

LECTURES

Igor Pro™を用いた生理学データ解析 第2回：データの表示と解析

東京慈恵会医科大学・総合医科学研究センター
神経生理学研究室

加藤 総夫

要 旨

最近、特に高impactの学術誌に掲載されている生理学論文の実験方法の中に「解析・作図にはIgor Proを用いた」という記述を頻繁に見かける。Igor Proは、一次的生データ観察から、統計解析・時系列解析・画像解析、そして、論文投稿レベルの図の作成に至るまでを可能にする科学技術演算用汎用ソフトウェアである。データの扱いやファイルの概念が一般的なオフィス用ソフトウェアと大きく異なるため、理解しにくく使いにくい、という声を聞くことも多いが、基礎的な概念を正しく理解し、データの扱いの基本を習得すれば、その奥に広がる可能性は無限大に近い。未知の現象を解明することを使命とする生理学者にとっても大きな力となるソフトウェアである。その入り口の「敷居をまたいで」もらうことを目的として、基礎概念を解説した前回に続き、今回はデータ解析の実例を紹介する。

キーワード：データ解析，時系列データ，画像データ，プログラム

III. Igorを用いたシナプス後電流の解析

今回は、Igorでデータ解析を行う上で必要となる wave, experiment および data folder の概念, data browser の使用法, graph, table および layout という Window 表示形式, そして, procedure, macro および function というプログラム宣言文について概説した。今回は、これらの理解を前提として実際のデータ解析の例を紹介する。

A. 波形の取り込み

まず、生体信号を Igor のデータに変換しなければならない。ソフトウェアから制御された AD コンバーター・インターフェースを用いるのが一般的だと考えられるので、電気生理学で用いられる主なデータ形式からどのように Igor にデータを wave データとして渡すか、簡単に説明する。例として、等間隔 (たとえば 0.25 ms) ごとに標準化された膜電流 (単位 pA) データを解析対象としてみよう。Igor の中でこのデータは、「1D wave」として扱われる。三つの方法を紹介するが生理学者にとって最も簡便なのは第三の方法である。

1. テキスト・ファイルを介して数値を渡す

あまりエレガントな方法ではない。Igor の重要な特色の一つは大量のデータ点からなる大規模なデータを高速で処理できる点にあり (我々は、4000-10000 Hz で数時間にわたって取り込まれた 1000 万点以上

のデータを一気に解析することもある), その目的のためには非圧縮のテキスト・ファイルはサイズが大きくなってしまい取り扱いが不便. カンマやタブなどで区切られたデータ, スペースで区切られたデータ, 桁位置の揃えられたデータなどのテキスト・ファイルをメニューバー>Data>Load Wavesからそのまま読み込むことができる. しかし, この方式で読み込んでwaveを作成した場合, 単なる数値の並びに過ぎず, waveに必要な横軸情報や縦軸情報は付加されていない(前回 wave の項参照). そこで, メニューバー>Data>Change Wave Scalingで, 横軸情報, 縦軸情報を手入力で付加する必要がある.

2. Microsoft Excel を介してデータを渡す.

Igor は Excel のデータをそのまま読み込むことができる. しかしこれも, 前項と同じ理由で, ポイント数が非常に多い場合にはあまり効率のいい方法ではない. Excel というどこにでもあるソフトを介してデータを受け渡しできるという以外何も利点はない.

3. 変換ソフトを介してバイナリー・データとして読み込む.

a. NeuroMatic (フリーウェア).

<http://www.neuromatic.thinkrandom.com/>

NeuroMatic は Igor の function 集. Pclamp ABF ファイル, Pulse ファイル, Axograph ファイルを直接 Igor から読み込むことができる. さらに, R.Angus Silver 博士らのさまざまなシナプス電流数値解析プログラムも含まれている.

b. DataAccess (市販).

<http://www.bruyton.com/DataAccess/>

P Clamp, Axoscope, Spike2 (CED), Pulse などのファイルを直接読み込んで Igor ファイルに変換して出力できる.

c. 取り込みソフト側で Igor ファイルに変換.

PowerLab (ADInstruments) に付属の取り込みソフト Chart は, データをそのまま Igor の experiment ファイルとして保存できる.

d. Igor から AD 変換ボードを操作する.

National Instruments の NIDAQ 取り込みボード, もしくは, InstruTech の ITC interface は Igor から直接操作してデータを直接 Igor に読み込ませることができる.

