Anaconda と Jupyter Notebook を用いた Python プログラミング（6）

※このドキュメントは相互参照型のハイパーテキストを意図しており，Web サイトおよびブックマーク，図表，数式，補足説明へのリンク（参照）を下線付き青文字で表し，（↩）の記号は初出の参照元へのリンクを表す。

この章で用いる Jupyter Notebook (iPython) のコードは，04ippf.ipynb にまとめた。

6．粉末X線回折装置の角度較正

6−1　装置の角度較正の概要

4 章と 5 章に示したように，Cu$K\alpha$ X線源を用いて Si 粉末の回折角 140° 以下の回折強度データを収集した場合には 111 反射から 533 反射まで 11 本の回折ピークが観測される。個々の回折ピークは隣接する回折ピークから 5° – 6° 程度以上離れた位置にある。

粉末試料を市販の粉末 XRD 測定装置に設置して，見かけ回折角 $\{2\Theta_j\}$ に対する回折強度 $\{Y_j\}$ の記録を得ることは容易である。しかし，このようにして得られる強度データの横軸値に対応する「見かけ回折角」 2Θ は，一般的には「真の回折角」 2θ からずれた値になる。このずれの原因は，4 章に示したように，装置に由来する要因と試料に由来する要因とからなる。

粉末 XRD 測定装置に由来する 2Θ 角度のずれは，原則的には装置に対する較正(こうせい)(calibration(キャリブレイション)) を施すことによってあらかじめ取り除いておかなければならない。

XRD 装置の角度較正のためには，米国国立標準技術研究所 NIST (National Institute of Standards and Technology) から頒布(はんぷ)される SRM640 シリーズ (SRM640, SRM640a, SRM640b, …, SRM640f) の標準 Si 粉末を用いるのが標準的な方法である。標準 Si 粉末について観測される回折ピーク位置が，NIST 保証書 (certificate(サーティフィケイト)) に記載された値 (Kaiser & Watters, 2010) からどのようにずれているかを調べれば良い。このためには「個別ピーク形状当て嵌(は)め」 (individual(インディヴィデュアル) peak(ピーク) profile(プロファイル) fitting(フィッティング) ; IPPF) と呼ばれる操作が必要となる。

本課題では，市販される粉末 XRD 装置の角度較正を目的とした個別ピーク形状当て嵌めについて，以下の手順をとる。

(1) 標準 Si について測定された XRD データに対して**逆畳込的処理** (deconvolutional treatment; DCT) を施し，**X 線源の分光強度形状** (spectroscopic profile of X-ray) と**軸発散収差** (axial-divergence aberration)，**赤道・透過性収差** (equatorial-transparency aberration)，またリガク社の販売した特定の装置 (Rigaku MiniFlex 600-C) の場合には**ミニフレ変形** (MiniFlex deformation) によるピークシフトと非対称なピーク形状の変形を除去する。この操作の手順は 4 章で示した。

(2) 逆畳込的処理後のデータから，予測されるピーク位置（保証書記載ピーク位置）の ±0.5° の範囲を，個々の回折ピークの強度図形データとして抽出・描画する。

(3) **曲線当て嵌め** (curve fitting) により，各ピークの位置を推定する。この際のピーク形状モデルとして，Lorentz 型函数モデルと Gauss 型函数モデルの畳込として定義される左右対称な形状を持つ Voigt 函数を用いる。

(4) NIST SRM640d 保証書に記載されたピーク位置 2θ (two theta) からの実測ピーク位置 2Θ (two Theta) のずれ $\Delta 2\Theta$ (Delta two Theta) （$\Delta 2\Theta \equiv 2\Theta - 2\theta$）について，ゴニオメータ角度 2Θ 依存性を確認する。

(5) 何らかのピーク・シフト・モデルを当てはめる。この課題では**定数・余弦・正弦項モデル** (constant-cosine-sine terms model) によって回折角ずれ $\Delta 2\Theta$ の角度 2Θ 依存性をモデル化する。定数・余弦・正弦項モデルでは，回折角ずれは

$$\Delta 2\Theta \approx \Delta 2\Theta_0 + \Delta 2\Theta_1 \cos\Theta + \Delta 2\Theta_2 \sin\Theta$$

と表される。

6-2　ピーク強度図形データの抽出

NIST SRM640d 標準 Si 粉末についての実測の粉末 XRD 図形と逆畳込的処理後の強度図形を確認する。実測 XRD 強度データのファイル名が “00raw.csv”，逆畳込的処理後の強度図形データのファイル名が “01dct.csv” であるとして，いずれもフォルダ「… / 20250702MiniFlex / 20250702-011 / 」に配置されているとする。4 章で見たように，全体の XRD パターンは Figure 6.2.1 のようになる。

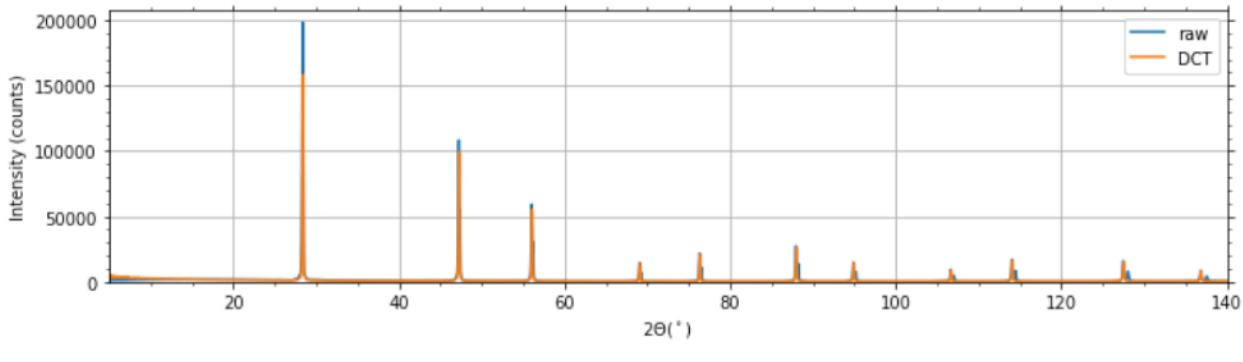

Figure 6.2.1　NIST SRM 640b 実測データ (raw) と逆畳込的処理後データ (DCT) の比較（↩）

Si 111, 220, 311, 400, 331, 422, 511/333, 440, 531, 620, 533 反射に相当する 11 本の反射ピークが出現していること，逆畳込的処理後のデータ (DCT) では，主ピークの主な強度はピーク位置 ±1° 程度の範囲に収まっており，隣接するピークの強度の寄与は無視しうることを確認する。

次節で個別ピーク形状当てはめ分析 (individual peak profile fitting) によりピーク位置を求める作業を行う。逆畳込的処理後の強度データ (DCT) からは，既知の回折ピーク位置の ±0.5° の範囲の強度データのみを抽出すれば，比較的単純な手順でピーク位置が求められる。

