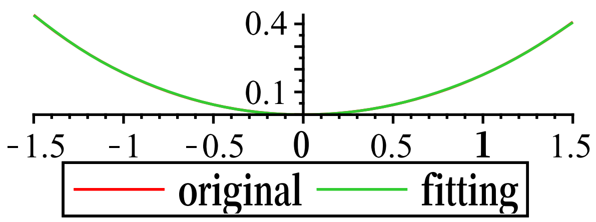
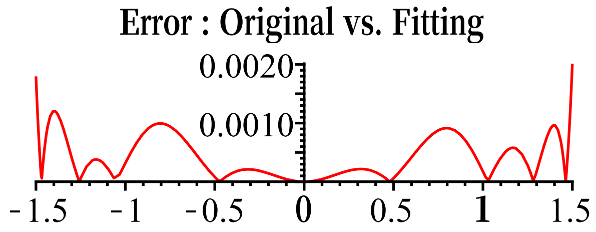
非球面レンズモデル式の同定





非球面レンズ式とサンプルデータ

まず、非球面レンズ式とサンプルデータの作成を行います。

> $restart$

非球面レンズ式は、以下の式で与えられます。ここで、c: 曲率、k: コーニック（円錐）定数、A1～A4: 補正係数です。

> $model := \dfrac{c \cdot x^2}{1 + \sqrt{1 - (1+k) \cdot c^2 \cdot x^2}} + A1\, x^4 + A2\, x^6 + A3\, x^8 + A4\, x^{10} :$

いま、我々は次の各定数値を持っていると仮定します；

> $param := \left\{ c = \dfrac{1}{2.87977}, k = -0.64159, A1 = 3.148\mathrm{e}{-4}, A2 = -2.546\mathrm{e}{-5}, A3 = -2.814\mathrm{e}{-6}, A4 = -3.307\mathrm{e}{-7} \right\}$

$$param := \{A1 = 0.0003148, A2 = -0.00002546, A3 = -0.000002814, A4 = -3.307\ 10^{-7}, c = 0.3472499540, k = -0.64159\} \quad (1.1)$$

このサンプルの定数値から、設計で用いたレンズ式 modelEq は次式となるとします。

> $modelEq := eval(model, param)$

$$modelEq := \frac{0.3472499540\, x^2}{1 + \sqrt{1 - 0.04321798479\, x^2}} + 0.0003148\, x^4 - 0.00002546\, x^6 - 0.000002814\, x^8 - 3.307\ 10^{-7}\, x^{10} \quad (1.2)$$

