

# GlobalOptimization Toolbox によるカーブフィッティング

非線形フィッティングを行うには Maple では Statistics パッケージに NonlinearFit コマンドが組み込みで用意されていますが、非線形フィッティングは最適化問題でもあり必ずしもうまくいくとは限りません。この事例では NonlinearFit ではうまくいかない場合の対処法として GlobalOptimization Toolbox (別売り) を使用したフィッティング方法を紹介し

> restart;

## フィッティングデータの作成

今回フィッティングの為にデータはベッセル関数含む式から作成します。この関数は、光学分野の円形開口における回折の関数です。関数に含まれるパラメータはそれぞれ以下の意味を持ちます。

a: 横軸の移動量 (最大値をとる横軸の位置)

b: 開口サイズの補正項

c: 最大値の値

```
> model := ((2*BesselJ(1, (x-a)/b))/((x-a)/b))^2 * c;
```

$$model := \frac{4 \operatorname{BesselJ}\left(1, \frac{x-a}{b}\right)^2 b^2 c}{(x-a)^2} \quad (1)$$

式に含まれるパラメータの値は次のようにします。

```
> params := {a=100.1, b=10, c=3/4};
```

$$params := \left\{ a = 100.1, b = 10, c = \frac{3}{4} \right\} \quad (2)$$

```
> base_eq := eval(model, params);
```

```
base_eq :=
```

$$\frac{1}{(x-100.1)^2} \left( 300 \operatorname{BesselJ}\left(1, \frac{1}{10} x - 10.01000000\right)^2 \right)$$

関数からフィッティングで使用するためのデータを算出します。

```
> pdata := [seq([i, subs(x=i, base_eq)], i=1..200)];
```

```
> Xd := pdata[1..-1, 1];
```

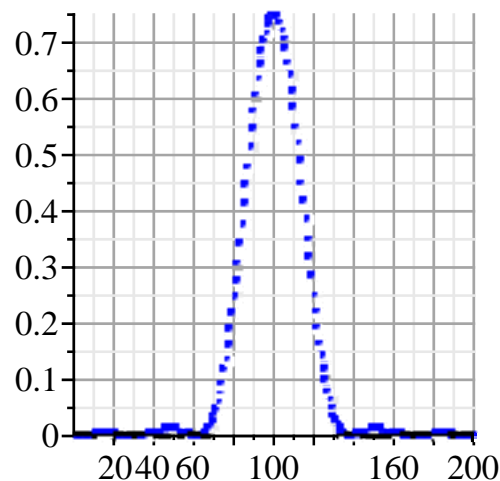
```
> Yd := pdata[1..-1, 2];
```

```
>
```

データ点を確認します。

```
> p1 := plots[pointplot](pdata, symbol=circle, symbolsize=10, color=blue, gridlines=true);
```

```
> p1;
```



