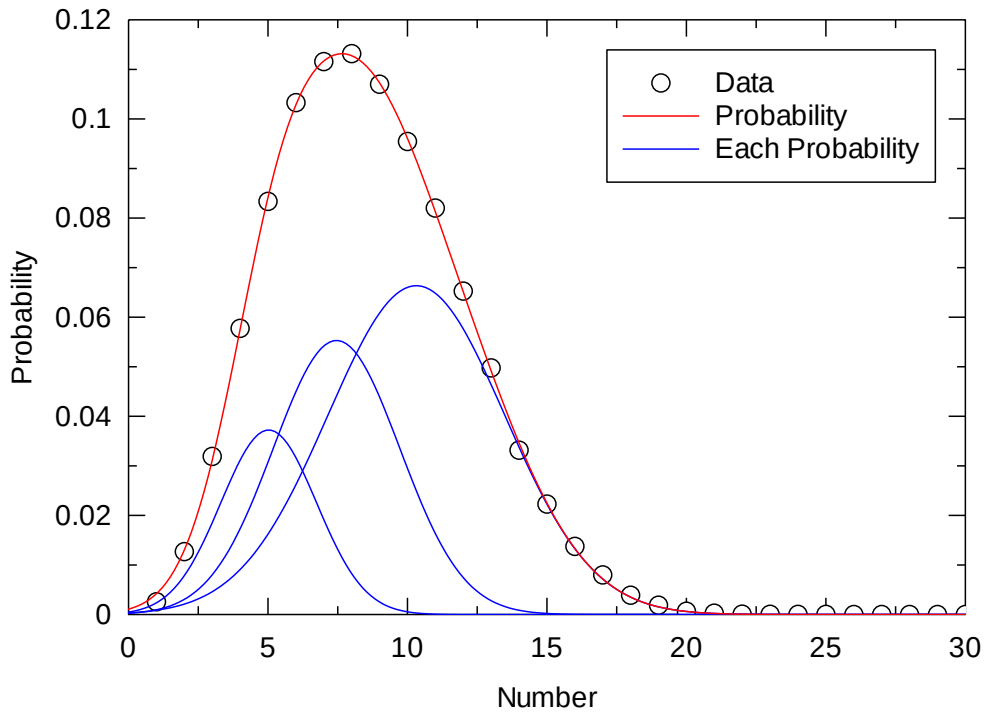
カーブフィット　バージョン 1.1

1. はじめに

1.1 カーブフィットの特徴

- 共役勾配法（Conjugate Gradient：CG）および焼きなまし法（Simulated Annealing：SA）を用いた非線形最小二乗法によるカーブフィッティングができる。
- Visual C++ネイティブ形式で開発しているため動作が高速。また、計算時間を要するフィッティングを別スレッドで実行できる。
- 高精細な科学論文用 2 次元グラフの作成し、それを拡張メタファイル形式で出力できる。
- CSV 形式でのデータ読み込みおよびフィッティング結果の出力ができる。
- フィッティング関数として正規分布、ポアソン分布、対数正規分布、多項式、指数関数、べき関数、サイン関数を使用できる。
- ユーザー独自のフィッティング関数 DLL（ダイナミックリンクライブラリ）を作成し、それを組み込むことができる（無償版の Visual C++ 2008, 2010 Express Edition で作成可能）。



上の図は共役勾配法により、ある確率分布を 3 つの正規分布に分解した結果を示しています。この場合のデータ点数は 30 個、パラメータ数は 9 個ですがフィッティングは数秒で完了します（計算時間はデータ数、パラメータの初期値および関数形に依存しま

す)。また、上の図は拡張メタファイル形式でグラフをクリップボードにコピーしてワードに直接貼り付けたものです。グラフが高精細であるため拡大してもシャギー（ギザギザ感）は殆ど見られません。

1.2 使用上の注意

- 本ソフトウェア（プログラム、マニュアル、付属ファイル）の著作権は杉尾健次郎が有します。
- 本ソフトウェアはフリーソフトであり、個人および法人を問わず自由に利用できます。
- 本ソフトウェアを使用する事によって発生した損失および損害に対して著作権者（杉尾健次郎）は一切責任を負いません。
- 本ソフトウエアの転載・配布は自由ですが、改変を行わずこのままの形態で配布してください。また、書籍、雑誌等への収録については、あらかじめご連絡ください。
- 本ソフトウェアに対して修正を加えること又は逆コンパイル、逆アセンブルを行うことはできません。
- 本ソフトウェアに関する要望、質問、バグ情報などがありましたら、杉尾健次郎まで電子メール（cvfit2010@yahoo.co.jp）にてご連絡をお願いします。

1.3 インストールおよびアンインストール

インストーラー（setup.exe）を起動して、その指示に従いカーブフィットをインストールします。コントロールパネルのプログラムのアンインストールからカーブフィットを削除できます。

