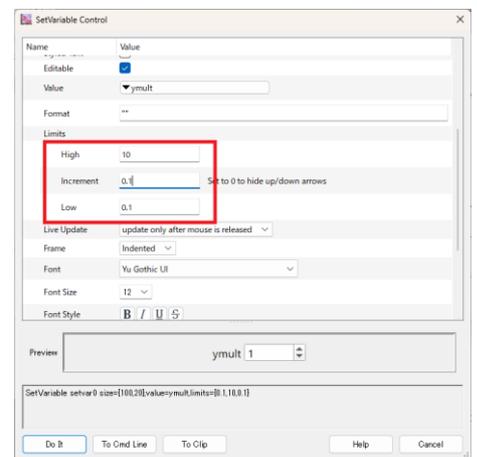
7. SetVariable Control ダイアログの Value で ymult を選択します。

Width のボックスに 100 を入力します。

この設定はスクロールリストの一番上に近い位置に戻っています。



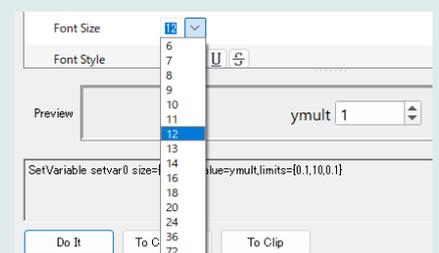
8. ダイアログを下にスクロールして、Limits で High, Low, Increment をそれぞれ 10, 0.1, 0.1 に設定します。



9. Font Size ポップアップメニューで 12 を選択します。

画面下部にプレビューが表示されるので、どのような表示になるかを確認できます。

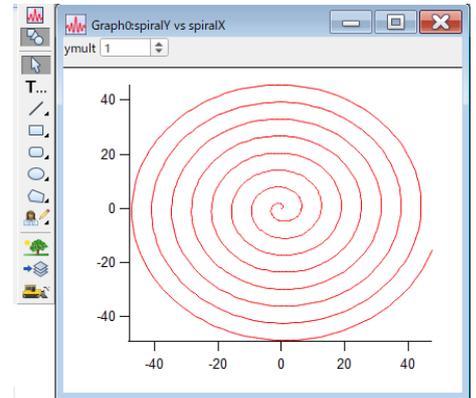
Do It をクリックします。



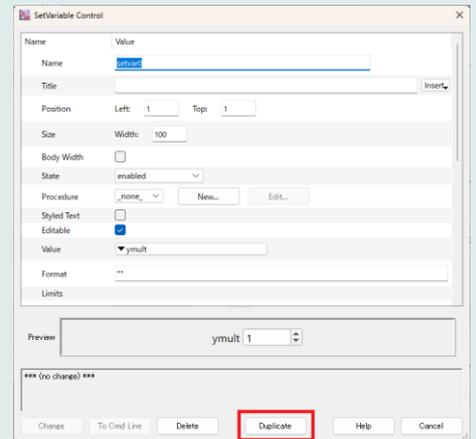
10. 変数 ymult にアタッチされた SetVariable コントロールがコントロールバーの左上に表示されます。

この画面の ymult をダブルクリックします。

SetVariable Control ダイアログが表示されます。

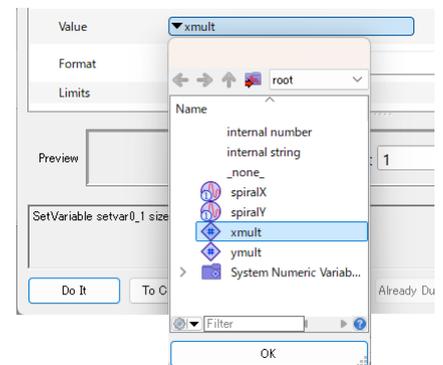


11. 画面下部の Duplicate をクリックします。



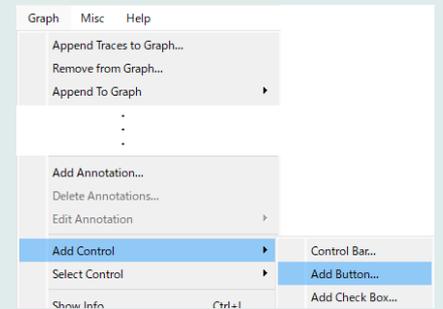
12. SetVariable Control ダイアログの Value で xmult を選択して、Do It をクリックします。

