

CONTENTS

サンプルの Experiment – Differential Equation Demo	2
はじめに.....	2
例 :	2
例を操作してみる :	3
そのほかに試してみること.....	4
フィッティングの答え	4
プロシージャの内容.....	4

サンプルの Experiment – Differential Equation Demo

メニュー File → Example Experiments → Analysis → Differential Equation Demo

この Experiment は、微分方程式の数値解に対するカーブフィッティのデモです。

このデモは、Jerry Eykholt 氏（2000年9月21日当時：ウィスコンシン大学マディソン校）が提供してくださったエクスペリメントファイルに基づいています。

はじめに

このエクスペリメントは、常微分方程式の数値解にデータをフィッティングするという簡単なデモンストレーションです。

特に、Igor Pro の2つの機能を示しています。

- IntegrateODE
ユーザーが指定した常微分方程式を積分するために使うコマンドです。
ODE の解法はマニュアル III-7 Analysis で説明されています。
- 「一括で」フィッティングする
ユーザー定義の関数フィッティングのオプションです。
通常、フィッティング関数は1度に1つのポイントだけを返します。
ODE の数値解は、最も便利にすべてのポイントを同時に返します。
そのためには、一度にすべてのポイントにフィットする関数が必要です。
一括でフィッティングする関数は、マニュアルの III-8 Curve Fitting のセクション（User-Defined Fitting Functions: All-At-Once Fitting Functions）で説明されています。

常微分方程式（ODE）に基づくモデルに実験データをフィッティングするには、いくつかの方法があります。

積分された方程式の一般的な形式は、線形化して線形回帰を適用できる場合があります。

あるいは、データを補間し、その補間関数を微分して非線形速度式に当てはめることもできます。

しかし、積分最小二乗法という優れた方法があります。

これは、積分された常微分方程式を実験データと直接比較する方法です。

このアプローチでは、データ過重への意図せぬ影響や、差異のある実験的反応の解釈へのノイズの影響を回避することができます。

例：

化学反応速度の問題を考えてみます。

よく攪拌されたバッチ反応器では、化学物質は過剰にある他の試薬によって分解されます（そのため、反応速度は他の試薬の影響を受けません）。

反応速度は次のように表すことができます：

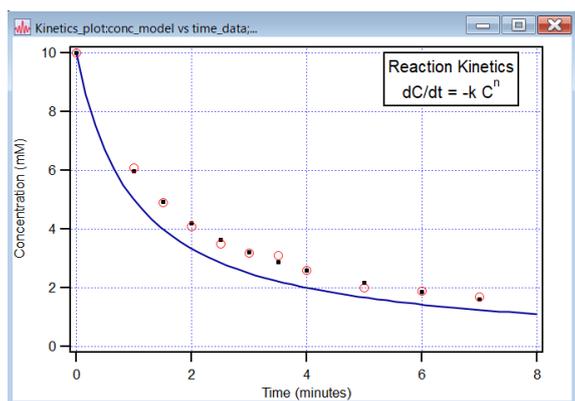
$$dC/dt = -k C^n$$

ここで、C は化学濃度、t は時間、k は反応速度係数、n は反応次数です。

C、t、k の単位は一貫していて、n は無次元です。

サンプル実験内の例では、10 mM 溶液を調製し、その後、数分間でバッチ反応器で反応させます。

実験データは「Kinetics_plot」ウィンドウにプロットされ、「Data_Table」ウィンドウに表形式で表示されます。表を表示するには、Macros メニューから Edit data or weighting を選択します。

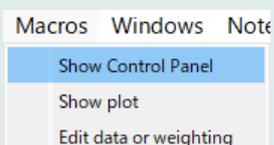


Point	time_data	conc_data	conc_model	weight_factor
0	0	10	10.0098	2.00767
1	1	6.1	5.9774	1.59227
2	1.5	4.9	4.94364	1.48978
3	2	4.1	4.20245	1.41705
4	2.5	3.5	3.64639	1.36281
5	3	3.2	3.21464	1.32085
6	3.5	3.1	2.87021	1.28743
7	4	2.6	2.58941	1.2602
8	5	2	2.15992	1.21854
9	6	1.9	1.84765	1.1882
10	7	1.7	1.61091	1.16512
11				