6－3　個別ピーク形状当てはめ分析

最小自乗法 (least-squares method)（最小二乗法）による曲線当て嵌(は)め分析 (curve fitting analysis) は，粉末 XRD データを解析する目的で頻繁(ひんぱん)に用いられる。結晶構造推定や定量相組成分析を目的などとして利用されるリートベルト解析 (Rietveld, 1969) は，そのうちの典型的な例の一つである。ここでは個々の回折ピーク形状にピーク形状モデル関数を当てはめる個別ピーク形状当てはめ (individual peak profile fitting; IPPF) を行う。

最小自乗法は，データの統計誤差が既知である場合の最尤(さいゆう)推定 (maximum likelihood estimation) とみなすこともできる。（補足 6.3.A）

6－1 節に示した (2)–(5) の操作を行うための Jupyter Notebook (iPython) コードを “04ippf.ipynb” として準備した。このコードを利用するためには，新しい Notebook を作成するか既存の Notebook にコードセルを追加して実行すれば良い。コードの冒頭 “%config InlineBackend.figure_formats = ['svg']” の記述は，スクリーン上で図を SVG 形式のベクター画像として表示することを意味するものであり重要な意味はない。

ノートブック “04ippf.ipynb” では，作業用ディレクトリ “20260702-011” に逆畳込的処理前の強度データのファイルが “00raw.csv” という名称で存在し，逆畳込的処理後の強度データのファイルが “01dct.csv” という名称で存在することを前提とする。個別ピーク形状当て嵌めの手法は，逆畳込的処理後の “01dct.csv” のみに適用するが，逆畳込的処理の影響がどのように現れたのか見やすくするため，処理前のデータ “00raw.csv” の示す強度図形も同時に表示する。

Python コード “ippf2.5/ippf2.py” では，scipy.optimize.minimize() メソッドを利用して，最適化を施す。（補足 6.3.B）

ピーク形状モデル函数としては，以下の式で定義されるような函数 $f_{\mathrm{fit}}(x; b, I, x_0, \sigma, w)$ を用いる。

$$f_{\mathrm{fit}}(x; b, I, x_0, \sigma, w) = b + I\, f_{\mathrm{Voigt}}(x - x_0; \sigma, w) \tag{6.1.1}$$

ここで b は定数背景強度 (constant background)，I はピーク積分強度 (integrated intensity)，x_0 はピーク位置である。Voigt（フォークト）函数 (Voigt function) $f_{\mathrm{Voigt}}(x; \sigma, w)$ は以下のように定義される。

$$f_{\mathrm{Voigt}}(x; \sigma, w) \equiv f_{\mathrm{Gauss}}(x; \sigma) * f_{\mathrm{Lorentz}}(x; w) \tag{6.1.2}$$

$$f_{\mathrm{Gauss}}(x; \sigma) \equiv \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{6.1.3}$$

$$f_{\mathrm{Lorentz}}(x; w) \equiv \frac{1}{\pi w}\left(1 + \frac{x^2}{w^2}\right)^{-1} \tag{6.1.4}$$

函数 $f_{\mathrm{Gauss}}(x; \sigma)$ は，平均 0，標準偏差 σ の「正規分布」の確率密度函数を表す。

ノートブック “04ippf.ipynb” から (Code 6.3.1) のようなコードを実行すれば 個々のピーク形状パラメータについて，(Output 6.3.1) のようなテキスト出力と Figure 6.3.1 のようなグラフが得られる。また Figure 6.3.1 には NIST の保証書に記載されたピークの位置を垂直線で示している。

(Output 6.3.1) 中の “Background” は定数（角度に依存しない）背景強度 b，”Integrated intensity” はピークの積分強度 I，“Peak location” はピーク位置 $2\Theta_0$，“Lorentzian HWHM” はローレンツ型函数（かんすう）成分 $f_{\mathrm{Lorentz}}(x; w)$ の半値半幅（はんちはんはば）(half width at half maximum; HWHM) w，“Gaussian sigma” はガウス型函数 $f_{\mathrm{Gauss}}(x; \sigma)$ の標準偏差 σ を表す。（補足 6.3.C）

(Code 6.3.1) Si の反射ピークデータ（実測値・逆畳込的処理後）の抽出と描画，逆畳込的処理後データへの scipy.optimize.minimize() メソッドによる曲線当て嵌め（最適化）のための Notebook コード（↩）

```
#
%run ../../ippf2.5/ippf2.py 00raw.csv 01dct.csv 02peaks_ref_640d_Si.csv
03peaks_opt.csv ippf.cfg 'NIST SRM640d'
# 生データ 00raw.csv は強度図形を表示するためだけに使う。
```

(↩)

(Output 6.3.1) Si 111 反射ピークデータ（実測値・逆畳込的処理後）の抽出と描画，逆畳込的処理後データへの scipy.optimize.minimize() メソッドによる曲線当て嵌め（最適化）のテキスト出力（↩）

```
*****
Optimization for 111-reflection, minimize(Nelder-Mead) method
Initial Guess...
 Background : 1222.2
 Integrated intensity : 12040.1
 Peak location : 28.4410
 Gaussian sigma : 0.1700
 Lorentzian HWHM : 0.0051
 ----
 chi**2 = 1702863.1
 ========================
 algorithm =  minimize(Nelder-Mead)
 ========================
 Optimization succeeded.
  Optimized ...
  Background : 957  ± 8
  Integrated intensity : 12589  ± 15
  Peak location : 28.4678  ± 0.0001
  Gaussian sigma : 0.0347  ± 0.0001
  Lorentzian HWHM : 0.0234  ± 0.0002
 chi**2 = 9202.2
 Times of function calls =  707

......
```

(↩)