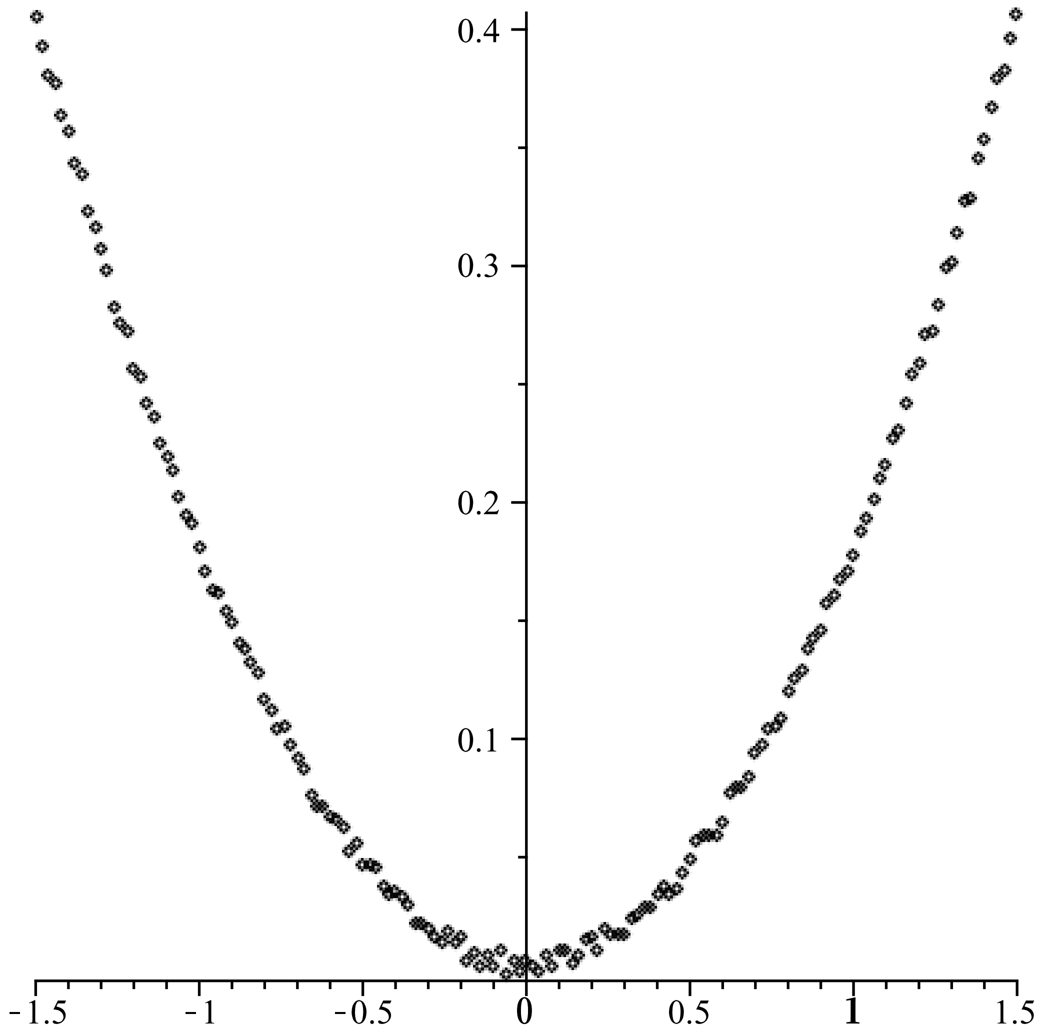
このレンズ式により生成されるサンプルデータを用意します。なお、ここで用意するサンプルデータには 10^{-14} 程度のノイズを載せていることに注意してください。

> $modelData := Matrix\left(\left[seq\left(\left[u, eval(modelEq, x = u) + \dfrac{rand(\,)}{10.0^{14}}\right], u = -1.5 .. 1.5, 0.02\right)\right]\right)$

$$modelData := \left[\begin{array}{c} \textit{151 x 2 Matrix} \\ \textit{Data Type: anything} \\ \textit{Storage: rectangular} \\ \textit{Order: Fortran\_order} \end{array}\right] \quad (1.3)$$

サンプルデータをプロットしてみます。

> $plots[pointplot](modelData)$



次では、このサンプルデータを用いてモデル式 model の各係数の同定を行ってみます。

モデル式の変形とフィッティング

非線形モデル式へのフィッティングは、通常 Statistics パッケージの NonlinearFit コマンドで行えます。

> $with(Statistics):$

NonlinearFit コマンドには、フィッティングを行うモデル式、X 座標値の行列（またはベクトル）と Y 座標値（ベクトル）およびモデル式の変数を与えます。

> $Xdata := modelData[1..-1, 1]$

$$Xdata := \left[\begin{array}{c} \text{1 .. 151 } Vector_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{array}\right] \quad (2.1)$$

> $Ydata := modelData[1..-1, 2]$

$$Ydata := \left[\begin{array}{c} \text{1 .. 151 } Vector_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{array}\right] \quad (2.2)$$

モデル式は、最初に定義した model です；

> model

$$\frac{c\,x^2}{1+\sqrt{1-c^2\,x^2-c^2\,x^2\,k}}+A1\,x^4+A2\,x^6+A3\,x^8+A4\,x^{10} \tag{2.3}$$