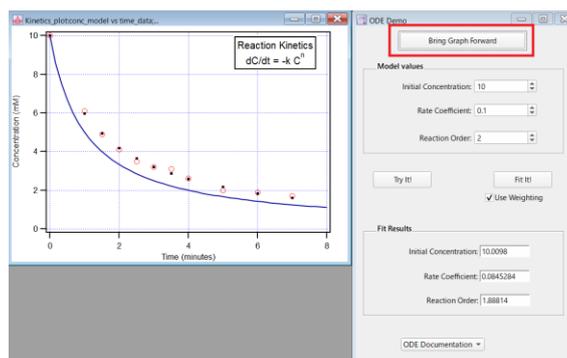
例を操作してみる：

以下の手順に従って、例を実行してみてください。

1. Macros メニューから Show Control Panel を選択します。



2. Bring Graph Forward ボタンをクリックし、パネルが隠れた場合は、パネルをクリックして位置を調整します。

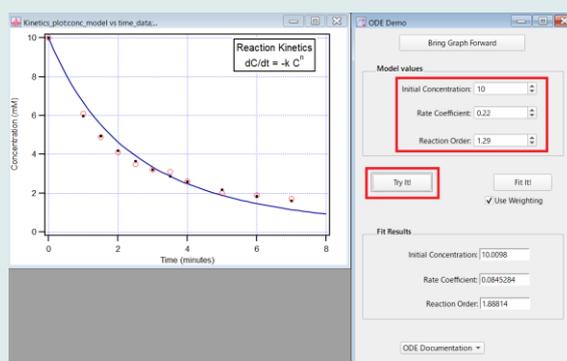


3. 上部の Model Values と表示された領域にあるボックスに値を入力します。

Try It ボタンをクリックして、入力した値で IntegrateODE を実行します。

かなり良いものができるまで、値をいろいろ試してみましょう。

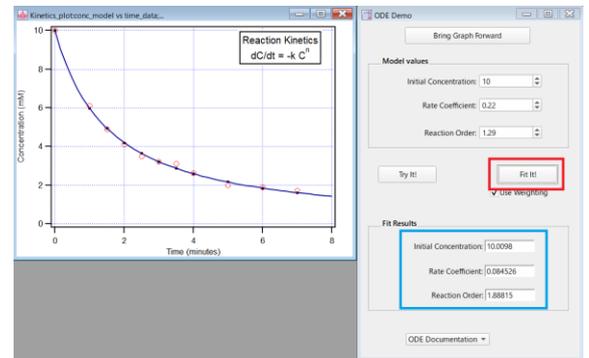
反応次数の大きな値や負の値など、特定の値の組み合わせにより、積分が何百万もの微小ステップを踏むことになります。