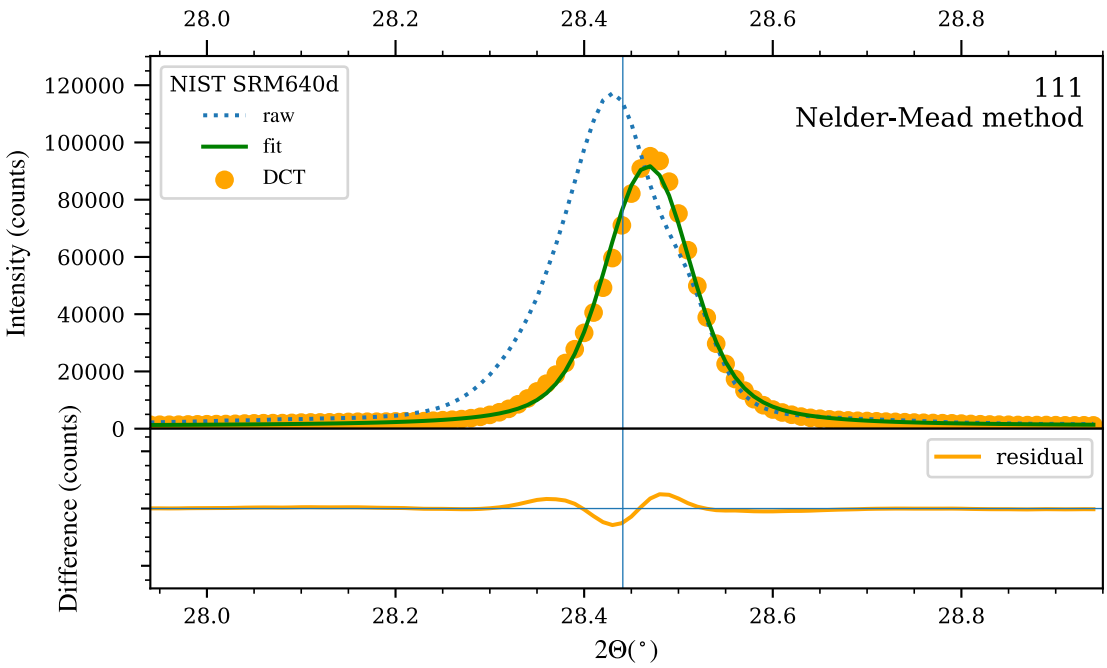

Figure 6.3.1 NIST SRM 640d Si, 111-反射の実測データ (raw) と逆畳込的処理後データ (DCT)，DCT データに対して Voigt 函数を Nelder-Mead アルゴリズムによってフィッティングした結果 (fit) の比較。垂直線は NIST SRM640d 保証書に記載されたピーク位置を示す。下側のパネルに当て嵌め残差 (residual) 曲線を示す（↩）。

Figure 6.3.1 に示した逆畳込的処理後の Si 111 反射ピークの強度曲線 (DCT) は，概ね左右対称な形状になるが，ピークの位置は逆畳込的処理前 (raw) のデータより NIST 保証書記載値から遠くなっている。

リートベルト解析を含むピーク形状当て嵌め分析 (peak profile fitting analysis) を用いる場合には，Figure 6.3.1 のように「実測値と当て嵌め曲線との差」のプロット (difference plot) を同時に示すべきとされる。

実測値と理論的な計算により予測された値との間に差・違いがある場合には，「仮説・理論が間違っている」ことを前提とした検討をすべきである。

差プロットの形状は，理論的な予測のどこに問題があるか，どのように修正すれば解決しうるかについての情報を含む。

Notebook コード “04ippf.ipynb” では，Si の 111 反射から 533 反射まで観測可能な 11 の回折ピークのすべてを処理対象とする。

Si 533 反射に関するピーク形状当て嵌め解析の出力の部分を (Output 6.3.2) に示す。

(Output 6.3.2) Si 533 反射ピークデータ（実測値・逆畳込的処理後）の抽出と描画，逆畳込的処理後データへの `scipy.optimize.minimize()` メソッドによる曲線当て嵌め（最適化）についてのテキスト出力（↩）

```
*****
Optimization for 533-reflection, minimize(Nelder-Mead) method
```

```
Initial Guess...
 Background : 300
 Integrated intensity : 1067
 Peak location : 136.8800
 Gaussian sigma : 0.1700
 Lorentzian HWHM : 0.0508
 ----
 chi**2 =  34526.68449437161
=====
Optimization succeeded.
Optimized ...
 Background : 247 ± 2
 Integrated intensity : 1237 ± 5
 Peak location : 136.8249 ± 0.0004
 Gaussian sigma : 0.0068 ± 0.0030
 Lorentzian HWHM : 0.0819 ± 0.0365
 chi**2 =  106.250169028452
 Times of function calls =  1001
```

6-4　定数・余弦・正弦項シフトモデル

6 – 3 節に示した個別ピーク形状分析により最適化されたピーク位置の NIST SRM640d 保証書記載値からのずれ Δ2Θ の回折角依存性に対して以下の式 (6.4.1) で表される定数・余弦・正弦項モデル (constant-cosine-sine terms shift model) を当て嵌める。

$$\Delta 2\Theta = \Delta 2\Theta_0 + \Delta 2\Theta_1 \cos\Theta + \Delta 2\Theta_2 \sin\Theta \tag{6.4.1}$$

このモデルに物理的・数学的に明確な根拠はない。Figure 6.4.1 に見られるように，見た目が直線的な変化なので直線を当てはめても良いかとも思われる。

ノートブックでコード (Code 6.4.1) を実行し，シフト（角度のずれ）の角度依存性をモデル化するための Python コード ../../ippf2.5/model_shift5.py を起動する。

解析の結果を (Output 6.4.1) と Figure 6.4.1 に示す。

(Output 6.4.1) テキスト中の “Δ2Θ_0” は定数項，“Δ2Θ_1” は余弦項係数，“Δ2Θ_2” は正弦項係数を表す。

(Code 6.4.1) Si の 11 反射ピークの回折角依存性に対して定数・余弦・正弦項ピークシフトモデルによる曲線当て嵌め解析をするための Notebook コード（↩）

```
#
# "model_shift5.py の第１引数（ひきすう）は最適化されたピーク位置のリスト
#  （入力ファイル）（デフォルト：03peaks_opt.csv）
```

```
# 第２引数は当てはめの結果（出力ファイル）（デフォルト：04peaks_fit.csv）
# 第３引数は当てはめモデルの名称（デフォルト：const_cos_sin）
# 第４引数は設定ファイルの名称（デフォルト：model_shift.cfg）
# 第５引数は仮定する角度誤差 (°)（デフォルト：0.005）
# 第６引数は凡例のタイトル（デフォルト：SRM640d Si）
# 第７引数は試料位置ずれ (mm)（デフォルト：0.0）
# 第８引数は試料のＸ線侵入深さ (mm)（デフォルト：0.22）
# 第９引数は格子定数 a (Å)（デフォルト：5.43123）
# 第10引数は格子定数 c (Å)（デフォルト：5.43123）
%run ../../ippf2.5/model_shift5.py 03peaks_opt.csv 04peaks_fit.csv
const_cos_sin model_shift.cfg 0.005 'SRM640d Si' 0.0 0.218 5.43123 5.43123
# 上のコマンドラインでは全ての引数を明示したが,
# %run ../../ippf2.5/model_shift5.py
# としても同じ動作をする。
```

(↩)