1.4 配布ファイル

setup.exe	インストーラー
cvfit.msi	インストーラーパッケージ
cvfitManual-J.pdf	日本語マニュアル（このファイル）
sample フォルダ	サンプルファイルのフォルダ
*.cvfit	
user_func フォルダ	フィッティング関数 DLL の作成例を含むフォルダ
user.sln	Visual Studio 2008 用のソリューションファイル
user.vcproj	Visual C++ 2008 用のプロジェクトファイル
user.h	ヘッダファイルのコーディング例
user.cpp	ソースファイルのコーディング例
fit.h	カーブフィット用のライブラリのヘッダファイル
cvfitlib.lib	カーブフィット用のインポートライブラリ

1.5 更新履歴

バージョン	更新日	更新内容
1.0	2010.05.17	新規公開
1.1	2011.07.09	インストーラーでの配布に変更 手動フィッティングの追加 最小化アルゴリズムの変更（二分法→黄金分割法）

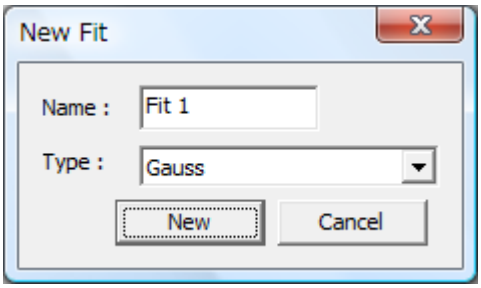
2. カーブフィットの方法

カーブフィットは以下の手順にて行います。

- [Fit][New Fit]で新しい Fit を作成する。
- [Fit][Data]で Fit にデータを登録する。
- [Fit][Function]で Fit に関数のパラメータを登録する。
- [Fit][Manual Fit]にて関数がデータに大体合うように、関数のパラメータを変化さる。
- [Fit][CG Fit]または[Fit][SA Fit]にて関数のパラメータをデータにフィッティングする。

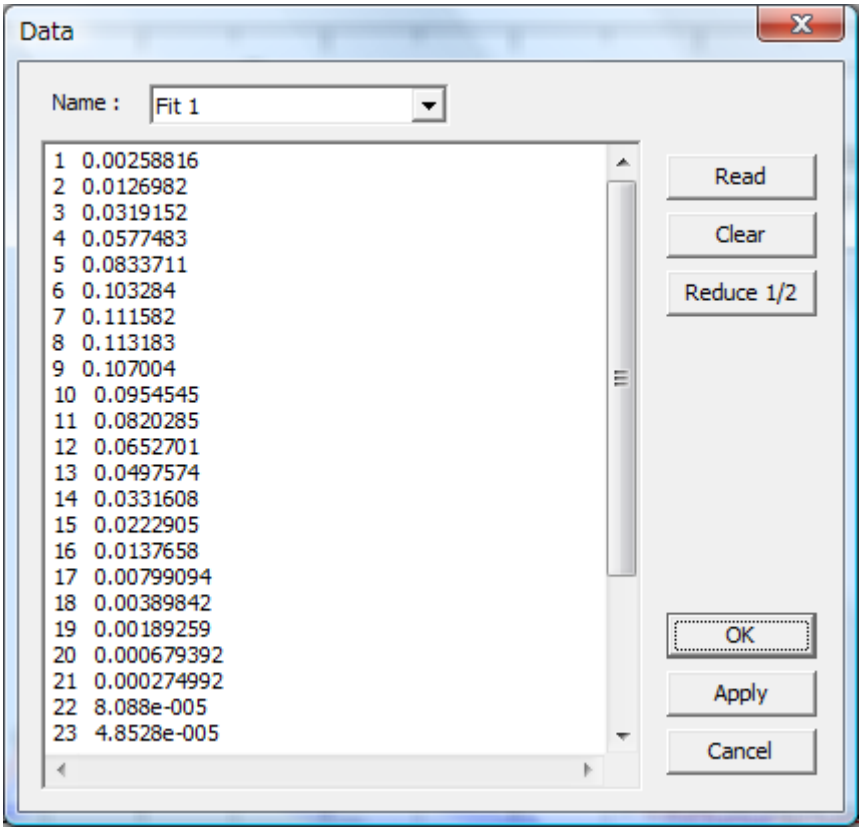
2.1 新しい Fit の作成

メニューから[Fit][New Fit]を選択すると次のようなダイアログが立ち上がります。Name を指定して Type（関数タイプ）をリストから選択した後、[New]を押すと新しい Fit を作成することができます。ここで作成された Fit の Type はこれ以降変更することができないので慎重に選択しておく必要があります。また、これ以降ここで登録した Fit に対してデータ、パラメータ、プロットスタイルの登録を行います。