実際に NonlinearFit コマンドでフィッティングを行ってみます；

> NonlinearFit(model, Xdata, Ydata, x)

Error, (in Statistics:-NonlinearFit) complex value encountered

上記のエラーは、モデル式に対する最小二乗フィッティング中に複素数に対する処理が用意されていないために発生するものです。そこで、ここではモデル式 model をテイラー級数（多項式）に変形してからフィッティングを行ってみます；

次のコマンドは、model を $x=0$ の点で taylor コマンドによりテイラー級数へ変形しています。(行端のコロンを消去すると展開した結果を確認できます)

> modelTaylor := taylor(model, x = 0, 12) :

taylor コマンドの結果には剰余項が含まれているので、convert コマンドで純粋な多項式へと変換します。

> fitModelApp := expand(convert(modelTaylor, polynom)) :

また、モデルから作られたサンプルデータにはノイズが載っていることも判明しています。そこで、モデル式全体に誤差値 Err が含まれているとします；

> fitModelApp := sort(fitModelApp + Err)

$$fitModelApp := \frac{7}{256}\,c^9\,k^4\,x^{10}+\frac{7}{64}\,c^9\,k^3\,x^{10}+\frac{21}{128}\,c^9\,k^2\,x^{10}+\frac{7}{64}\,c^9\,k\,x^{10} + \frac{7}{256}\,c^9\,x^{10}+\frac{5}{128}\,c^7\,k^3\,x^8+\frac{15}{128}\,c^7\,k^2\,x^8+\frac{15}{128}\,c^7\,k\,x^8+\frac{5}{128}\,c^7\,x^8 + \frac{1}{16}\,c^5\,k^2\,x^6+\frac{1}{8}\,c^5\,k\,x^6+A4\,x^{10}+\frac{1}{16}\,c^5\,x^6+A3\,x^8+\frac{1}{8}\,c^3\,k\,x^4+A2\,x^6 + \frac{1}{8}\,c^3\,x^4+A1\,x^4+\frac{1}{2}\,c\,x^2+Err \tag{2.4}$$

この fitModelApp 式を用いて、改めてフィッティングを行ってみます。

なお、ここでは NonlinearFit コマンドの戻り値をパラメータ値とするために、output オプションに parametervalues を指定しています。また、フィッティングに要した時間も計測しておきます。

> st := time( ) :
  fitParam := NonlinearFit(fitModelApp, Xdata, Ydata, x, output = parametervalues);
  time( ) - st;

fitParam := [A1 = -0.127761600656121576, A2 = -0.326047218510421178, A3 = -0.621422564795779953, A4 = -1.65736613208662064, Err = 0.00497334805301575394, c = 0.340873966931799055, k = 30.4029264535317197]