(Output 6.4.1) Si の 11 反射ピークの回折角依存性に対する定数・余弦・正弦項ピークシフトモデルによる曲線当て嵌め解析結果の出力（↩）

```
**********************************
 Source Python code file name :  ../../ippf2.5/model_shift5.py
 Optimized peak list file name :  03peaks_opt.csv
 Fit peak list file name :  04peaks_fit.csv
 Peak shift model :  const_cos_sin
 Configuration file name :  model_shift.cfg
 Title of legend :  SRM640d Si
 Specimen displacement (mm) :  0.0
 Penetration depth (mm) :  0.201
 Lattice constant a (Å) :  5.43123
 Lattice constant c (Å) :  5.43123
 Assumed error (º) :  0.005
**********************************
 Relative height of lower panel:  0.5
 Relative height of upper panel:  0.5

Model for Peak Shift :  const_cos_sin
*****
Initial Guess...
Optimization for peak shift
 by a constant, cosine, and sine terms model
 Δ2θ_0 (º) : 0.0100
 Δ2θ_1 (º) : 0.0100
 Δ2θ_2 (º) : 0.0001
 chi**2 = 2545.49
```

```
*****
Optimization starts...
Optimization finished.
 Δ2θ_0 = -0.05158º ± 0.03724º
 Δ2θ_1 = 0.11520º ± 0.02902º
 Δ2θ_2 = -0.11440º ± 0.02626º
 chi**2 = 4.97

Data saved to  04peaks_fit.csv
Data Frame:
```

	h	k	l	ttRef	Delta2T	Err	Fit	Residual	I
0	1	1	1	28.441	0.027	0.005	0.032	-0.005	100.000
1	2	2	0	47.300	0.011	0.005	0.008	0.003	60.090
2	3	1	1	56.119	0.001	0.005	-0.004	0.005	34.029
3	4	0	0	69.126	-0.019	0.005	-0.022	0.003	8.836
4	3	3	1	76.371	-0.030	0.005	-0.032	0.002	13.622
5	4	2	2	88.024	-0.050	0.005	-0.048	-0.002	18.889
6	5	1	1	94.946	-0.063	0.005	-0.058	-0.005	10.815
7	4	4	0	106.700	-0.077	0.005	-0.075	-0.002	6.969
8	5	3	1	114.082	-0.085	0.005	-0.085	-0.000	14.275
9	6	2	0	127.532	-0.105	0.005	-0.103	-0.002	15.631
10	5	3	3	136.877	-0.111	0.005	-0.116	0.005	8.731 (↩)

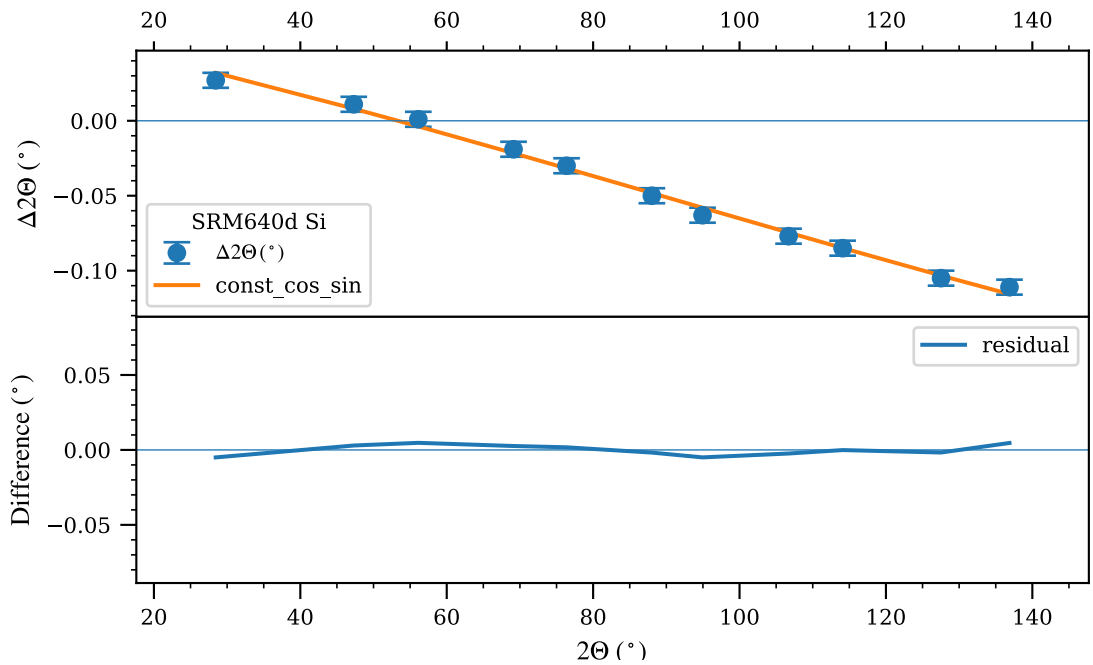

Figure 6.4.1 NIST SRM 640d Si, 逆畳込的処理後データのピーク位置ずれ Δ2Θの回折角依存性に対する定数・余弦・正弦項ピーク・シフト・モデルの当て嵌め解析結果（↩）

Figure 6.4.1 に示す（直線的な）曲線を「較正曲線」(calibration curve) と呼んでも良い。全角度範囲にわたってピーク位置ずれ $\Delta 2\Theta$ が同一の較正曲線に載るとみなせれば，それで「装置の角度較正ができた」ことになる。

一方で，粉末回折装置 (Rigaku MiniFlex 600-C) に付属する計測制御ユーザー・インターフェス (SmartLab Studio II) に付属する「リアルタイム角度補正」機能では，装置の角度補正ができていることにならないことは前述の通りである。

(Output 6.4.1) に示されるように，定数項 $\Delta 2\Theta_0 = -0.052(37)^\circ$，余弦項係数 $\Delta 2\Theta_1 = +0.115(29)^\circ$，正弦項係数 $\Delta 2\Theta_2 = -0.114(26)^\circ$ としてモデル化された。ただしカイ自乗が $\chi^2 = 4.97$ と標本点数 (11) と比較して極端に小さい値になっており，過学習 (over-fitting) の傾向を示している。このことは，仮定した角度の統計誤差 0.005° が過大に評価されていたことを示唆する。

仮に角度の統計誤差を 0.002° と仮定すれば，定数項 $\Delta 2\Theta_0 = -0.0516(15)^\circ$，余弦項係数 $\Delta 2\Theta_1 = +0.115(11)^\circ$，正弦項係数 $\Delta 2\Theta_2 = -0.114(11)^\circ$ としてモデル化され，カイ自乗の値は $\chi^2 = 31.0$ と標本点数 (11) よりやや大きい値になり，過学習 (over-fitting) の傾向は示さない。

しかし，Figure 6.4.1 に見られるようにフィッティング残差は回折角 2Θ に対して，やや系統的に変化しているようにも見え，仮定した定数・余弦・正弦項モデルの不備による可能性がある。一方で，このモデルによって，測定装置の機械誤差として許容しうる 0.01° 程度以内に偏差が抑えられると予想されることから，このモデルの正当性を否定することも困難である。