4. パラメーターが概ねフィットしていることを確認したら、「Fit It」ボタンをクリックします。

IntegrateODE コマンドで何らかの問題が検出された場合、エラーメッセージが表示されることがあります。

フィッティングが終了すると、パネルの下部にフィット値が表示されます。



そのほかに試してみるごと

コントロールパネルの下部にあるメニューから、さまざまなトピックに関するオンラインヘルプにアクセスできます。

実験データポイントに異なる重みづけを適用できます。

Macros メニューの Edit data or weighting を選択して、重み付けウェーブを調整します。

通常の Curve Fitting ダイアログで提供されているその他の機能の多くも使うことができます。

この場合のユーザー定義関数は「Integral_Fit」です。

試してみるべきことの1つは、フィッティングされたパラメーターの値を1つ以上 Hold して試みることです。

コマンドを学ぶために、Procedure Window を編集または確認します。

Igor Pro プログラミングに中程度の経験を持っているれば、これは簡単な作業でしょう。

最後に簡単なコメント付きで記載しています。

フィッティングの答え

提供されたデータ（過重なし）に最もフィットするパラメーターの結果は次のようになります。

Fit parameters \pm one standard deviation

Initial Concentration, $C_{init} =$ $K0 = 10.01 \pm 0.50$

Rate constant, $k =$ $K1 = 0.085 \pm 0.066$

Reaction order, $n =$ $K2 = 1.89 \pm 0.45$

プロシージャの内容

Procedure Windows を開くと次のプログラムが表示されます。

わかりやすくするために、コメントを日本語にしておきます。

```

#pragma rtGlobals=1                      // Use modern global access method.

Menu "Macros"                            // Macros メニューにアイテムを追加
    "Show Control Panel", DoWindow/F ODE_Demo_Panel
    "Show plot", Show_plot()
    "Edit data or weighting", Edit_Data_Table()
End

// 各メニューアイテム用のマクロ
Macro Edit_Data_Table()
    DoWindow/F Data_Table
End

Macro Show_plot()
    DoWindow/F Kinetics_plot
End

// Try It ボタンを押したときの処理プロシージャ
Function Enter_initial_guesses(ctrlName) : ButtonControl
    String ctrlName

    NVAR InitialC
    NVAR RateCoefficient
    NVAR ReactionOrder

    Wave pw
    Wave fit_conc_data

    // コントロールパネルの値をパラメーターウェーブに読み込む
    pw[0] = InitialC
    pw[1] = RateCoefficient
    pw[2] = ReactionOrder
    // 初期条件を宛先ウェーブに読み込む
    fit_conc_data[0] = pw[0]
    // 微分方程式を積分
    // 初期推定値に積分結果を渡す
    IntegrateODE Homo_rxn, PW, fit_conc_data
End

// Fit It ボタンを押したときの処理プロシージャ
Function Integrate_and_Fit(ctrlName) : ButtonControl
    String ctrlName

    NVAR InitialC
    NVAR RateCoefficient
    NVAR ReactionOrder

```

```

// コントロールパネルの値を係数ウェーブに読み込む
Wave PW
pw[0] = InitialC
pw[1] = RateCoefficient
pw[2] = ReactionOrder

// フィッティングを合理的な範囲内に収めるための制約条件。いくつかの特異な行列エラーを防止する
Make/O/T ConstraintWave = {"K0 > 0.00001", "K1 > 0.00001", "K2 < 5"}
ControlInfo AddWeightingCheckBox
// フィッティングの実行
if (V_value)
    FuncFit Integral_Fit PW conc_data /X=time_data /W=weight_factor
/D=conc_model /C=ConstraintWave
else
    FuncFit Integral_Fit PW conc_data /X=time_data /D=conc_model
/C=ConstraintWave
endif

// フィッティング曲線
Wave fit_conc_data
fit_conc_data[0] = pw[0]
IntegrateODE Homo_rxn, PW, fit_conc_data

NVAR fit_InitialC
NVAR fit_RateCoefficient
NVAR fit_ReactionOrder
// フィッティング結果をコントロールパネル下部の表示用に読み込む
fit_InitialC = pw[0]
fit_RateCoefficient = pw[1]
fit_ReactionOrder = pw[2]
End

// IntegrateODE を使うのに便利な形で微分方程式を実装する関数
Function Homo_rxn(pw, tt, Cw, dCdt)
Wave pw // pw[0] = 初期濃度, pw[1] = 反応定数, pw[2] = 反応次数
Variable tt // この例では使われない: 時間微分は現在の濃度のみに依存する
// 例えば、強制関数の値を計算するときに必要な可能性がある
Wave Cw // 入力: 現在の濃度。1つの要素のみ持つ。1つだけ方程式がある
Wave dCdt // 出力: 濃度の導関数。1つの要素のみ持つ。1つだけ方程式がある

// 方程式は1つだけであるため、式は1つだけ
// 微分方程式の一般的な形式は:  $dC/dt = -k \cdot C^n$ 
dCdt[0] = -pw[1]*Cw[0]^pw[2]
End

// カーブフィッティング関数。一括で機能するため、すべてのパラメータはウェーブ
Function Integral_Fit(pw, yw, xw) : FitFunc

```

```

Wave pw          // ノーマルフィッティング係数ウェーブ
Wave yw          // 出力- モデル結果
Wave xw          // 入力- データ x 値      (yw の x スケーリングからも取得可能)

yw[0] = pw[0]    // 積分の初期条件を設定
IntegrateODE /X=xw Homo_rxn, pw, yw // 「一括」フィッティング用に選択した積分関数
End

// Bring Graph Forward ボタンを押したときの処理プロシージャ
Function GraphForwardButtonProc(ctrlName) : ButtonControl
    String ctrlName

    DoWindow/F Kinetics_plot
End

// テーブルウィンドウの表示
Window Data_Table() : Table
    PauseUpdate; Silent 1 // ウィンドウの作成中
    Edit/W=(149,124,545,373) time_data,conc_data,conc_model,weight_factor
    ModifyTable width(time_data)=64,width(conc_data)=72
EndMacro

// グラフウィンドウの表示
Window Kinetics_plot() : Graph
    PauseUpdate; Silent 1 // ウィンドウの作成中
    Display /W=(83,86,485,350) conc_model vs time_data
    AppendToGraph fit_conc_data
    AppendToGraph conc_data vs time_data
    ModifyGraph mode(conc_model)=3,mode(conc_data)=3
    ModifyGraph marker(conc_model)=16,marker(conc_data)=8
    ModifyGraph rgb(conc_model)=(0,0,0),rgb(fit_conc_data)=(1,3,39321)
    ModifyGraph msize(conc_model)=1
    ModifyGraph grid=2
    ModifyGraph mirror=2
    // 軸ラベル
    Label left "Concentration"
    Label bottom "Time"
    SetAxis/A/E=1 left
    // 右上のテキストボックス
    TextBox/N=text0/A=MC/X=31.31/Y=40.93 YYZ12YYJCRReaction KineticsYrdC/dt = -k
CYYSn
    SetDrawLayer UserFront
    DrawText -0.0309032498158258,-0.431590686541182,"dC/dt = - k C^n"
    DrawText -0.00333411812955211,-0.417854422804918,"dC/dt = - k CYYSn"
EndMacro