$$1.547 \tag{2.5}$$

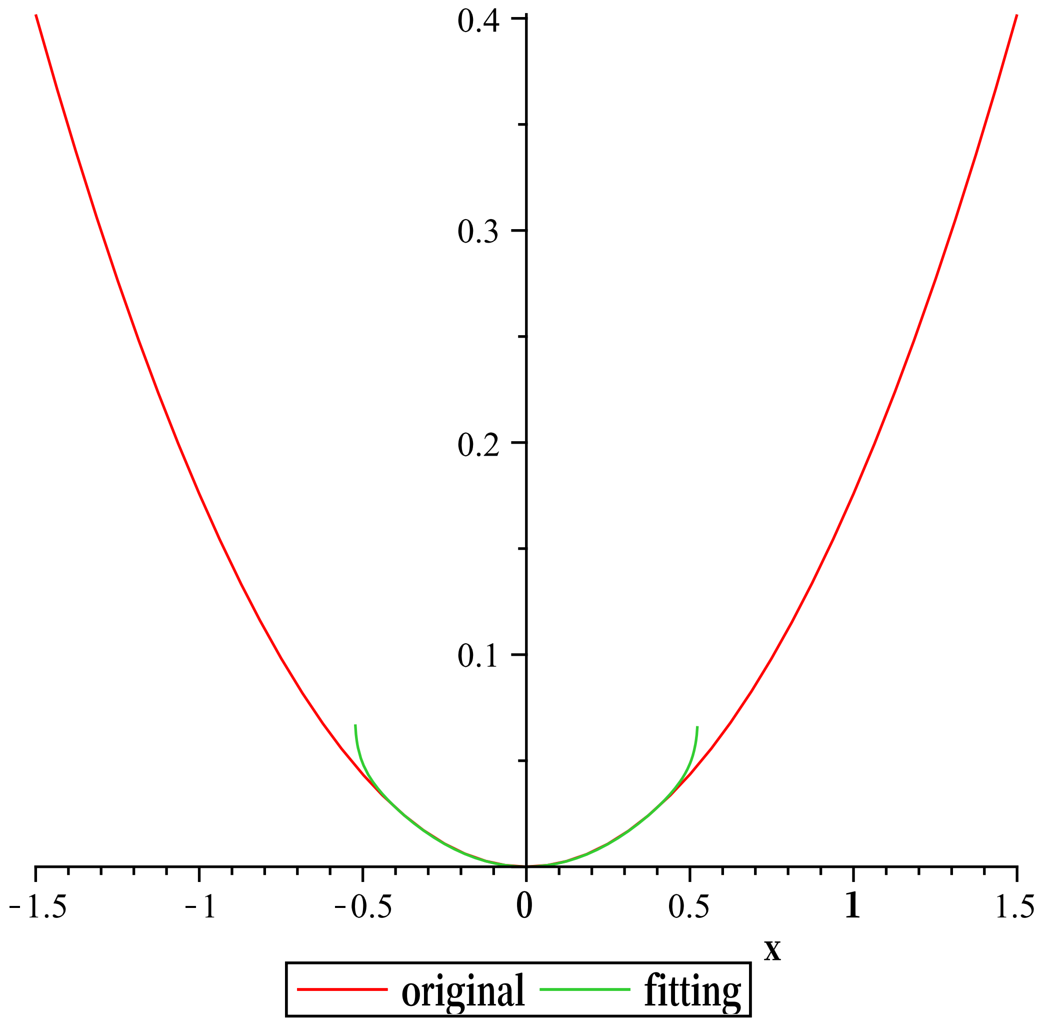
得られたパラメータ値を、model 式に代入してみます。

> fitModelEq := eval(model, fitParam)

$$fitModelEq := \frac{0.340873966931799055\,x^2}{1+\sqrt{1-3.648864964\,x^2}}-0.127761600656121576\,x^4 - 0.326047218510421178\,x^6-0.621422564795779953\,x^8 - 1.65736613208662064\,x^{10} \tag{2.6}$$