B. 波形ビューワーとしての Igor

どのような生体信号解析を行うにしても, 生データの詳細な観察こそがもっとも重要であることはいうまでもない. Igor の優れたユーザー・インターフェースの特徴のひとつは大規模データの波形の閲覧がきわめて容易かつ高速だということである.

まず wave をグラフとして表示する. メニューバー>Windows>New Graph で閲覧する wave を選ぶ. 規定の縦軸の名前は left, 横軸は bottom. 他にも right, top という選択もあり, また, 自分で固有の軸の名称をつければ, 一つの graph の中に複数の wave を異なる軸で表示することが可能になる. 同じ時間軸 bottom に対して, 生波形, 頻度, 振幅などのプロットを別の段に表示したいときに必要なテクニックである.

ここでは, 標準的な 1D wave を表示したいので, left には, wave の名前 (今仮に Im とする) を, bottom には _calculated_ を選ぶ. 「_calculated_」は, 1D wave を定義したときに規定した横軸情報で表示させるときに用いる (図1). OK をクリックすると新たな Graph window が作成され波形がプロットされる (図2).

この操作を行った後, command 領域を見てみると, history 領域 (command 領域のすぐ上) に「Display Im」と表示されている (図3). これが Igor の強力な「history 記録機能」で, 自分が行った操作が,

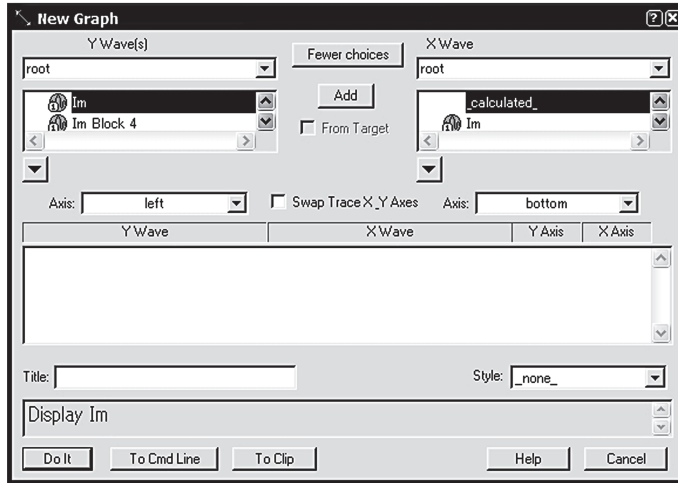


図 1. 新規 Graph 作成時の対話型メニュー. 実行されるコマンド (Display Im) が最下部に表示されている.

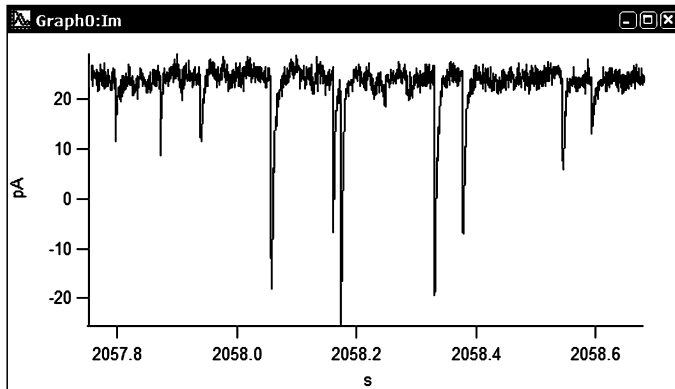


図 2. 1D waveをプロットしたGraphの例. Waveに定義されている縦軸・横軸情報に沿って表示される.

(特に指定しない限り) すべて history 領域に書き込まれる. 初めて自作の procedure を作成するときは, この history を前回述べた Procedure window 中の Macro 宣言文と End で挟まれた部分にコピー・ペーストして自分のやったことをそのまま再現させるところから始めるのがわかりやすい (詳しくは後述).

試しに, ここで, Im が表示されている graph window を 1 回閉じてみよう (普通の Windows や Mac の操作どおり). 「Graph 作成用のマクロを作成するか?」と聞いてくるがここでは No save を選択して閉じる (Save を選ぶとどうなるかは後述). 当然, さっき作った graph window は消えてしまう. そこで, history 領域からさきほど生成された 「Display Im」 の行をコピーして command 領域にペーストしてみよう (history で指定したい行をクリックすると行全体が選ばれ, そこでリターンを押すと自動的に command 領域にコピー&ペーストされるという便利な機能があるのでそれを利用するとよい; 図 3). Command 領域でリターンを押してコマンドを実行すると, さきほど, 対話型メニューで作成したのとまった



図3. History領域（上の広いスペース）とcommand領域（最下段の空白行）. この二つは一つのexperimentファイルに必ず1セット組になって表示され、windowのタイトル部にはexperimentファイル名が表示される. この二つを持たないexperimentファイルはない.