この装置 (Rigaku MiniFlex 600-C / SmartLab Studio II) に備えられる「リアルタイム角度補正」機能を無効化したときに，装置製造会社が「回折角」を意図して出力するらしい値を $2\Theta_{\mathrm{Rigaku}}$ とする。半導体ストリップ型X線検出器を用いる場合，ゴニオメータ角 $2\Theta_{\mathrm{G}}$ と検出ストリップの有効オフセット角 $2\Psi_{\mathrm{eff}}$ との差 $2\Theta = 2\Theta_{\mathrm{G}} - 2\Psi_{\mathrm{eff}}$ として表される「見かけ回折角」 2Θ は

$$\begin{aligned} 2\Theta &= 2\Theta_{\mathrm{Rigaku}} - \Delta 2\Theta_0 - \Delta 2\Theta_1 \cos\Theta - \Delta 2\Theta_2 \sin\Theta \\ &= 2\Theta_{\mathrm{Rigaku}} + 0.052^\circ - 0.115^\circ \cos\Theta_{\mathrm{Rigaku}} + 0.114^\circ \sin\Theta_{\mathrm{Rigaku}} \end{aligned} \tag{6.4.2}$$

と表されると仮定する。

6−5　角度補正の考え方

6 – 4 節に示したように，角度補正式として以下の式：

$$\Delta 2\Theta = \Delta 2\Theta_0 + \Delta 2\Theta_1 \cos\Theta + \Delta 2\Theta_2 \sin\Theta \tag{6.5.1}$$

で表される定数・余弦・正弦項モデルを用いたとき，補正式の含むパラメータは，$\Delta 2\Theta_0 = -0.052^\circ$，$\Delta 2\Theta_1 = +0.115^\circ$，$\Delta 2\Theta_2 = -0.114^\circ$ として最適化された。

実測のデータが角度値 $2\Theta_j$ と強度値 Y_j の組として $(2\Theta_1, Y_1), \cdots, (2\Theta_n, Y_n)$ のように表される場合，角度値を $2\Theta_j \to 2\Theta'_j = 2\Theta_j - \Delta 2\Theta_j$ と読み替えれば角度補正ができたことになる。

しかし，一般的には角度値 $2\Theta_j$ は等間隔に取られることが多いのに対して，補正後の角度 $2\Theta'_j$ は等間隔にならない。一般的なリートベルト解析ソフトウェアを用いるのであれば問題にならないが，逆畳込的処理のためのコードも含めて等間隔標本点のデータの方が扱いやすい場合も多い。

角度補正のための Python コード ippf2.5/correct_shift3.py では，補間（内挿）(interpolation) のための NumPy ライブラリ中のメソッド numpy.interp() を用いて $(2\Theta'_j, Y_j) \to (2\Theta_j, Y'_j)$ となるよう擬似的に等間隔な標本点に対する強度値を得ている。

データの端点で $2\Theta_1 < 2\Theta'_1$，$2\Theta'_n < 2\Theta_n$ などのようになった場合には補間ではなく補外 (extrapolation) が必要になるが，numpy.interp() メソッドではそのような場合には強制的に $Y'_1 \leftarrow Y_1$，$Y'_n \leftarrow Y_n$ のように端点の値を拡張する処置が施され，結果的には大きな問題は起こらない。

SciPy ライブラリには，NumPy ライブラリの numpy.interp() メソッドより高機能な scipy.interpolate.interp1d() メソッドも含まれるが，指定できるオプションが多く，不慣れなユーザには扱いにくい面もあるかと思われる。

6-6　角度補正の効果の確認

本節に示す内容は，本来は不要な作業とも言えるが，逆畳込的処理とミニフレ補整を含めた適応的逆畳込的処理 (adaptive deconvolutional treatment; aDCT)，個別ピーク形状当てはめ解析 (IPPF)，定数・余弦・正弦項シフト・モデルを適用した角度補正が合理的に進行するかを確認する意味がある。SRM 640d Si のデータに角度補正を施してから (1) 逆畳込的処理，(2) 個別ピーク形状分析を行い，結果を確認する。

角度補正を適用するための Python コード correct_shift3.py を利用するには， Jupyter Notebook から (Code 6.6.1) のようなコードを実行すれば良い。

```
(Code 6.6.1) 角度補正を実行するための Notebook のコード（↩）
# 装置（ミニフレ）の出力するデータを 00raw.csv として保存したが,
# 角度補正を施したデータのファイル名を 01raw.csv とすることに注意する。
%run ../../ippf2.5/correct_shift3.py 00raw.csv 01raw.csv -0.0516 +0.1152
-0.1144 +0.0 +0.0
 (↩)
```

(Code 6.6.2) のように角度補正後のデータ "01raw.csv" に対して逆畳込的処理を施した結果を "02dct.csv" として保存する。

(Code 6.6.2) 逆畳込的処理を実行するための Notebook のコード (↩)

```
%%time
%run ../../exterm7.3/exterm7.py 01raw.csv 02dct.csv dct.cfg
# 逆畳込的処理 (exterm7.3)
# 入力ファイル名は 01raw.csv, 出力ファイル名は 02dct.csv
```
(↩)

(Code 6.6.3) のようなコードを実行し，畳込的処理前後の回折強度図形をグラフ表示して変化を確認する。

(Code 6.6.3) 複数ファイルの強度データをグラフにする Notebook のコード (↩)

```
#
# 角度補正を施したデータ 01raw.csv
# 角度補正後に逆畳込的処理・ミニフレ補整を施したデータ 02dct.csv を表示する。
#
import numpy as np
import matplotlib.pyplot as plt
source0 = np.loadtxt('01raw.csv',delimiter=',')
x,y0 = source0[:,0],source0[:,1]
source1 = np.loadtxt('02dct.csv',delimiter=',')
x,y1 = source1[:,0],source1[:,1]
fig,ax = plt.subplots(figsize=(6,3))
ax.axvline(28.441,ls='--',color='gray') # NIST SRM640d 保証書記載 Si 111 ピーク位置
ax.plot(x,y0,label='Raw data')
ax.plot(x,y1,label='adaptive DCT data')
ax.xaxis.set_ticks_position('both')
ax.yaxis.set_ticks_position('both')
ax.set_xlim(27.941,28.941)
ax.set_ylim(0,)
ax.minorticks_on()
ax.set_xlabel(r'$2\Theta (^\circ)$')
ax.set_ylabel(r'${\rm Relative\ intensity,}\ s_0 $')
ax.legend()
plt.show()
```
(↩)