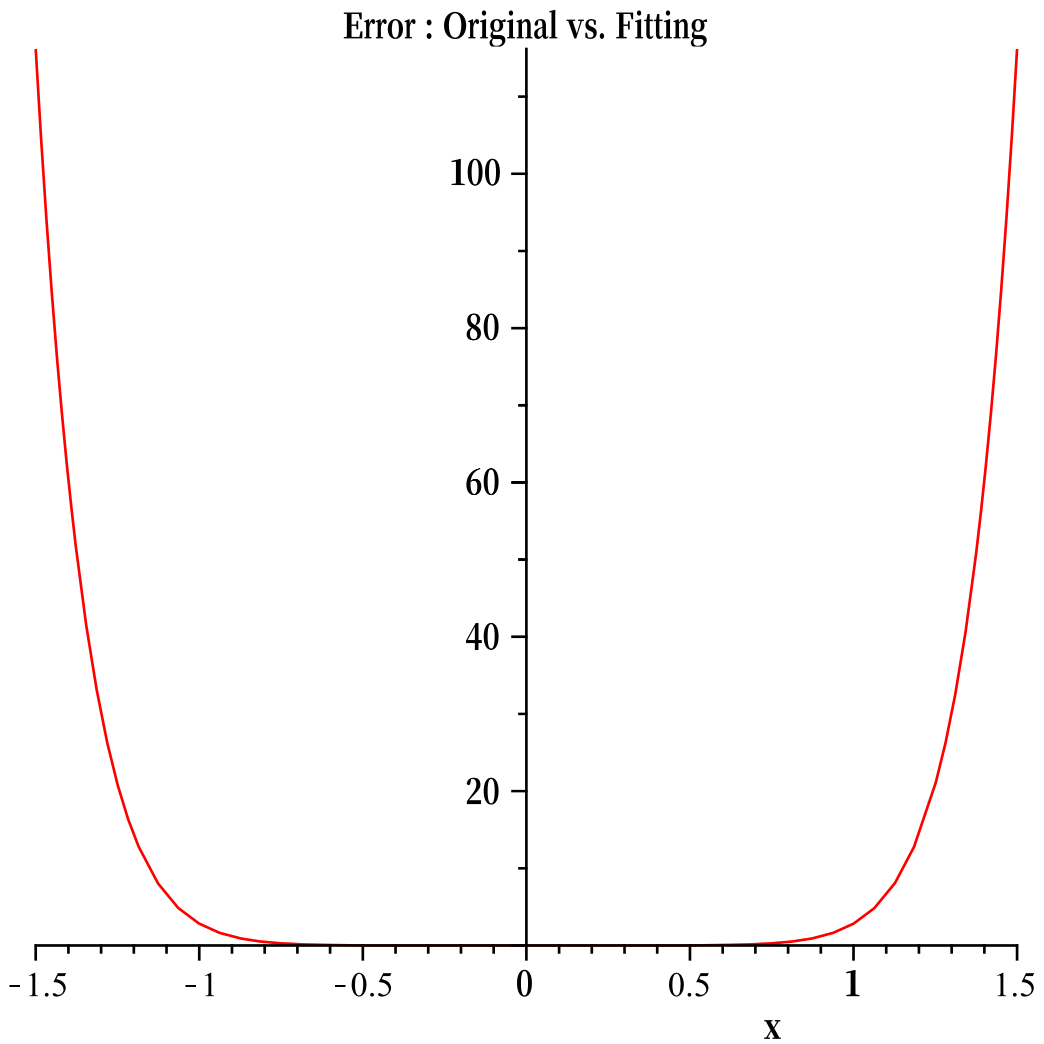
フィッティングの結果をグラフで確認してみます；

> *plot*([*modelEq*, *fitModelEq*], *x* =- 1.5 ..1.5, *legend* = ["original", "fitting"])



元のモデル式とフィッティングしたモデル式の誤差をプロットしてみます；

> *plot*(*abs*(*modelEq* − *fitModelEq*), *x* =- 1.5 ..1.5, *title* = "Error : Original vs. Fitting")



なお、あらかじめパラメータ値の一部（例：曲率、コーニック定数等）の取りうる範囲が判明しているような場合は、NonlinearFit コマンドの parameterranges オプションを指定してフィッティングを行うことも可能です。

> $NonlinearFit(fitModelApp, Xdata, Ydata, x, output = parametervalues, parameterranges = [c = 0.34 ..0.35, k = -2.0 .. -0.5])$

$$[A1 = 0.03264864963737771 50, A2 = -0.0426612354481841066, A3 = 0.0254741104515752612, A4 = -0.00508835576702243444, Err = 0.00497304277763629174, c = 0.340884481774707593, k = -1.99999850349951468] \tag{2.7}$$