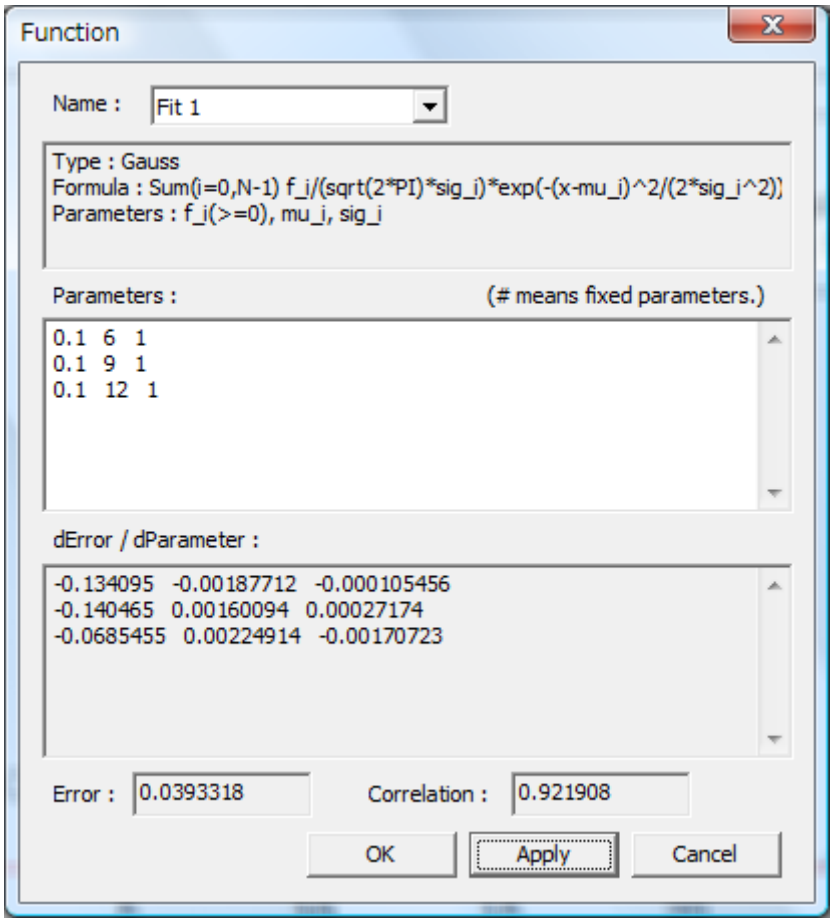
2.2 Fit へのデータの登録

メニューから[Fit][Data]を選択すると次のようなダイアログが立ち上がります。まず、データを登録したい Fit 名（Name）を上段のリストから選択します。データはテキストボックスにスペース区切りのテキスト形式で入力します。この時、列の並びは 1 列目が X 軸の値、2 列目が Y 軸の値です。また、エラーバー付きのグラフを作成したい場合は 3 列目に Y 軸の最小値、4 列目に Y 軸の最大値を入力します。[Read]ボタンを押すとファイルダイアログが立ち上がるので、テキストファイル（.txt）又は CSV ファイル（.csv）をテキストボックスに直接読み込むことができます。また、クリップボードからテキストボックスにデータを張り付けることも可能です。[Clear]ボタンを押すとテキストボックスはクリアされます。[Reduce 1/2]ボタンを押すとデータを平均してデータ数を 2 分の 1 に減らすことができます。入力を終了しデータを Fit に登録する場合は、[OK]または[Apply]ボタンを押します。



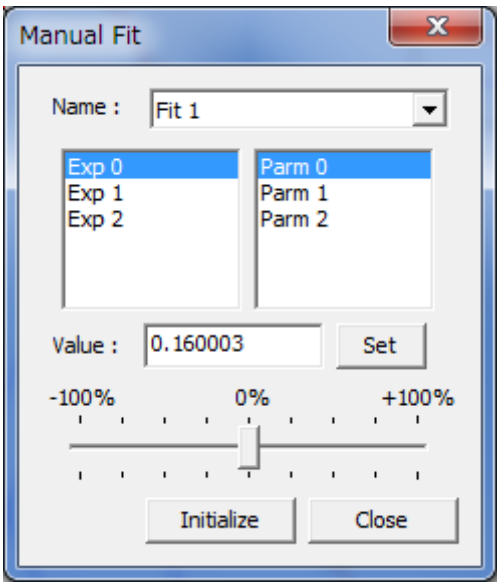
2.3 Fit への関数パラメータの登録

メニューから[Fit][Function]を選択すると次のようなダイアログが立ち上がります。まず、関数パラメータを登録したい Fit 名（Name）を上段のリストから選択します。関数タイプ（Type）、式（Formula）、パラメータ（Parameters）がヘルプとして表示されるので、それに従ってテキストボックスにスペース区切りのテキスト形式でパラメータを入力します。この時、数値の後ろに#を付けるとそのパラメータは固定パラメータとなります。ダイアログ例では 3 行 3 列のパラメータが指定されており、これは 3 つのガウス分布（1 つのガウス分布は 3 パラメータで表現）の足し合わせでフィッティングカーブを表現するという意味です（フィッティング関数の詳細については第 4 章を参照してください）。データが登録されていれば、二乗誤差のそれぞれのパラメータでの偏微分（dError / dParameter）、二乗誤差（Error）および相関係数（Correlation）が表示されます。また、Error が小さくなるように適正な初期パラメータを登録しておけば、カーブフィッティングに費やす時間を短縮することができます。入力を終了しパラメータを Fit に登録する場合は、[OK]または[Apply]ボタンを押します。