く同じgraphが再現される. MS WordやExcelの「記録・再生型マクロ」とはだいぶ勝手が違うが、このhistory画面に、行った操作がほとんどすべて、コマンド文として記録されている、というところがIgorのデータ操作の重要なポイントである（試しに、history領域にカーソルを移動させて何か文字を消去したり書き込んだりしてみると、Igorが昔、まだ、マニアックな一部のMacファンだけに使われていた頃を彷彿とさせるイラストが表示される；ただしこのイラストはカスタマイズ可能）.

換言すると、Igorの「対話型メニュー」とは、実は、「コマンド文生成ユーティリティ」に他ならない. 事実、対話型メニューの下部には必ず「Do It」以外に「To Cmd Line」（生成されたコマンド文をcommand領域にペーストする）、および、「To Clip」（コマンド文をクリップボードにコピーする）というボタンが表示される. これらを用いれば、対話型メニューで生成されたコマンドをそれぞれcommand領域やプログラムの中に貼りこんで、適宜書き換えた上で利用することができる. この意味においてIgorは、本質的に「command領域から1行ずつコマンド文を入力して実行させる」というコマンド・インタプリタ型のソフトに、強力なマクロ・コンパイラ機能を加え、さらに、メニューバーから操作できる対話式コマンド文生成ユーティリティを備えた「プログラミング言語」であるといってよい. これが、メニューからの選択によってすべての操作を行ういわゆるオフィス用ビジネス・ソフトの使い勝手と一線を画する点である. プルダウンメニューの選択だけでもさまざまなデータ処理が可能だが、厚さ9cmのIgorマニュアルには対話型メニュー（あるいはメニューバー）からは生成できない（つまり直接プログラム中に打ち込むしかない）多くの解析用コマンドが記載されている. Igor上級ユーザーになるためにはこれらの理解が必須である.

その一方、コマンド・ラインから実行することのできない、主にカーソルとクリック操作のみによって実行可能な便利な機能もIgorには多く存在する. その例が、波形表示に関する処理である. その事例を紹介する. たとえばここで、波形の一部分だけを拡大してみよう. Igorにはmarqueeという強力かつ快適な表示サポート・ツールがある. この便利さを覚えてしまうと他のソフトの拡大・縮小操作がまどろっこしく感じられる. 波形の拡大したい部分にマウス・カーソルを置いて斜めにドラッグすると点線で囲まれた長方形が表示される. マウスボタンを離しても長方形は表示され続ける. その長方形で囲まれた領域の中にマウス・カーソルを移動させるとポインターが音楽の全休符の記号に変わる. そこでマウスをポイントすると、この長方形で囲まれた領域に対して何をしたいか、という選択メニューが表示される（図4）. これで任意の範囲の拡大・縮小が瞬時にして行われる.

このMarqueeメニューに自分で作成したコマンドを指定して、波形の処理に用いることも可能である（上級テクニック；図5）.

また、波形上にマウス・カーソルを置いて1秒以上押し続けると、マウス・カーソルが消えて波形ご

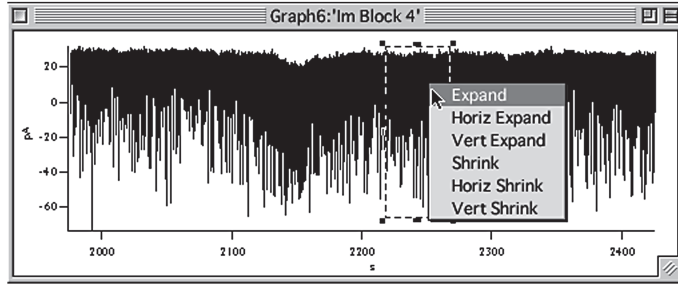


図4. Marqueeメニュー．破線の長方形で囲まれた部分に対して横軸・縦軸および両方の拡大と縮小を選択できる。

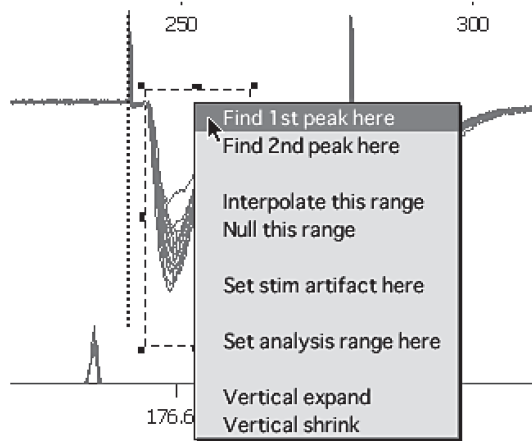


図5. カスタム marqueeメニュー．長方形で選んだ範囲からピークを探させたり，その左端を刺激時点であると認識させたり，範囲内の値を消去したり（刺激 artifactの部分だけ波形から消す処理など）といった操作をこのメニューから実行させることが可能で，カーソルで範囲をいちいち指定するよりも簡単。