今回のフィッティングではデータは式から直接算出したデータを使用しております。実際のフィッティングで使用されるデータには計測誤差や製造誤差が含まれているケー

スがありますので、フィッティングの前に使用されるデータに対してデータの特異値の除去などが必要になる場合があります。

## Statistics[NonlinearFit] を使用したフィッティング

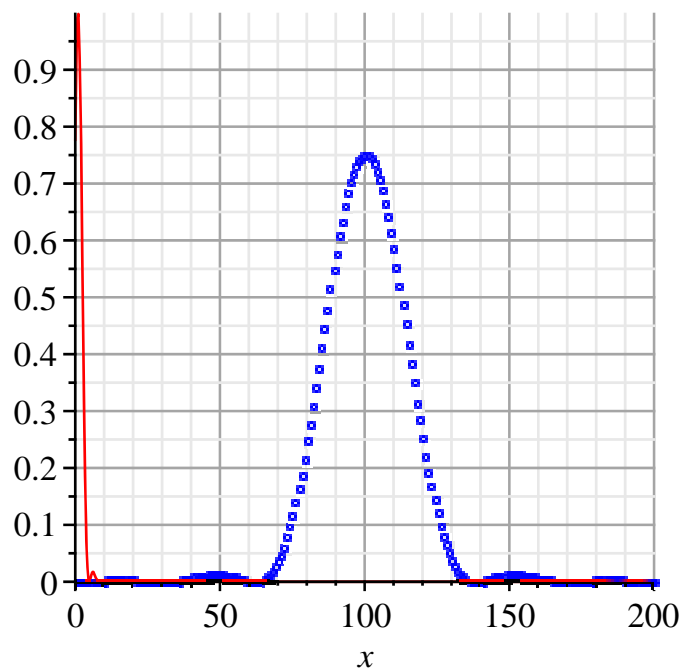
まず Statistics パッケージの NonlinearFit コマンドを使用してフィッティングを試みます。

```
> NFSol := Statistics[NonlinearFit](model, Xd, Yd, x, output=
  parametervalues);
      NFSol := [a = 1., b = 1., c = 1.00000002108371144] (4)
```

```
> NFeq := subs(NFSol, model);
      NFeq := 
$$\frac{4.000000084 \text{ BesselJ}(1, 1. x - 1.)^2}{(x - 1.)^2}$$
 (5)
```

求まった式をデータ点と共にプロットします。

```
> p2 := plot(NFeq, x=0..200):
> plots[display](p1, p2);
```



NonlinearFit コマンドは内部的に局所的な最適化コマンドが使用されており、プロットからわかるように今回のケースではフィッティング結果はよくありません。

## GlobalOptimization Toolbox を使用したフィッティング

次に GlobalOptimization Toolbox の GlobalSolve コマンドを用いてフィッティングを行います。

まず座標点とモデル式から誤差の二乗の和の式を定義します。

```
> obeq := expand(add((Yd[i] - subs(x=Xd[i], model))^2, i=1..nops(Xd))):
```

GlobalSolve を使用しフィッティングを実行します。

GlobalSolve の結果の 1 つ目の値は目的関数の最小値(誤差の二乗和の最小値)、2 つ目は各パラメータの値となります。

```
> GOTSol := GlobalOptimization[GlobalSolve](obeq, a=0..200, b=5.
  .15, c=0..1, timelimit=3600);
GOTSol := [2.39208929999999986 10-8, [a = 100.099999911068679, b (6)
```

```
= 10.0000001358960482, c = 0.749999985336130326]]
```

GlobalSolve は近似解法を使用し、かつ倍精度で計算するため近似解が返されます。また、求まった結果に対して何桁までを使用するかによって結果は変わってきます。例えば、結果の値を 10 桁と 7 桁で評価すると次のようになります。

```
> vars_dig10 := evalf(GOTSol[2], 10);  
vars_dig10 := [a = 100.0999999, b = 10.00000014, c = 0.7499999853] (7)
```

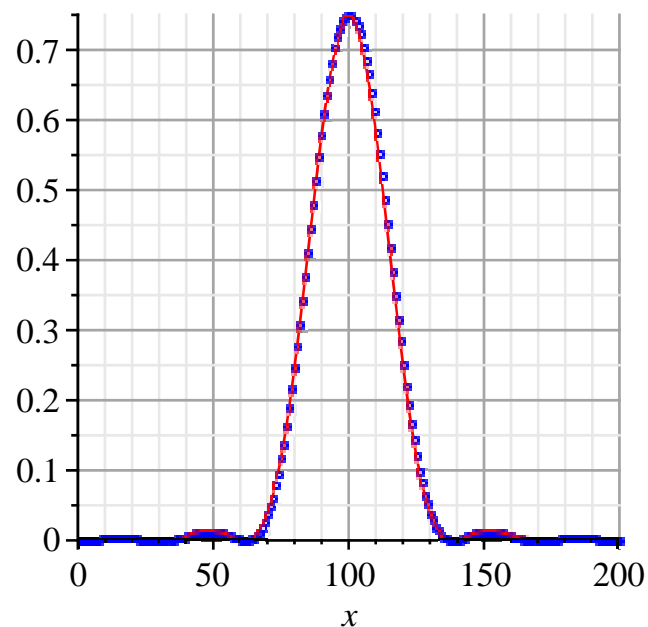
```
> vars_dig7 := evalf(GOTSol[2], 7);  
vars_dig7 := [a = 100.1000, b = 10.00000, c = 0.7500000] (8)
```