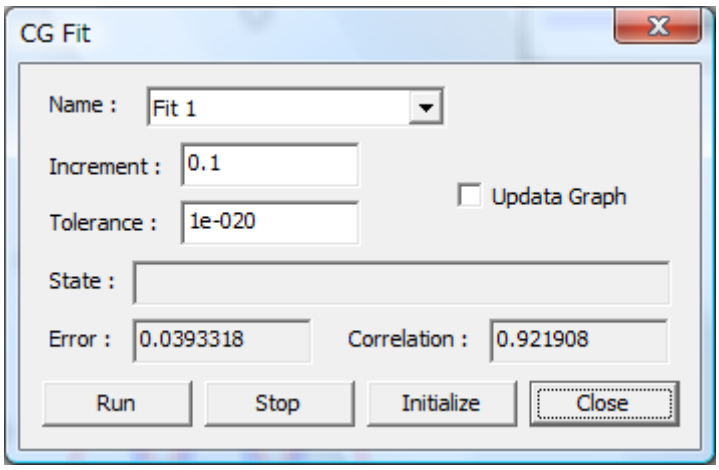
2.4 手動によるパラメータのフィッティング

メニューから[Fit][Manual Fit]を選択すると次のようなダイアログが立ち上がります。まず、カーブフィッティングを行いたい Fit 名（Name）を上段のリストから選択します。変化させたい式（Exp）とパラメータ（Parm）をリストから選びスライダーで値（Value）を変化させます。値は直接入力することもできます。Exp と Parm の選択を変えながら、関数がデータに大体合うようにパラメータを調整します。初期値に戻す場合は[Initialize]ボタンを押します。



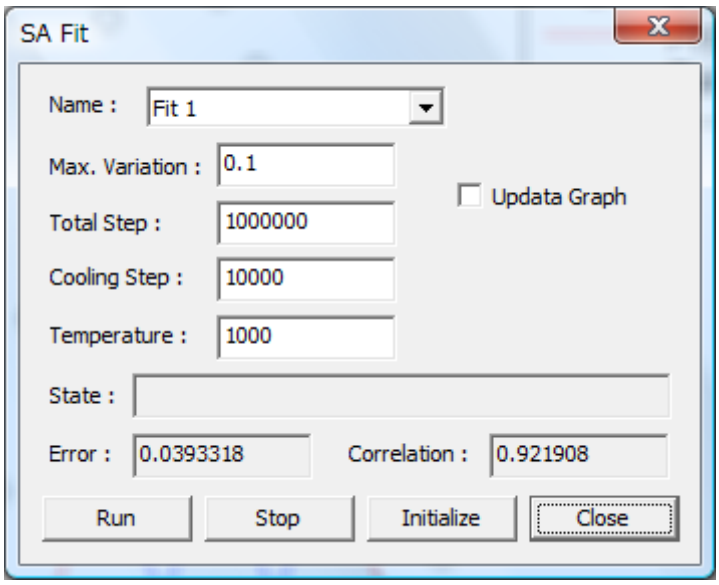
2.5 CG Fit（共役勾配法）によるパラメータのフィッティング

メニューから[Fit][CG Fit]を選択すると次のようなダイアログが立ち上がります。まず、カーブフィッティングを行いたい Fit 名（Name）を上段のリストから選択します。パラメータの初期増加量（Increment）と許容差（Tolerance）を指定し、[Run]ボタンを押してフィッティングを開始します。グラフを自動更新する場合は Update Graph にチェックを入れます。異常終了（Infinity or nonnumeric is appeard.）する場合は Increment を小さくして再度フィッティングを試みてください。Error の変化量が Tolerance 以下になればフィッティングは終了しますが、もし、途中で止めたい場合は[Stop]ボタンを押します。パラメータを初期値に戻す場合は[Initialize]ボタンを押します。