```

```

// パネルの各要素の定義 (種類、大きさ、機能)
Window ODE_Demo_Panel() : Panel
    PauseUpdate; Silent 1 // ウィンドウの作成中
    NewPanel /W=(20,48,282,453) as "ODE Demo"
    // Model values セクション
    SetVariable SetInitialConcentration,pos={34,65},size={183,15},title="Initial
Concentration:"
    SetVariable SetInitialConcentration,limits={0,Inf,1},value=
InitialC,bodyWidth= 75
    SetVariable SetRateCoefficient,pos={59,94},size={158,15},title="Rate
Coefficient:"
    SetVariable SetRateCoefficient,limits={0,Inf,0.1},value=
RateCoefficient,bodyWidth= 75
    SetVariable SetReactionOrder,pos={63,127},size={154,15},title="Reaction
Order:"
    SetVariable SetReactionOrder,limits={-Inf,Inf,0.25},value=
ReactionOrder,bodyWidth= 75
    // 上段・中段のボタン3つ
    Button
InitialGuessTryItButton,pos={21,175},size={70,20},proc=Enter_initial_guesses,title="T
ry It!"
    Button
GraphForwardButton,pos={53,8},size={150,20},proc=GraphForwardButtonProc,title="Bring
Graph Forward"
    CheckBox AddWeightingCheckBox,pos={155,200},size={87,14},title="Use Weighting"
    CheckBox AddWeightingCheckBox,value= 1
    Button
FitItButton,pos={165,175},size={70,20},proc=Integrate_and_Fit,title="Fit It!"
    // Fit Results セクション
    ValDisplay DisplayInitC,pos={41,264},size={168,14},title="Initial
Concentration:"
    ValDisplay DisplayInitC,limits={0,0,0},barmisc={0,1000},bodyWidth= 60
    ValDisplay DisplayInitC,value= #"fit_InitialC"
    ValDisplay DisplayRateK,pos={66,294},size={143,14},title="Rate Coefficient:"
    ValDisplay DisplayRateK,format="%g"
    ValDisplay DisplayRateK,limits={0,0,0},barmisc={0,1000},bodyWidth= 60
    ValDisplay DisplayRateK,value= #"fit_RateCoefficient"
    ValDisplay DisplayReactionOrder,pos={70,323},size={139,14},title="Reaction
Order:"
    ValDisplay DisplayReactionOrder,format="%g"
    ValDisplay DisplayReactionOrder,limits={0,0,0},barmisc={0,1000},bodyWidth= 60
    ValDisplay DisplayReactionOrder,value= #"fit_ReactionOrder"
    GroupBox ResultsGroup,pos={9,237},size={241,112},title="Fit Results",fStyle=1
    GroupBox modelSettingsGroup,pos={9,42},size={241,112},title="Model values"
    GroupBox modelSettingsGroup,fStyle=1
    PopupMenu
ODEHelpButton,pos={54,374},size={153,20},proc=HelpMenuProc,title="ODE Documentation"

```

```

    PopupMenu ODEHelpButton,mode=0,value= #"¥"Solving ODE's;All-At-Once Fit
Functions;IntegrateODE reference¥""
EndMacro

// パネル下部のヘルプのメニューアイテム
Function HelpMenuProc(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr

    Switch (popNum)
        case 1:
            DisplayHelpTopic "Solving Differential Equations"
            break
        case 2:
            DisplayHelpTopic "User-Defined Fitting Function: Detailed
Description[All-at-once Fitting Functions]"
            break
        case 3:
            DisplayHelpTopic "Operations[IntegrateODE]"
            break
    endswitch
End

```