上記のように 10 桁とすると微妙な誤差が残っていますが、7 桁とすると正確なパラメータの値となります。

次の章では誤差の検証を行うため、今回は 10 桁の値を解として使用します。

```
> FitEq := subs(vars_dig10, model);  
FitEq := 
$$\frac{300.0000025 \text{ BesselJ}(1, 0.09999999860 x - 10.00999985)^2}{(x - 100.0999999)^2}$$
 (9)
```

```
> p3 := plot(FitEq, x=0..200);  
> plots[display](p1, p3);
```

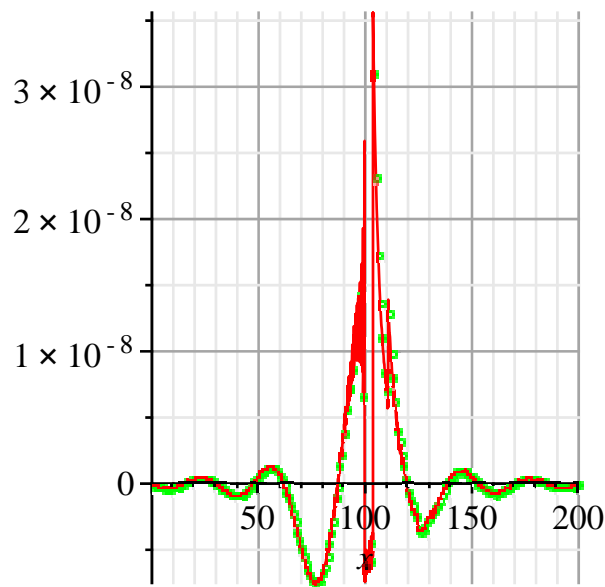


## 結果の検証

各点での誤差と関数同士の差のプロット

各点での誤差を計算したものと、設計式とフィッティングで得られた結果の差を重ね描きします。

```
> dif_data := [seq([Xd[i], evalf(Yd[i]-subs(x=Xd[i], FitEq))], i=1.  
.nops(Xd))];  
> p4 := plots[pointplot](dif_data, symbol=circle, symbolsize=10,  
color=green, gridlines=true);  
> p5 := plot(base_eq-FitEq, x=0..200, numpoints=1000);  
> plots[display](p4, p5);
```



### 統計的な検証

誤差量の絶対値のリストを作成します。

```
> dif_val := map(abs, dif_data[1..-1, 2]):
```

誤差の二乗和

```
> f := x -> x^2:
```

```
> dif_val2 := map(f, dif_val):
```

```
> convert(dif_val2, `+`):
```

4.422346638 10<sup>-15</sup>

(10)

誤差の総和

```
> convert(dif_val, `+`):
```

4.775723443 10<sup>-7</sup>

(11)

誤差の平均

```
> Statistics[Mean](dif_val):
```

2.387861722 10<sup>-9</sup>

(12)

誤差の標準偏差(不変分散)

```
> Statistics[StandardDeviation](dif_val):
```

4.061072659 10<sup>-9</sup>

(13)

## 使用環境

OS Windows XP SP2

CPU Intel Core 2 Duo T7600 2.33GHz

RAM 2GB

Maple 11.0 (Build ID: 277698) ワークシートの一括実行に要した時間：約 563 秒 (最適化計算に約 561 秒)  
GlobalOptimization Toolbox (時間はいくまでも目安です。その時の PC の状態により変化します。)