2つ目の SetVariable コントロールがコントロールバーに表示されます。これは xmult 変数にアタッチされています。



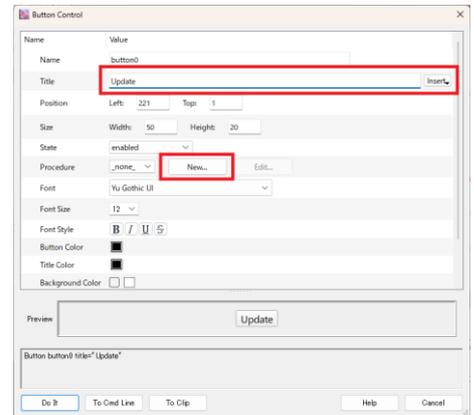
13. メニュー Graph → Add Control → Add Button を選択します。

Button Control ダイアログが表示されます。



14. Title の編集ボックスに Update を入力します。

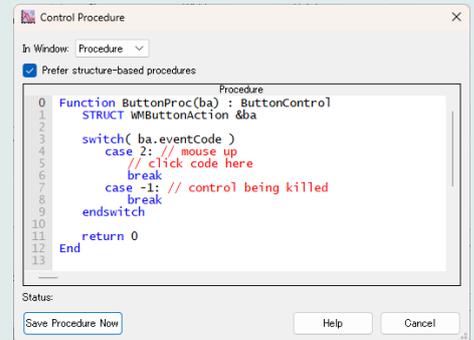
Procedure の項の New ボタンをクリックします。



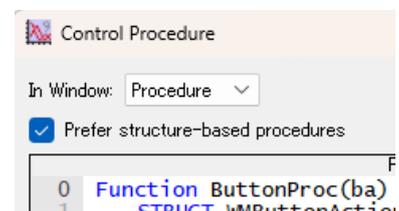
15. Control Procedure ダイアログが表示されます。

このダイアログは、コントロールに関するイベントが発生したとき（今回はボタンを押したとき）に呼び出されるプロシーダを作成、編集するためのものです。

このようなプロシーダを「Control Action Procedure」と呼びます。



16. Prefer structure-based procedures がチェックされていることを確認してください。



17. プロシーダのテキストを次のように書き換えてください。

```
Function ButtonProc(ba) : ButtonControl
    STRUCT WMBUTTONACTION &ba

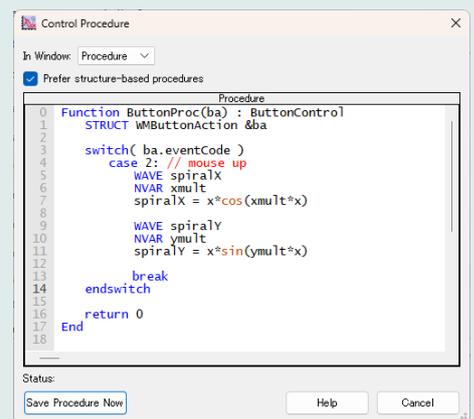
    switch( ba.eventCode)
        case 2: // Mouse up
            WAVE spiral
            NVAR xmult
            spiralX = x*cos(xmult*x)

            WAVE spiraly
            NVAR ymult
            spiraly = x*sin(ymult*x)
        break
    endswitch

    return 0
End
```

左下の Save Procedure Now ボタンをクリックしてください。

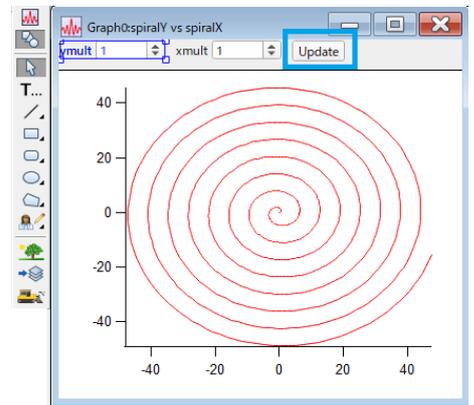
Control Procedure ダイアログが消え、編集したテキストが（現在非表示になっている）プロシーダウィンドウに挿入されます。



18. Button Control ダイアログに戻るので、**Do It** をクリックします。

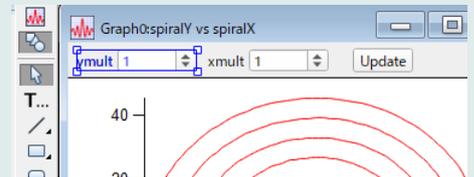
ボタンコントロールがコントロールバーに追加されます。

3つのコントロールは動作しますが、見た目が整然と配置されているわけではありません。



19. 各コントロールの位置を調整するには、ツールパレットから矢印  **を選択し、位置を調整します。**

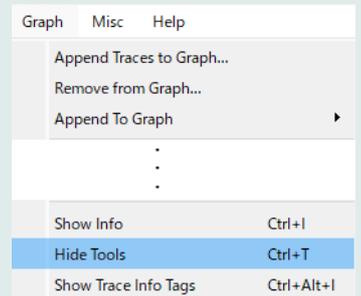
ドラッグするか、キーボードの矢印キーを使って、その位置を微調整することができます。