Figure 6.6.1 に逆畳込的処理前後の Si 111-反射強度図形の変化を示す。

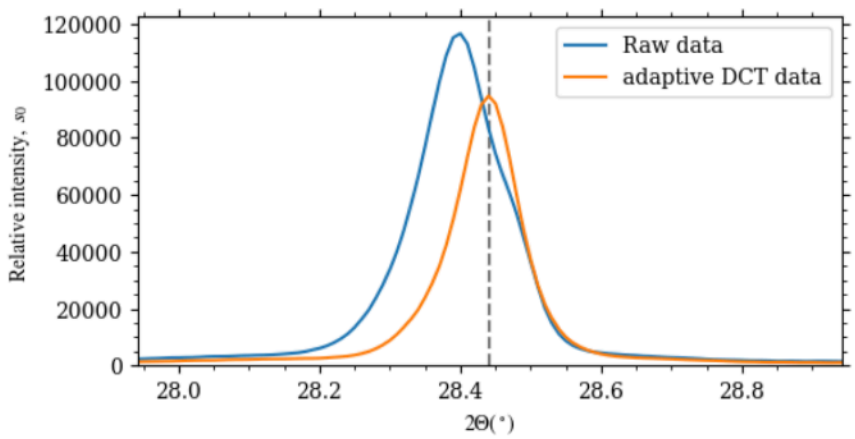

Figure 6.6.1 角度補正・畳込的処理前後の NIST SRM640d Si の 111-反射ピーク図形の変化。灰色縦破線は NIST の保証書に記載されたピーク位置を示す。 (↩)

NIST SRM 640d 標準 Si 粉末の回折ピーク位置に合わせて角度補正をしたので当然だが，回折ピークは単峰性の (unimodal) 概ね左右対称な形状になり，ピーク位置は NIST 保証書記載値 (28.441°) と一致する。しかし逆畳込的処理後のピーク形状は，やや低角側に長い裾を引く非対称性を示している。

次に (Code 6.6.4) のように角度補正・逆畳込的処理後の強度データに対して個別ピーク形状当てはめ (IPPF) 分析を行う。角度補正後データのファイル名を "01raw.csv"，逆畳込的処理後のファイル名を "02dct.csv"，最適化 (optimize) されたパラメータのリストを出力するファイル名を "04peaks_opt.csv"，グラフ中の凡例タイトルを 'NIST SRM640d' とすることに注意する。

(Code 6.6.4) 個別ピーク形状分析を実行するための Notebook のコード (↩)

```
#
# 個別ピーク形状当てはめ分析
#
%run ../../ippf2.5/ippf2.py 01raw.csv 02dct.csv 02peaks_ref_640d_Si.csv
04peaks_opt.csv ippf.cfg 'NIST SRM640d'
```

(↩)

Si 111-反射についての個別ピーク形状当てはめ分析のコンソール出力の一部を (Output 6.6.1) に，グラフを Figure 6.6.2 に示す。

(Output 6.6.1) 個別ピーク形状分析の結果の一部 (↩)

```
*****
Optimization for 111-reflection, minimize(Nelder-Mead) method
Initial Guess...
 Background : 1277.3
 Integrated intensity : 11990.4
 Peak location : 28.4410
 Gaussian sigma : 0.1700
 Lorentzian HWHM : 0.0051
```

```
----
chi**2 = 1684960.7
=========================
algorithm =  minimize(Nelder-Mead)
=========================
Optimization succeeded.
 Optimized ...
 Background : 913  ± 8
 Integrated intensity : 12649  ± 15
 Peak location : 28.4359  ± 0.0001
 Gaussian sigma : 0.0348  ± 0.0001
 Lorentzian HWHM : 0.0238  ± 0.0002
chi**2 = 9128.0
Times of function calls =  731
(↩)
```

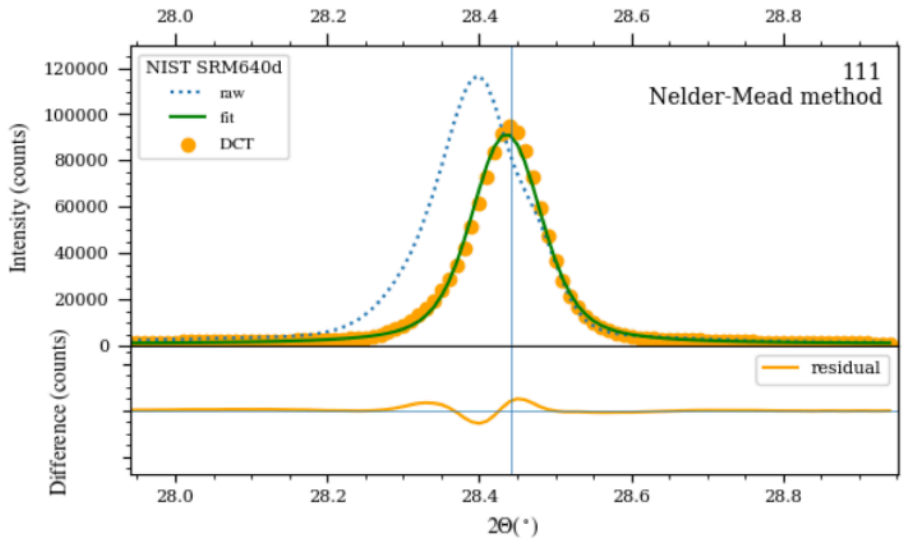

Figure 6.6.2　角度補正適用後の強度図形 (raw) と逆畳込的処理後の図形 (DCT)，111-反射ピーク付近の DCT 図形に対してフォークト関数を当てはめた図形 (fit) とフィッティング残差 (residual)。青縦線は NIST SRM640d の保証書に記載されたピーク位置を示す。（↩）

補足

（補足 6.1.A）試料位置ずれの効果（↩）

Bragg-Brentano（ブラッグ ブレンターノ）型の粉末X線回折測定装置を用いる場合，試料粉末表面の高さが，ゴニオメーターの回転軸となるべく厳密に一致するように調整しなければならない。ゴニオメーター軸と精度良く一致するように調整されているはずの基準面と，試料板との間に，粉末や塵埃（じんあい）などを挟（はさ）み込んでしまい，試料板の角度や位置の狂うことのないように注意する。

Figure 6.1.A.1 に示すように，Bragg-Brentano（ブラッグ ブレンターノ）型の回折計について，試料の面（つら）がゴニオメーター軸からずれている場合に，どれだけ回折ピーク位置がずれるかは，ゴニオメータ半径 R と試料の位置ずれ ΔS によって決

まる。試料面の上方向に試料の位置がずれた場合に ΔS が正の値をとるとして，試料位置ずれ効果 (specimen displacement effect) によるピークの位置のずれ $\Delta 2\Theta_{\mathrm{SD}}$は

$$\Delta 2\Theta_{\mathrm{SD}} = \frac{2\Delta S \cos\Theta}{R} \tag{6.1.A.1}$$

と表される。ただし式 (6.1.A.1) はラジアン単位での表現であり，度 (degree) 単位に変更するためには $180°/\pi$ をかける。回折ピーク位置のずれは $\cos\Theta$ に比例して変化することに注意する。