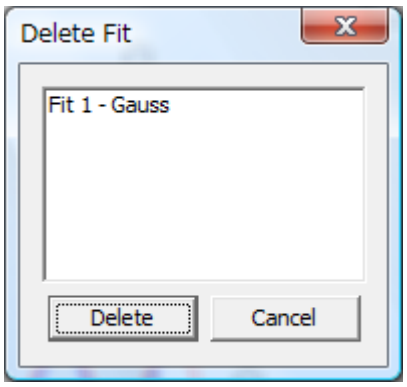
2.6 SA Fit（焼きなまし法）によるパラメータのフィッティング

メニューから[Fit][SA Fit]を選択すると次のようなダイアログが立ち上がります。まず、カーブフィッティングを行いたい Fit 名（Name）を上段のリストから選択します。パラメータの最大変化量（Max. Variation）、総ステップ数（Total Step）、冷却ステップ（Cooling Step）と初期温度（Temperature）を指定し、[Run]ボタンを押してフィッティングを開始します。グラフを自動更新する場合は Update Graph にチェックを入れます。Cooling Step おきに Temperature を 10%減少させながらアニーリングを行います。Error がうまく減少しない場合は Max. Variation と Temperature を調整して再度フィッティングを試みてください。Total Step 回ループが回ればフィッティングは終了しますが、もし、途中で止めたい場合は[Stop]ボタンを押します。パラメータを初期値に戻す場合は[Initialize]ボタンを押します。



2.7 Fit の削除

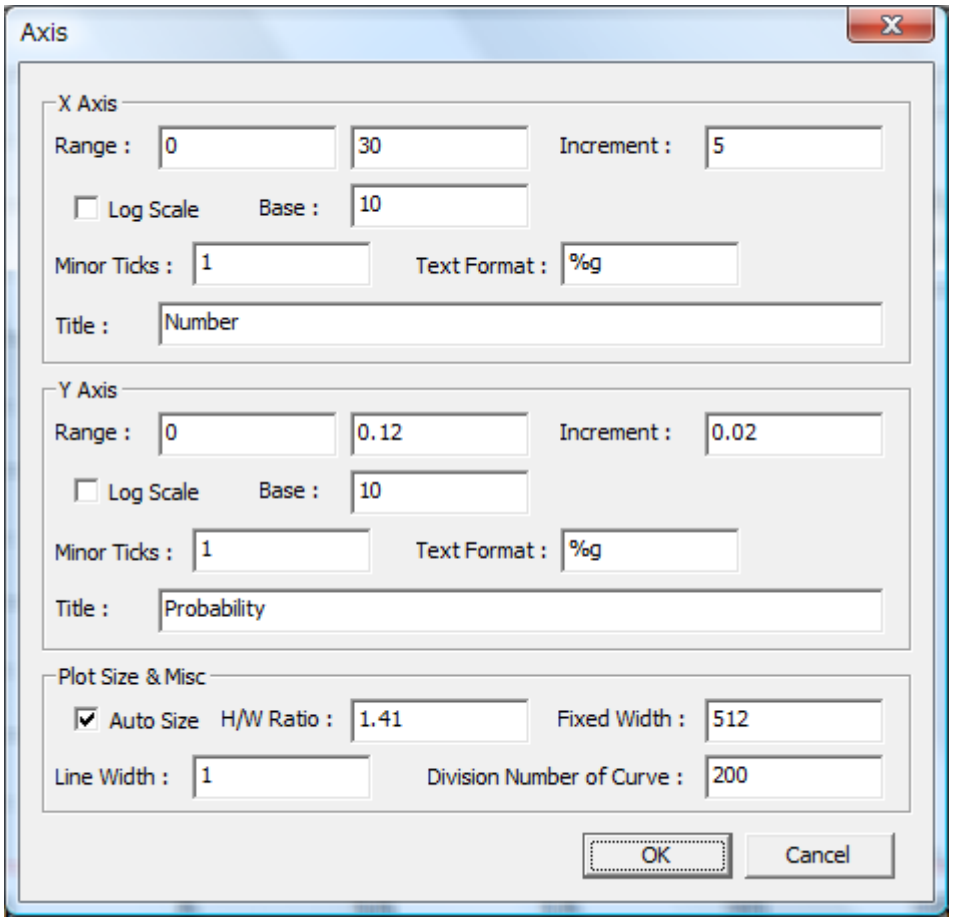
メニューから[Fit][Delete Fit]を選択すると次のようなダイアログが立ち上がります。リストから削除したい Fit 名を選択して[Delete]ボタンを押すとその Fit は削除されます。Fit を削除するとそこに登録されているデータ、パラメータ、プロットスタイルはすべて消去されます。