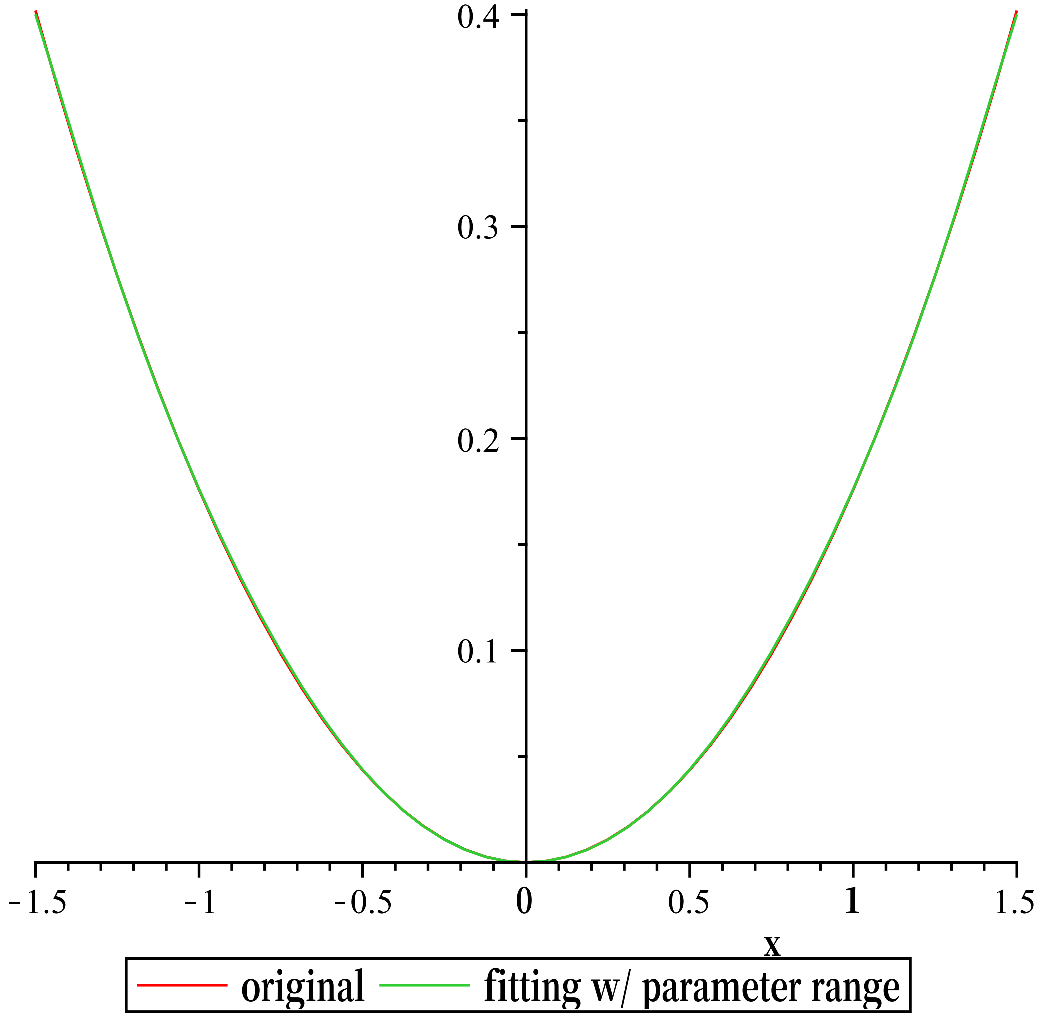
この結果で得られたモデルは次のようになります。

> $eval(model, \mathbf{(2.7)})$

$$\frac{0.340884481774707593\, x^2}{1 + \sqrt{1 + 0.1162020559\, x^2}} + 0.0326486496373777150\, x^4 - 0.0426612354481841066\, x^6 + 0.0254741104515752612\, x^8 - 0.00508835576702243444\, x^{10} \tag{2.8}$$