と自由にドラッグできるモードになる。Graph windowの中での波形の表示位置を自由に変えることができる。この操作は実は、メニューバー>Graph>Modify Trace Appearanceで表示されるグラフ設定パラメーターの中の「offset」の値を変化させて縦軸・横軸に対する波形の相対的位置を変えるという操作に等しい。この対話型メニューで表示されるoffsetの前のチェックボックスをクリックして値を直接手入力する代わりに、表示を見ながらマウスでoffsetを変更しているのである。

また、この機能を利用して、同じwaveの違う部分を重ね描きすることもできる。メニューバー>Graph>Append Traces to Graphを選び、今、表示しているGraph window内にさらに同じwaveを複数個表示させる。個々のX軸のoffsetを変えれば、同じwaveの違う時点、たとえば刺激時点を揃えたoverlay表示が可能となり、また、さらにY軸のoffsetをずらせば、連続した時間軸延長波形を表示できる(図6)。

Alt (Windows) もしくはopt (Mac) を押しながら、グラフ上でマウスをクリックするとマウス・カーソルが手の形に変わる。この状態でドラッグすると波形の縦軸・横軸の表示範囲をシフトできる。こ

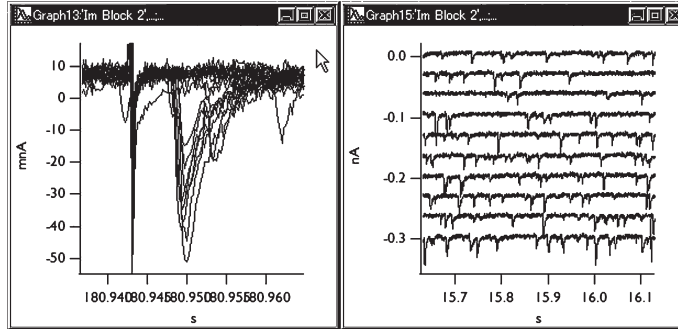


図6. 同じ wave を X offset をずらして重ねてプロットしたもの。左, 20 回の重ねあわせ, 右, 10 回の重ねあわせ。右のプロットは Y offset もずらして表示させた。

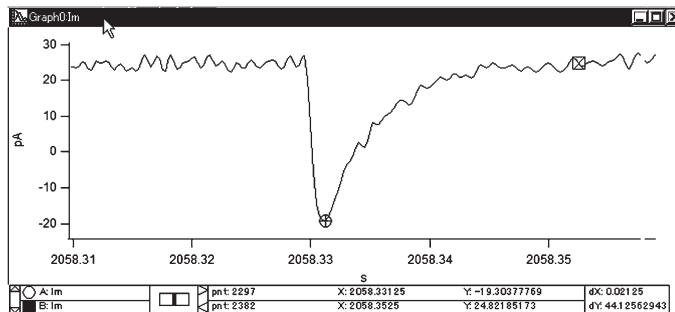


図7. Marquee メニューで横軸を拡大したプロットに対し, Show Info コマンドを用いて, Graph の下部にカーソル A と B の情報を表示させた。

の場合, 表示される範囲が変わるのであって, offset は変わらない。EPSC のイベントを探したりするのに便利。

メニューバー > Graph > Show Info で graph の下部にカーソル表示窓が表示される。2 種類のカーソル A と B をプロット上にドラッグすると, A, B それぞれについて, その wave の X 位置, ポイント数, Y 軸の数値, および, A と B の X および Y 値の差が表示される (図7)。

C. 波形測定・解析ソフトとしての Igor

Igor は単なる生理学実験データ専用の解析ソフトでは (もちろん) なく, 宇宙工学から素粒子物理まで幅広く用いられている汎用解析ソフトである。NASA の火星探査機の軌道解析にも用いられているし, Igor メーリングリストのサーヴァーは最近までアメリカのペンタゴンに置かれていた。Igor の持つ数値解析能力の強力さをここで説明し尽くすことは不可能である。一例として EPSC 波形の curve fitting を紹介する。