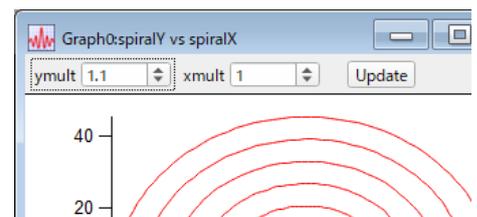
20. ツールパレットの一番上のアイコン  **をクリックして、動作モードに入ります。**

21. メニュー Graph → Hide Tools を選択します。

ツールパレットが表示されなくなります。

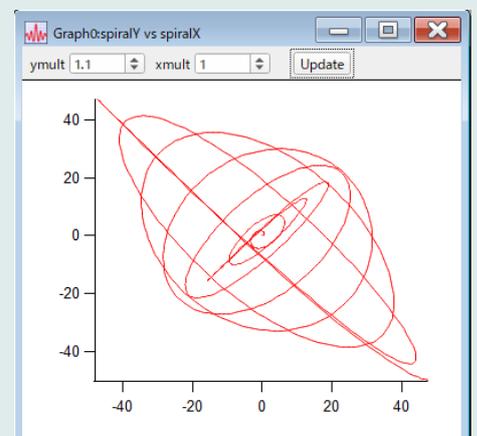


22. ymult の上矢印をクリックして、値を 1.1 にします。



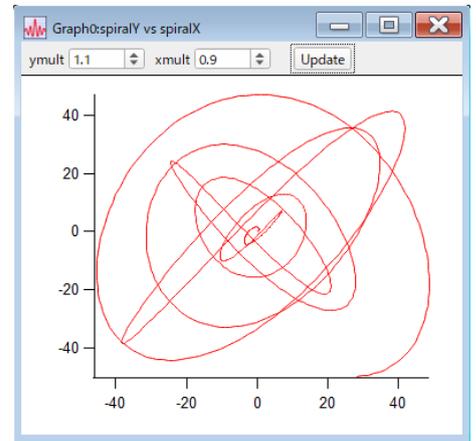
23. Update ボタンをクリックします。

これにより、作成したプロシージャが実行され、spiralX と spiralY が再計算され、グラフが更新されます。



24. ymult、xmult の値を変えていろいろ試してみてください。

SetVariable コントロールに値を入力して編集し、Enter キーを押して入力します。



25. ymult、xmult の値を 1.0 にします。

Update ボタンをクリックして元に戻します。

依存関係の作成

依存関係とは、Igor のウェーブや変数の値が、他のウェーブや変数の値に関係するルールを示すものです。依存関係を設定すると、あるウェーブ変数が増化したとき、自動的に別のウェーブが更新されます。データは前のセクションのものをそのまま使います。

1. コマンドウィンドウをフロントにもってきます。

コマンドラインに次を入力します。

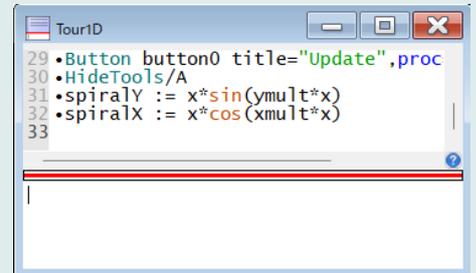
```
spiralY := x*sin(ymult*x)
spiralX := x*cos(xmult*x)
```

これは以前に作成したものと似ていますが、`=` が `:=` になっています。

`:=` は依存式を示します。

最初の式は、ウェーブ `spiralY` は変数 `ymult` に依存することを示します。

新しい値が `ymult` に保存されると、`spiralY` の値が自動的に再計算されます。



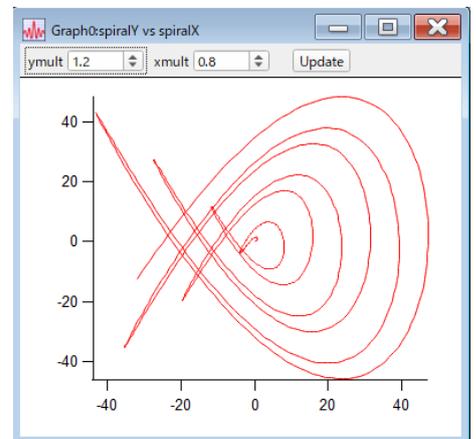
```
Tour1D
29 •Button button0 title="Update",proc
30 •HideTools/A
31 •spiralY := x*sin(ymult*x)
32 •spiralX := x*cos(xmult*x)
33
```