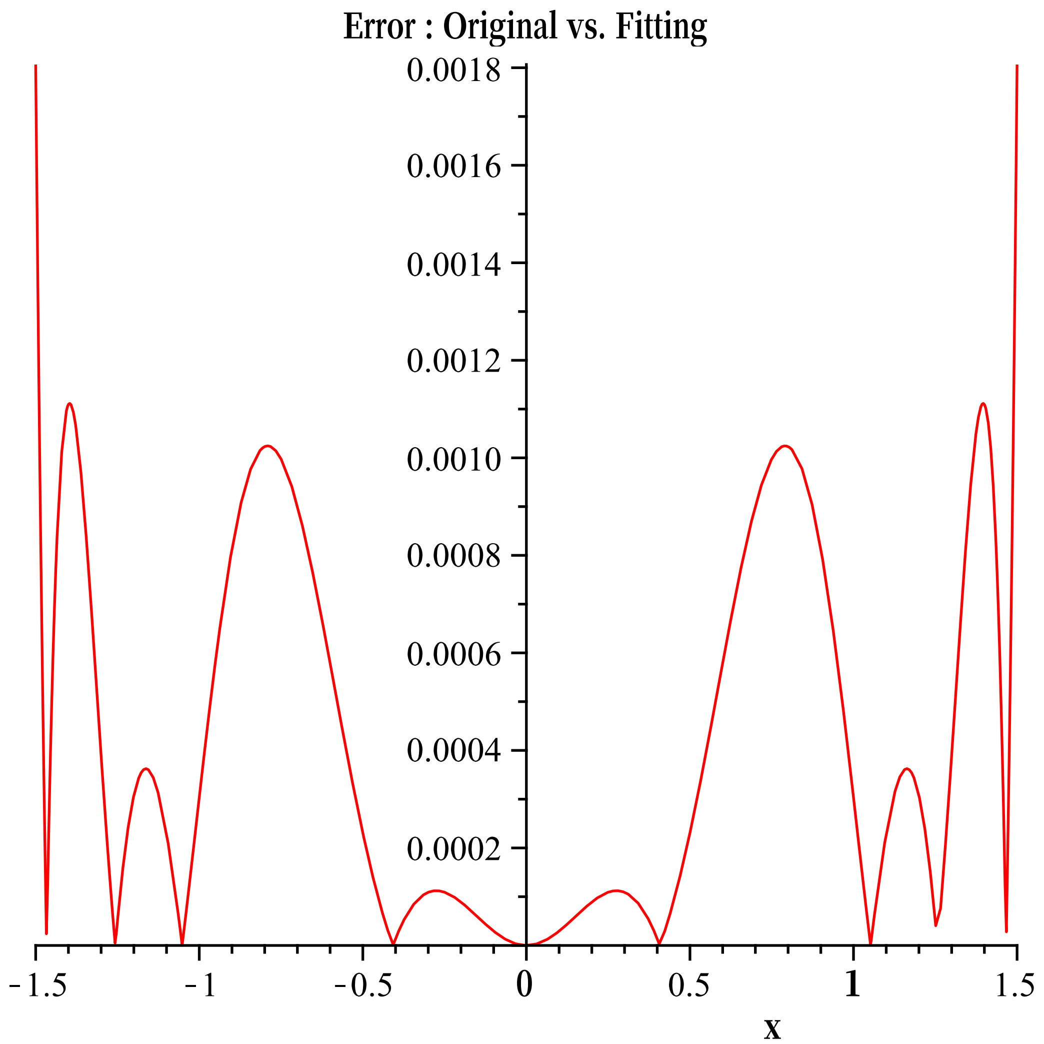
元のモデル式と比較してみます。

> $plot([modelEq, \mathbf{(2.8)}], x = -1.5 ..1.5, legend = [$"original", "fitting w/ parameter range"$])$



誤差についてもグラフで確認してみます。

> $plot(\mathrm{abs}(modelEq - \mathbf{(2.8)}), x = -1.5 ..1.5, title = $ "Error : Original vs. Fitting" $)$


Error : Original vs. Fitting
0.0018
0.0016
0.0014
0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
-1.5
-1
-0.5
0
0.5
1
1.5
x

>