さきほど表示した EPSC の波形に対して single exponential を適合させて減衰時定数を推定してみよう。メニューバー > Analysis > Curve Fitting を選び, 対話型メニューを表示させる (図8)。Function and Data のタブから Function : exp_Xoffset を選ぶ (注: この選択肢は Igor version 5 以降にしかでてこない。Igor 4 以前で同じ処理をするには少し工夫が必要)。Function のリストにきわめて多くの種類の関数が表

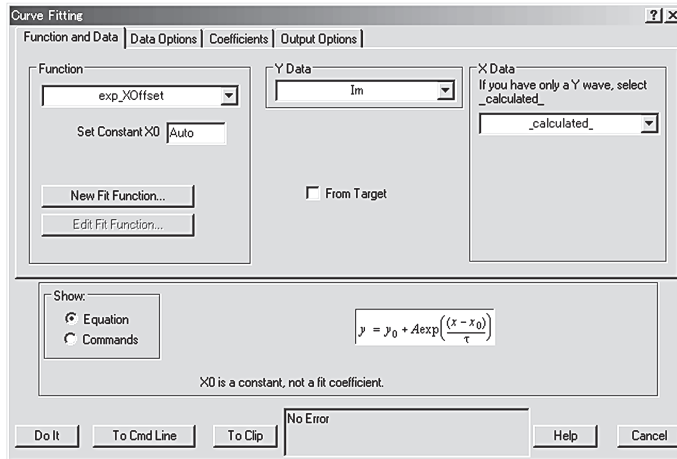


図8. Curve-fitting 対話型メニュー.

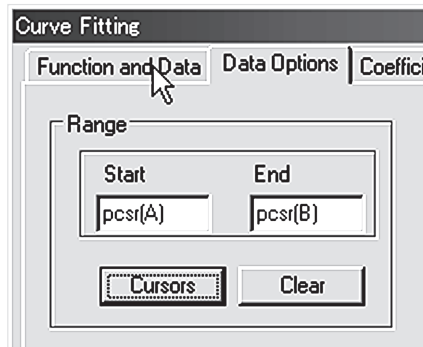


図9. Fittingの範囲を cursor A と B の間に設定.

示されていることにご注目いただきたい。しかも、ユーザーが定義した関数をこのリストに表示させて fit させることも容易である。Y Data : Im を選ぶ。次に Data Options のタブから、Range の項の Cursor ボタンをクリックする (図9)。これで、cursor A と B で囲まれた範囲が fitting の対象になる。

Do It をクリックすると (あるいはリターンを押すとーリターンは Do It のクリックと同じ効果を持つ)、指数関数の fitting が始まり、うまく収束すれば結果が history 画面に表示され、graph window には、最適化された指数関数が元の波形に重ねて表示される (図10)。

D. 自動化された波形測定プログラムを Igor 上で走らせる

この「手操作」で行った curve fitting を自動化する簡単なプログラムを書いてみよう。メニューバー > Windows > Procedure Window で Procedure window を開く。そこに、下記のように記入する。

```
Macro TestExponentialDecay( )
End
```

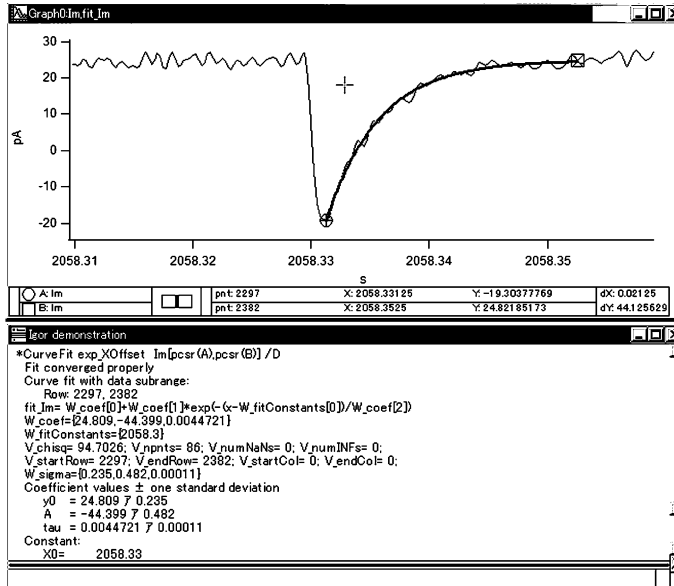


図10. Curve-fittingの結果. 適合波形（上）と適合のパラメーター（下）.