2. Graph0 ウィンドウをフロントに持ってきます。

ymult と xmult の値を変化させてみてください。

このとき、Update ボタンをクリックしないでください。

`ymult` と `xmult` の値を変えると、自動的に `spiralY` と `spiralX` の値が再計算され、グラフが更新されます。



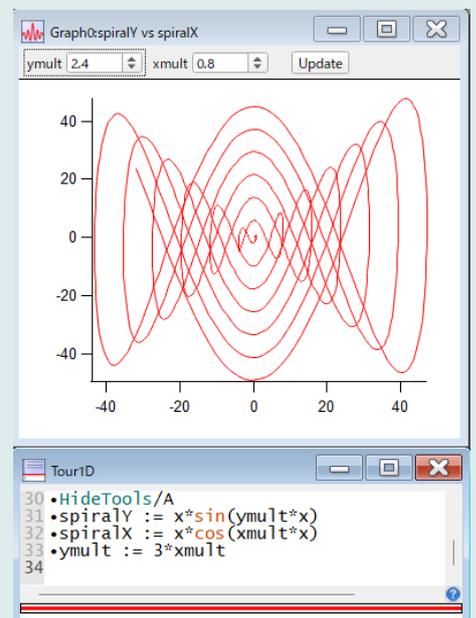
3. コマンドラインに次を入力します。

```
ymult := 3*xmult
```

`ymult SetVariable` コントロールとグラフが更新されます。

xmult の値を変えてみてください。

`ymult` の値が変わるだけでなく、グラフも変わることが確認できます。



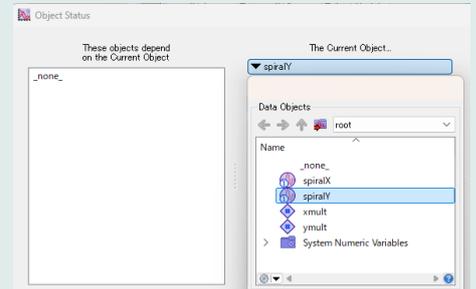
4. メニュー Misc → Object を選択します。

Object Status ダイアログが表示され、ここで依存関係を確認することができます。



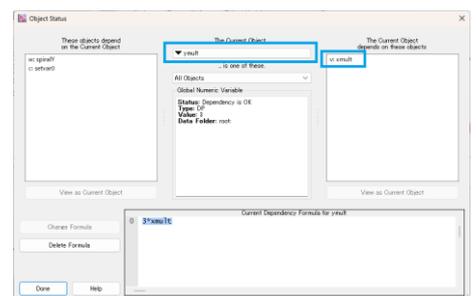
5. 中央の Current Object ポップアップメニューで、データオブジェクトのリストから spiralY を選択します。

右側のリストで spiralY が変数 ymult に依存していることがわかります。



6. 右側のリストで ymult をダブルクリックします。

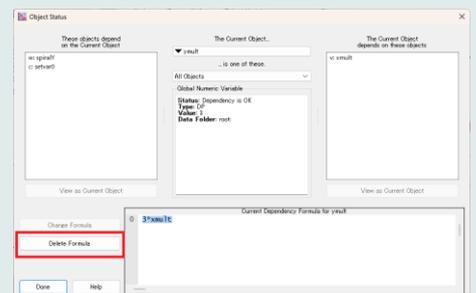
ymult が Current Object になり、ymult が xmult に依存していることがわかります。



7. 左下の Delete Formula をクリックします。

これにより、ymult が xmult に依存しなくなりました。

Done をクリックします。



8. グラフ内のコントロールで xmult を変えてみます。

ymult は変化せず、xmult に依存しなくなったことがわかります。

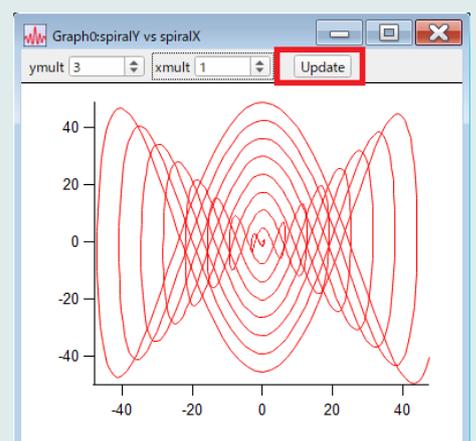
9. グラフ内のコントロールで Update をクリックしてみます。

xmult と ymult を変えてみてください。

spiralX と spiralY が更新されなくなったことがわかります。

これは Update ボタンで呼び出される Button-Proc ファンクションが、通常の = 割り当てではなく、:= となり、依存関係の式が削除されたことによります。

※実際の作業では、追跡やデバッグが難しくなるため、依存関係の使用は避けるべきです。



10. ここまでの作業を別名で保存します。

メニュー **Files** → **Save Experiment As** を選択して、
"Tour1E.pxp" として保存します。

以上で、Igor Pro の一般的なツアーは終了です。

一休みするときは、ここで Igor Pro を終了しても構いません。