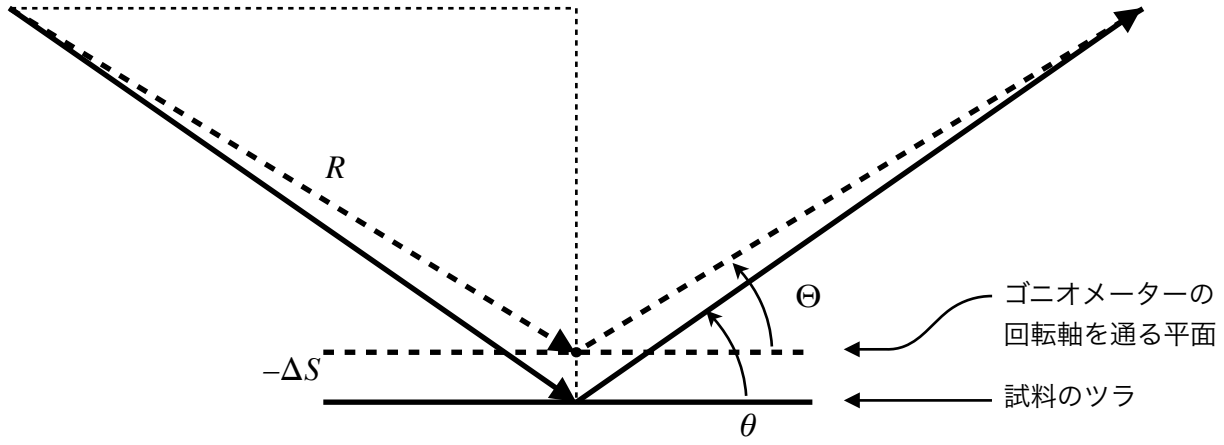

Figure 6.1.A.1　試料位置ずれの効果 。ゴニオメーターの角度を 2Θ（Θ はギリシャ大文字のシータ），回折角を 2θ（θ はギリシャ小文字のシータ）で表す。試料のツラが凹んでいると，本来の回折角 2θ より浅い（低い）ゴニオメーター角度 2Θ で回折ピークが出現し，回折ピーク位置が低角側にずれる。（↩）

ゴニオメーター半径 R，ゴニオメーター角 2Θ，試料位置ずれ ΔS（高すぎの場合プラス，低すぎの場合マイナスの数値），ずれた位置での回折角 $2\theta \equiv 2\Theta + \Delta 2\theta$ とすれば，Figure 6.1.A.1 から，一般的に

$$\begin{aligned} & R\cos\Theta\tan\theta = R\sin\Theta - \Delta S \\ & \Rightarrow R\cos\Theta\tan(\Theta+\Delta\theta) = R\sin\Theta - \Delta S \end{aligned} \tag{6.1.A.1}$$

の関係が成立する。回折角のずれ $\Delta 2\theta$ が充分に小さい値であれば，三角関数の加法定理と $\lim_{x\to 0}\tan x = x$ の関係から，

$$\tan(\Theta+\Delta\theta) = \frac{\tan\Theta + \tan\Delta\theta}{1-\tan\Theta\tan\Delta\theta} \simeq \frac{\tan\Theta + \Delta\theta}{1-\Delta\theta\tan\Theta} \tag{6.1.A.2}$$

だから，

$$\begin{aligned} & \frac{R\cos\Theta\,(\tan\Theta+\Delta\theta)}{1-\Delta\theta\tan\Theta} \simeq R\sin\Theta - \Delta S \\ & \Rightarrow R\sin\Theta + R\Delta\theta\cos\Theta \simeq R\sin\Theta - R\Delta\theta\tan\Theta\sin\Theta - \Delta S + \Delta S\Delta\theta\tan\Theta \\ & \Rightarrow R\Delta\theta\cos\Theta \simeq -R\Delta\theta\tan\Theta\sin\Theta - \Delta S + \Delta S\Delta\theta\tan\Theta \\ & \Rightarrow (R\cos\Theta + R\sin\Theta\tan\Theta)\Delta\theta \simeq -\Delta S \\ & \Rightarrow \frac{R(\cos^2\Theta+\sin^2\Theta)\Delta\theta}{\cos\Theta} \simeq -\Delta S \\ & \Rightarrow \Delta\theta \simeq -\frac{\Delta S\cos\Theta}{R} \\ & \Rightarrow \Delta 2\theta \simeq -\frac{2\Delta S\cos\Theta}{R} \end{aligned} \tag{6.1.A.3}$$

となる。マクローリン展開 (Maclaurin expansion) を使って，ほぼ機械的な手順で同じ関係式を導くこともできる（補足 6.1.A.1）

試料粉末の面(つら)が高過ぎれば，回折角 2θ はゴニオメーター角 2Θ より浅い角度になり，試料粉末の面(つら)が低過ぎれば，回折角 2θ はゴニオメーター角 2Θ より深い角度になる。

粉末回折ピークは，回折角 2θ がブラッグ条件を満たす時に現れる。**真の回折角** 2θ と装置の出力する**見かけ回折角** 2Θ とは意味が違うこと，粉末回折データは，**見かけ回折角** 2Θ に対して回折強度を記録するものであることに注意する。粉末回折データの横軸の値 2Θ を便宜上「回折角」と呼ぶことも実際には多いが，正しくは「理想的な場合には回折角に相当する値」として理解すべきである。

試料の面(つら)が高過ぎれば，回折データ中でピークが出現する 2Θ 位置は回折角 2θ より高い角度になり，ピークは高角側へシフトしているように見える。

試料の面(つら)が低すぎれば，回折データ中でピークが出現する 2Θ 位置は回折角 2θ より低い角度になり，ピークが低角側へシフトしているように見える。

これらのことから，「粉末回折データに関するピーク・シフト・モデル」としては，むしろ「見かけ回折角 2Θ の真の回折角 2θ からのずれ」を $\Delta 2\Theta \equiv 2\Theta - 2\theta$ と定義して，

$$\Delta 2\Theta \simeq + \frac{2\Delta S \cos\Theta}{R} \tag{6.1.A.4}$$

と表現する方がわかりやすい。

この効果を**試料位置ずれ効果** specimen displacement effect (スペシメン ディスプレイスメント エフェクト) と呼ぶ。

例えば，ゴニオメーター半径 $R = 185$ mm，回折角（ゴニオメーター角） $2\Theta = 30°$，位置ずれ $\Delta S = +0.05$ mm の時，見た目の回折ピーク位置のずれは，

$$\Delta 2\Theta \simeq \frac{2 \times 0.05 \times \cos(30° \times \pi/360°)}{185} \times \frac{180°}{\pi} \simeq 0.0299°$$