Curvefitを行わせた命令の行を history window からコピーし，Macro 文と End 文で挟まれた中にペーストする．

```

Macro TestExponentialDecay()
    CurveFit exp_XOffset Im[pcsr(A),pcsr(B)] /D
End

```

これで最も簡単なプログラムができた．Procedure window を（通常のやり方で）閉じる．これは，Cursor A と B で囲まれた範囲を自動的に指数関数で適合するプログラムである．適当な EPSC を画面に大きく表示させて先ほどと同じように二つのカーソルを適合したい範囲に置く．(1) 「TestExponentialDecay ()」 と command 領域にタイプしてリターンを押す．または，(2) メニューバー > Macro > TestExponentialDecay を選ぶ，のいずれかによってプログラムが実行される．(2) の方法が便利．

このマクロをもとに，もう少し応用可能性を高めたのが次のマクロである．前回述べた Igor マニュアルの勧めに従い，Macro 文ではなく Function 文を用いる．

```

Function TestExponentialDecay()
    WAVE W_coef
    Variable FitRange = 0.02 //fix the range to fit to 20 ms (0.02 s)
    String waveCursorAIsOn = CsrWave(A)
    CurveFit/Q exp_XOffset $waveCursorAIsOn (xcscr(A),xcscr(A) +FitRange)/D
    print "amplitude (pA) = ", W_coef[1], " tau (ms) = ", W_coef[2]*1000
End

```

改正点は、以下の4点。(1) Cursor Aが置かれた時点 (EPSCのピーク) から20 ms後までの波形をfittingの対象にする (Cursor Bの指定は不要)。この20 msという値は3行目のFitRangeというユーザー変数で定義されている。(2) fittingの時の細かい結果表示をさせない (CurveFit文の/Q)。(3) Cursor Aが乗っているwaveのwave名を自動的に読み取り、そのwaveに対してcurve fitを行う (4行目)。(4) 振幅と時定数だけをhistory欄に表示する。//は注釈を表す記号で行のどこにおいてもそれ以降は無視され実行されない。Function文を用いたので、このままだとメニューバー>Macrosには表示されないが、上記のFunctionの上でも下でもどこでもよいかから以下の3行をProcedure window中に書き加えれば、新たに作られるメニューバーの項目「Try Igor Pro」からプルダウンメニューで実行できる。

```
Menu "Try Igor Pro"  
    "Fit Exponential Decay on EPSC".TestExponentialDecay()  
End
```

ところで、先ほどGraphをDisplayするために、対話型メニューを使う方法と、command領域にコマンドを入力する方法を説明した。実は、描かれたGraphをマクロとして記録し、それを実行することによって描記させることも可能である。Igorではすべての要素がプログラムとして表現可能なのである。

上記で作られた原波形とfittingの適合波形が表示されたGraph windowを1回閉じてみる。このとき、「recreation macroを保存するか?」と訊かれるので、ここでは「Save」をクリックする。そして、Procedure windowを開いてみると、そこに、Window Graph0で始まり、やはりEndで終わる新たなプログラムができています。これは、先ほどのグラフ表示とまったく同じグラフを(再)作成するためのプログラムである。

試しに、このGraphプログラムを実行してみよう。二つ方法がある。その1. Procedure Windowを閉じ、command領域にGraph0()と入力してリターンを押してみると、先ほど閉じたものと全く同じグラフ表示が再現される。その2. もう1回Graph windowを閉じて、メニューバー>Windows>Graph Macros>Graph0で選択しても同じグラフが再び作られる。

また、先ほどはGraph windowを閉じることによってrecreation macroを保存させたが、メニューバー>Windows>Control>Window Control>Create Window Macroで「recreation macro」を作成することもできる。同様に、table window, layout windowに対してもこのようなrecreation macroを作成することができる。これらのWindowマクロを実行するには、メニューバー>Windows>X macrosで選択する(X = Table, Layout, etc...)。Procedure windowの中で、このrecreation macroを修正すれば表示の体裁などを変えることもできる(これらのプログラムを読解してGraphやLayout表示のプログラミング技法を研究することができる)。