3. グラフの作成

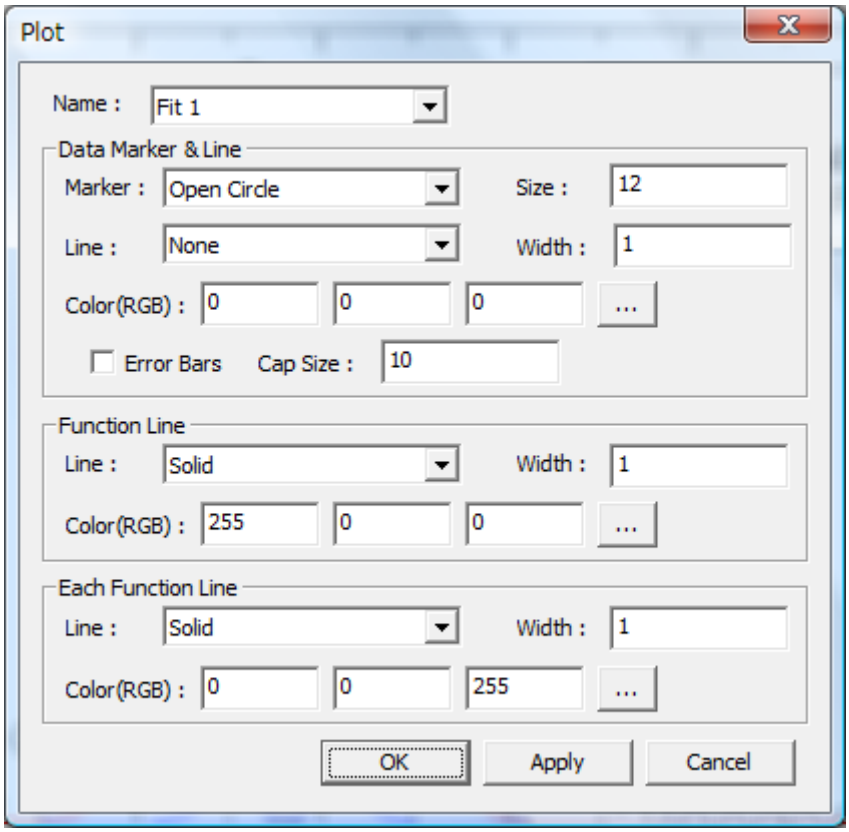
3.1 軸の設定

メニューから[View][Axis]を選択すると次のようなダイアログが立ち上がります。このダイアログでは X,Y 軸の範囲（Range）、目盛の増加量（Increment）、対数目盛表示の選択（Log Scale）、対数の底（Base）、副目盛の数（Minor Ticks）、目盛の文字フォーマット（Text Format：printf 型の書式文字列）、軸タイトル（Title）を設定することができます。また、プロットサイズ自動調整の選択（Auto Size）、縦横比（H/W Ratio）、自動調整しない場合のプロット幅（Fixed Width：ポイント単位）、軸の線幅（Line Width：ポイント単位）、フィッティングカーブの描画分割数（Division Number of Curve）を設定することができます。



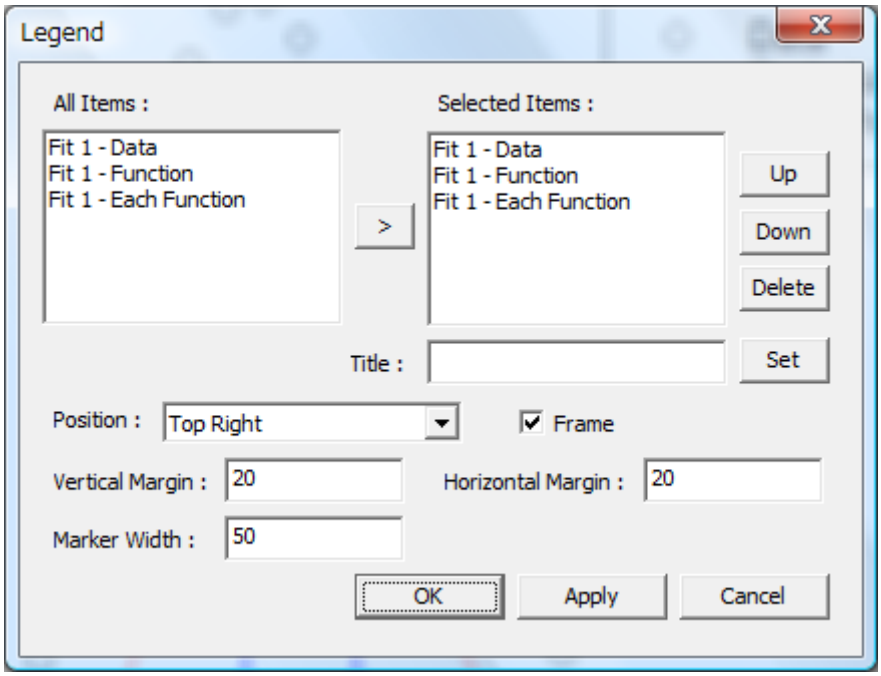
3.2 プロットスタイル

メニューから[View][Plot]を選択すると次のようなダイアログが立ち上がります。プロットスタイルを設定したい Fit 名（Name）を上段のリストから選択します。このダイアログではデータマーカーの種類（Marker）、マーカーサイズ（Size：ポイント単位）、マーカーをつなぐ線の種類（Line）、線幅（Width：ポイント単位）、色（Color）、エラーバー表示の選択（Error Bars）、エラーバーの幅（Cap Size：ポイント単位）を指定することができます。また、フィッティングカーブ（Function Line）および足し合わされるそれぞれのカーブ（Each Function Line）の線種（Line）、線幅（Width：ポイント単位）、色（Color）を設定することができます。変更したプロットスタイルを Fit に登録する場合は、[OK]または[Apply]ボタンを押します。