E. 波形解析プログラムの作成

上の単純なプログラムをさらに発展させれば、もっと複雑なプログラミングも可能である。下の例は、他のサブルーチンから呼び出されて自動処理し、振幅と時定数のあるwaveに格納するというFunctionである。解析対象となるwave名はImで、すでにEPSCのピークの時点がPeak_addressというwaveに数値データとして収められているとする。このfunctionは、他のサブルーチンから呼び出され、自動的に指数関数のcurve fittingを行い、振幅と時定数を、それぞれ、Peak_amplitudeおよびPeak_decayというすでに定義されているwaveに格納する、というプログラムである。Fittingが収束しなかった場合のエラー処理などは何もしていない最も単純なプログラムである。Function独特の変数受け渡し宣言と引数の呼び出し方などを参考にしたい。

```

Function ExponentialDecay( )
  WAVE Im, Peak_address, Peak_amplitude, Peak_decay, W_coef
  SVAR NameofTargetWave
  NVAR NumofPeaks
  Variable CurrentPeakNum = 0
  Variable FitRange = 0.02 //fix the range to fit to 20 ms (0.02 s)
Do
  CurveFit/Q exp_XOffset $NameofTargetWave (Peak_address[CurrentPeakNum],
  Peak_address[CurrentPeakNum]+FitRange)/D
  Peak_amplitude[CurrentPeakNum]=W_coef[1]
  Peak_decay[CurrentPeakNum]=W_coef[2]*1000
  CurrentPeakNum+=1
While (CurrentPeakNum<NumofPeaks-1)
End

```

(注：8行目と9行目は，紙面の都合で改行してあるが，Igorのプログラムの上では同じ1行に続けて入力する)

さらにパネルを用いた入出力など，より高度な処理をFunction文を用いて付け加えていくことによって，特定の解析に特化したプログラムを書くことも可能である．Igorにはパネルに表示させるためのさまざまなオブジェクト指向型ボタンやスイッチが数多く備わっていて，高度なユーザー・インターフェースを用いたプログラミングも容易である．図11は，筆者の研究室で刺激誘発シナプス後電流の解析用

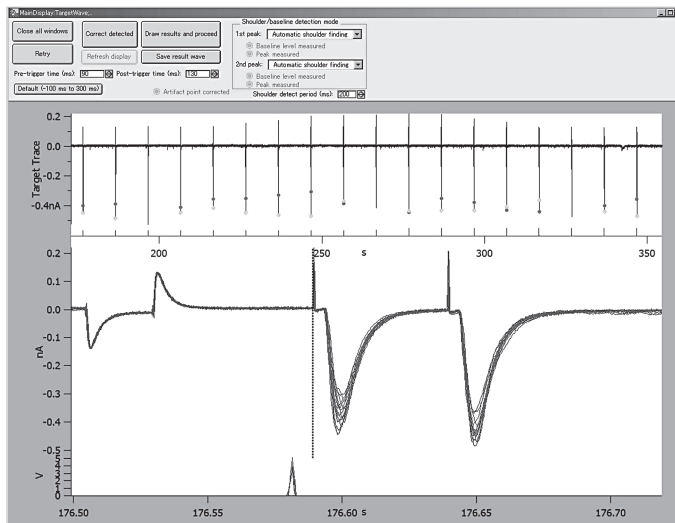
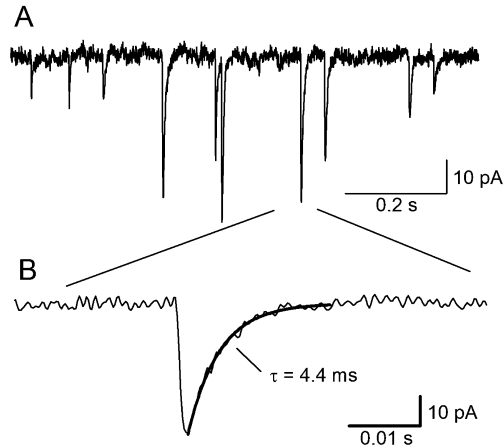


図11. カスタム・メードIgorプログラム「Evoked Analysis Pro (© Fusao Kato, 2004)」の解析条件設定画面。



Kato et al., Fig. 1

図12. 論文投稿用（を意識して作成した）マルチパネルの図。

に用いている「Evoked Analysis Pro (© Fusao Kato, 2004)」の解析条件設定画面である。

IV. さらに進んだIgorの用法

A. 画像解析ソフトとしての用法

われわれはIgorで画像解析も行っている。付属しているImage Processingという解析マクロ集に非常に便利な機能が多くあり、さまざまな解析に応用している。ただ画像処理をするだけではなく、形状の認識や数値化、フーリエ変換を含む各種フィルター、Ca²⁺ imagingの表示に有用なpsuedocolor表示など定量化に便利な機能が豊富にある。詳細は別稿に譲る。