3.3 凡例

メニューから[View][Legend]を選択すると次のようなダイアログが立ち上がります。表示したい凡例を全アイテム（All Items）から選び[>]ボタンを押して選択アイテム（Selected Item）に追加します。ここで、表示文字列を変更する場合はタイトル（Title）を編集して[Set]ボタンを押します。また、凡例の順序は[Up][Down]ボタンで入れかえることができ、不要な凡例は[Delete]ボタンで消去することができます。さらに、このダイアログでは凡例の位置（Position）、枠の表示の有無（Frame）、垂直マージン（Vertical Margin）、水平マージン（Horizontal Margin）、マーカーの表示幅（Marker Width）を設定することができます。



3.4 フォントの設定

メニューから[View][Font]を選択するとフォントダイアログが立ち上がります。初期フォントは Arial ですが日本語フォントを設定すると日本語の表示も可能です。

3.5 グラフのコピー

メニューから[View][Copy Graph]を選択すると表示されているグラフが拡張メタファイル形式でクリップボードにコピーされます。そして、それはワードやパワーポイントに直接張り付けることができます。

4. フィッティング関数

4.1 正規分布（Gauss）

パラメータは N行 3 列で指定します（ここで Nは足し合わせの数）。

式 $$F=\sum_{i=0}^{N-1}\frac{f_i}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

各パラメータ $f_i(\geq 0),\mu_i,\sigma_i$

4.2 ポアソン分布（Poisson）

パラメータは N行 3 列で指定します（ここで Nは足し合わせの数）。x が実数値をとる場合があるため階乗の計算にはガンマ関数を用います。通常のポアソン分布としてフィッテングする場合はb_i を 0 に固定してください。

式 $$F=\sum_{i=0}^{N-1}\frac{f_i a_i^{x-b_i}}{\Gamma(x-b_i+1)}\exp(-a_i)$$

各パラメータ $f_i(\geq 0),a_i(>0),b_i(\geq 0)$

4.3 対数正規分布（Lognormal）

パラメータは N行 3 列で指定します（ここで Nは足し合わせの数）。

式 $$F=\sum_{i=0}^{N-1}\frac{f_i}{\sqrt{2\pi}\sigma_i x}\exp\left(-\frac{(\log(x)-\mu_i)^2}{2\sigma_i^2}\right)$$

各パラメータ $f_i(\geq 0),\mu_i,\sigma_i$

4.4 多項式（Polynomial）

パラメータは N行 1 列で指定します（ここで Nは足し合わせの数）。

式 $$F=\sum_{i=0}^{N-1}a_i x^i$$

各パラメータ a_i

4.5 指数関数（Exponential）

パラメータは N行 2 列で指定します（ここで Nは足し合わせの数）。

式 $$F=\sum_{i=0}^{N-1}a_i\exp(b_i x)$$

各パラメータ a_i,b_i

4.6 べき関数（Power）

パラメータは N行 2 列で指定します（ここで Nは足し合わせの数)。

式　　$F = \sum_{i=0}^{N-1} a_i x^{b_i}$

各パラメータ　　a_i, b_i

4.7 サイン関数（Sin）

パラメータは N行 3 列で指定します（ここで Nは足し合わせの数)。

式　　$F = \sum_{i=0}^{N-1} a_i \sin(b_i x + c_i)$

各パラメータ　　$a_i, b_i, c_i (0 \le c_i \le \pi/2)$

5. フィッティング関数 DLL（ダイナミックリンクライブラリ）の作成

5.1 カーブフィッティングについて

二乗誤差（Error：E）は次のように表すことができます。

$$E = \sum_{j=0}^{M} \left(y_j - F_s(x_j) \right)^2$$

ここで、Mはデータ数、x_i, y_iは各データ、F_sはフィッティング関数です。また、F_sをN個の関数F_i足し合わせとして次のように表します。

$$F_s = \sum_{i=0}^{N-1} F_i(a_i, b_i, c_i \ldots)$$

カーブフィッティングでは E が最小となるようなF_iのパラメータ$(a_i, b_i, c_i \ldots)$を探索する必要があり、そのために本プログラムでは共役勾配法（CG Fit）と焼きなまし法（SA Fit）を用います。また、共役勾配法ではEの勾配を計算する必要があり、例えばEのa_iに関する偏微分は次のように計算します。

$$\frac{\partial E}{\partial a_i} = -2 \sum_{j=0}^{M} \left(y_j - F_s(x_j) \right) \frac{\partial F_s(x_j)}{\partial a_i}$$

$$\frac{\partial F_s(x_j)}{\partial a_i} = \sum_{i=0}^{N-1} \frac{\partial F_i(x_j)}{\partial a_i}$$

本プログラムではF_iとその偏微分（$\partial F_i(x_j) / \partial a_i$ 等）をオーバーライドしてユーザー独自のフィッティング関数 DLL を作成することができます。また、F_sとその偏微分（$\partial F_s(x_j) / \partial a_i$ 等）をオーバーライドすることも可能です。

5.2 DLL の作成方法

Visual Studio 2008 を用いた DLL の作成方法について以下に説明します（無償版の Visual C++ 2008, 2010 Express Edition でも作成可能)。Visual Studio 2008 を立ち上げて新しいプロジェクトを「Win32 プロジェクト」で作成します。この時アプリケーションの設定ーアプリケーションの種類において「DLL」を選択します。ソリューションの構成を「Release」に切り替えます。プロパティー構成プロパティを「マルチバイト文字セットを使用する」に設定します。ヘッダファイル（fit.h）とインポートライブラリファイル（cvfitlib.lib）を開発フォルダにコピーして、プロパティー構成プロパティーリンカー追加の依存ファイルに cvfitlib.lib を設定します。

まず、ヘッダファイル(.h)を以下のようにコーディングします。

```
#pragma once
#include "fit.h"

class Cuser :
	public Cfit
{
public:
	Cuser();
	~Cuser();
public:
	char *FuncType();
	char *FuncHelp();
	double Func_i(double x, int i, double parm1[]);
	void DFunc_i(double x, int i, double parm1[], double dfunc1[]);
	void Limit_i(double parm1[]);
};
```

次にソースファイル（.cpp）を以下のようにコーディングします。ここでは、正規分布関数の例を示してあります。

```
#include "stdafx.h"
#include "user.h"
#include <math.h>
```

```
#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif

Cuser::Cuser()
{
	// number of parameter in formula
	nParm = 3;
}

Cuser::~Cuser()
{
}

char *Cuser::FuncType()
{
	return "User";
}

char *Cuser::FuncHelp()
{
	return "Formula : Sum(i=0,N-1) f_i/(sqrt(2*PI)*sig_i)*exp(-(x-mu_i)^2/(2*sig_i^2))¥r¥n
		Parameters : f_i(>=0), mu_i, sig_i¥r¥n";
}

double Cuser::Func_i(double x, int i, double parm1[])
{
	double f = parm1[0];
	double mu = parm1[1];
	double sig = parm1[2];
	return f/(sqrt(2.0*M_PI)*sig)*exp(-pow(x-mu,2.0)/2.0/sig/sig);
}

void Cuser::DFunc_i(double x, int i, double parm1[], double dfunc1[])
{
	double f = parm1[0];
```

```
        double mu = parm1[1];
        double sig = parm1[2];
        dfunc1[0] = 1.0/(sqrt(2.0*M_PI)*sig)*exp(-pow(x-mu,2.0)/2.0/sig/sig);
        dfunc1[1] = f*dfunc1[0]*(x-mu)/sig/sig;
        dfunc1[2] = f*dfunc1[0]*(pow(x-mu,2.0)/pow(sig,3.0)-1.0/sig);
}

void Cuser::Limit_i(double parm1[])
{
        if (parm1[0] < 0.0) parm1[0] = 0.0;
}

extern "C" __declspec(dllexport) Cfit *CVFIT_CreateFit()
{
        return new Cuser;
}
```

例では、コンストラクタで nParm（F_i のパラメータ数）を設定し、FuncType 関数、FuncHelp 関数、Func_i 関数、DFunc_i 関数、Limit_i 関数をオーバーライドしています。FuncType 関数と FuncHelp 関数では関数のタイプ名とヘルプをそれぞれ定義しています。正規分布関数の場合 F_i は次のように表され、それを Func_i 関数で定義しています。

$$F_i = \frac{f_i}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

また、次のように表される F_i の偏微分を DFunc_i 関数で定義しています。

$$\frac{\partial F_i}{\partial f_i} = \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

$$\frac{\partial F_i}{\partial \mu_i} = \frac{f_i}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)\times\frac{x-\mu_i}{\sigma_i^2}$$

$$\frac{\partial F_i}{\partial \sigma_i} = \frac{f_i}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)\times\left(\frac{(x-\mu_i)^2}{\sigma_i^3}-\frac{1}{\sigma_i}\right)$$

また、$f_i \geq 0$ なので Limit_i 関数において $f_i < 0$ とならないようにパラメータの変更に制限をかけています。CVFIT_CreateFit は DLL が読み込まれた時に呼び出される関数

であり、必ずエクスポートしておく必要があります。

ビルドすると Release ディレクトリに DLL ファイル（.dll）が作成されるので、カーブフィット(cvfit.exe)を起動してそれを[File][Load Function]で読み込んでみます。読み込み時にエラーが発生せず、また、[Fit][New Fit]の関数タイプのリストに作成した関数が新しく登録されていれば DLL の作成は成功していると考えられます。カーブフィット（cvfit.exe）は起動時に起動ディレクトを走査してフィッティング関数 DLL を自動的にロードするので、新たに作成した DLL を起動ディレクトにコピーしておきます。