B. 投稿用図の作成とPowerPoint用の図の作成

生理学者の仕事の中で最も重要なものの一つが、論文を投稿することであることは言うまでもないが、Igorは、解析のみならず、それだけで論文投稿用の図まで作成可能である。いくつかのGraphをLayoutに貼りこんで、ひとつのマルチパネルの図を作成する。さきほど表示させたGraphを二つ、貼りこんで一つの図にしてみよう（図12）。

このLayoutの質は、完全に“publishing quality”である。我々はこのLayoutをそれぞれの学術誌の指定に合わせた形式で保存してそのまま投稿している（on line投稿の場合）。ほとんどすべてのファイル形式、解像度で保存可能である（version5以降）。また、高画質のjpeg形式で保存したファイルをPowerPointに貼りこんでプレゼンテーション用のスライドを作成している（しかし、多くの場合、そのままのjpeg形式だとPowerPointのファイルが「重く」なるので、1回、PhotoShopを介してweb用に圧縮保存したjpegファイルをPowerPointに貼り付けることが多い）。PowerPointからは、「挿入>図>ファイルから」で読み込み、「トリミング」機能を活用して見やすいように配備しなおす（∵論文投稿用の図は縦長のsingle columnの図が多く、PowerPointのスライド発表用画面は横長だから）。

C. 電気生理学的データの動画化

Ca²⁺ imagingを用いた研究成果を口演発表するとき、動画を呈示することは常套手段となっている。ダイナミックなシナプス入力の変化や、膜電位の変動もリアルタイム（に近い）動的な変化として見ることによって電気生理のデータに慣れていない人はもちろん、見慣れている人に対しても説得力のあ

るプレゼンテーションが可能である。講義用スライドにも有効。Igor の Movie 作成コマンドを用いれば電気生理データの動画を簡単に作成することができる。

原理は、(1) Graph で波形の wave を表示させる。(2) 表示させる区間を変数で表すようにプログラムを組む。(3) 表示部分を NewMovie/O と CloseMovie コマンドで囲み、間に、変数を少しずつ変化させて表示区間を少しずつずらしていく部分 (Do ~ While ループなど) を書き、表示区間を 1 回ずらすごとに AddMovieFrame というコマンドでムービーファイルに一コマずつ加えていく、ということに尽きている。少しずつ動かすパラパラマンガの原理である。

下記はその例。すでに wave が表示されており、変数なども定義されているとする。1 フレームごとに 0.05 秒ずつ波形が流れていくムービーができる。

```
NewMovie/O
do
    starttime+=0.05
    SetAxis bottom starttime,starttime+0.5
    DoUpdate
    AddMovieFrame
    framenumber+=1
while (framenumber<200)
CloseMovie
```

同様に、ヒストグラムなども時々刻々変化させていくことが可能である。筆者の研究室のホームページで例を見られたい (<http://www.jikei.ac.jp/lnp/index.htm> > 研究活動 > 孤束核シナプス前プリン受容体)。

このコマンドは、QuickTime Movie を作成するコマンドなので、PowerPoint に貼り付ける場合は、AVI 形式に変換しておく必要がある。QuickTime 形式から AVI 形式への変換は本稿の範囲を逸脱するので省略する。

おわりに

Igor の解析能力のすべてをここで紹介することはできない。Version 5 のマニュアルは全部で 9cm の厚みがあり、そこには、波形処理、統計処理、時系列解析、画像解析を含む生理学データの解析に有用な機能がたくさん掲載されている。Igor そのものの機能の高さに加え、開発している WaveMetrics 社のスタッフの技術力と熱意、世界中のあらゆる自然科学分野のユーザーの貢献と意欲 (Igor メーリングリストのレベルとアクティビティの高さは筆舌に尽くしがたい) が、Igor を無限の応用可能性を持つソフトウェアに育ててきた。筆者は WaveMetrics 社から宣伝を頼まれているわけでもないし、みんなに Igor を使ってもらいたいと思っているわけでもないが (もちろんユーザーが増えれば突然市場から消えてなくなる心配が減るという効果はある)、自然科学者が、パソコンで大規模データを処理するために「電卓代わり」に用いる最適のツールとして、現時点では Igor がベストと考え、11 年来のユーザーとして紹介した。その動機は、筆者の研究グループが「Igor の奥深い機能を駆使して誰もやっていないような解析を行い、既成の解析手法では見つけにくいような現象を見つけ、定量的に評価して発表している」ことをちょっぴり自慢したい、ということかもしれない。実験データが目の前にあるのに、誰かが解析ソフトを作ってくれるのをのんびり待っているわけにはいかない。