となり，ピークの位置が 0.03° 程度高角側にずれることになる。このずれは，現在市販される粉末X線回折測定システムでは，十分に検出可能な大きさである。

試料位置ずれ効果は，純粋に装置に由来するのではなく，試料によって変化する。試料位置ずれ効果によるピークシフトを補正するために，実測の回折データを再現できるように，式 (6.1.A.4) 中の位置ずれパラメーター ΔS の値を調整する方法をとる場合がある。 (↩)

（補足 6.1.A.1）マクローリン展開を使った試料位置ずれ効果のモデル化（↩）

式 (6.1.A.1) に基づいて，

$$\cos\Theta \tan(\Theta - \Delta\Theta) = \sin\Theta - \Delta S/R \tag{6.1.A.1.1}$$

とする。式 (6.1.A.1.1) から $\Delta S = 0$ のとき $\Delta\Theta = 0$ となることはすぐにわかる。

式 (6.1.A.1.1) の両辺を ΔS で微分すれば，

$$\frac{\partial}{\partial \Delta S}\left[\cos\Theta \tan(\Theta - \Delta\Theta)\right] = -\frac{1}{R} \quad \Rightarrow \quad -\frac{\partial \Delta\Theta}{\partial \Delta S} \cos\Theta \sec^2(\Theta - \Delta\Theta) = -\frac{1}{R} \tag{6.1.A.1.2}$$

$\Delta S = 0$ のとき

$$-\left(\frac{\partial \Delta\Theta}{\partial \Delta S}\right)_{\Delta S=0} \sec\Theta = -\frac{1}{R} \quad \Rightarrow \quad \left(\frac{\partial \Delta 2\Theta}{\partial \Delta S}\right)_{\Delta S=0} = \frac{2\cos\Theta}{R} \tag{6.1.A.1.3}$$

となる。$\Delta 2\Theta$ の ΔS についてのマクローリン展開 (Maclaurin expansion) の一次の項までをとれば,

$$\Delta 2\Theta = (\Delta 2\Theta)_{\Delta S=0} + \left(\frac{\partial \Delta 2\Theta}{\partial \Delta S}\right)_{\Delta S=0} \Delta S + \cdots \approx \frac{2\Delta S\cos\Theta}{R} \tag{6.1.A.1.4}$$

となる。（↩）

（補足 6.3.A）最小自乗法と最尤推定（↩）

最小自乗法（最小二乗法）(least-squares method) は，データの統計誤差が既知である場合の最尤推定 (maximum likelihood estimation) に相当するものと考えても良い。

横軸値 $\{x_j\}$ $(j = 0,1,\cdots,n-1)$ に対して観測された強度値が $\{y_j\}$ であり，強度値の統計誤差が $\{\epsilon_j\}$ と表されるとする。強度値 $\{y_j\}$ が横軸値 $\{x_j\}$ の函数として

$$y_j \approx f(x_j, \mathbf{a}) \tag{6.3.A.1}$$

とモデル化されるとする。$\mathbf{a} = (a_0, a_1, \cdots, a_{m-1})$ が，強度モデル $f(x_j, \mathbf{a})$ の含む「調整可能なパラメータ」であるとする。

最尤推定は，観測結果 $\{y_j\}$ の実現確率が最大となるようなパラメータの組 $\mathbf{a}$ を求めることとみなせる。

通常の粉末 XRD 装置では，X線の強度は光子計数法 (photon counting method) によって測定され，X線の強度は「観測されたX線光子のカウント数」として記録される。現代的な半導体X線検出器は，適切に設計・選定され，適切な制御・信号処理が施される場合には，例えば 95% 以上の量子効率でX線光子を信号パルスとして検出する。

計数回路が充分に高速に動作する光子計数検出システムでは，光子検出のイベントはポアソン (Poisson) 過程に従うと仮定しうる。ポアソン過程では，平均カウント数 λ のときに n カウントされる確率はポアソン分布に従い，確率質量函数 (probability mass function; PMF) が

$$P_{\text{Poisson}}(n;\lambda) = \frac{\lambda^n}{n!}\,\mathrm{e}^{-\lambda} \tag{6.3.A.2}$$

と表される。ポアソン分布の平均が λ であれば，分散も λ であり，標準偏差は $\sqrt{\lambda}$ となる。ポアソン分布には（平均と分散も含む）1 階以上のキュムラントはすべて λ になると言う特徴がある。（補足 6.3.A.1）

Poisson 分布は，$\lambda \gtrsim 20$ では平均 λ，標準偏差 $\sqrt{\lambda}$ の正規分布で良く近似できる（e.g. 菊田, 1992）。つまり，

$$P_{\text{Poisson}}(n;\lambda) \approx \frac{1}{\sqrt{2\pi\lambda}}\,\exp\left[-\frac{(n-\lambda)^2}{2\lambda}\right] \tag{6.3.A.3}$$

と近似できる。

Figure 6.3.A.1 に $\lambda = 20.0$ の場合のポアソン分布の確率質量函数 (probability mass function; PMF) $P_{\text{Poisson}}(n,\lambda)$ の値と平均 $\lambda = 20.0$，標準偏差 $\sqrt{\lambda} = \sqrt{20}$ の正規分布の確率密度函数 (probability density function; PDF) $f_{\text{normal}}\left(x;\lambda,\sqrt{\lambda}\right) = \exp\left[-(x-\lambda)^2/2\lambda\right]$ の値とを示す。離散確率分布の確率質量函数と連続確率分布の確率

密度函数とは本来まったく意味が異なるはずだが，ここでは離散確率変数 n の間隔が 1 なので，結果的にポアソン分布の確率質量函数値と正規分布の確率密度函数値とが直接比較しうる値になっている。

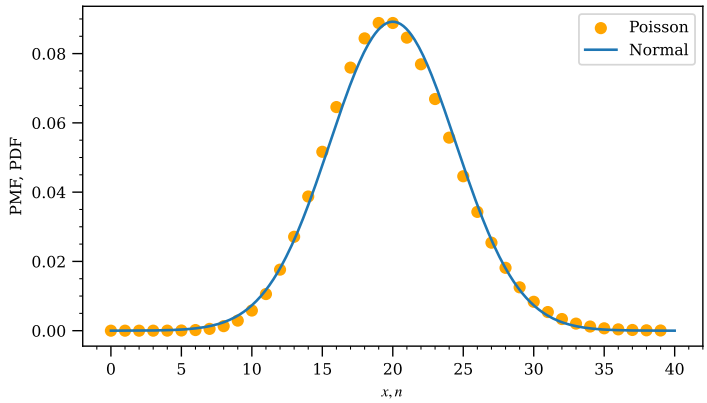

Figure 6.3.A.1　平均 $\lambda = 20$ の Poisson 分布の確率質量函数（橙色のマーカー）と，平均 $\lambda = 20$，標準偏差 $\sqrt{\lambda} = \sqrt{20}$ の正規分布 (normal distribution) の確率密度函数（水色の曲線）　(↩)

X線回折分野では，計数法によって得られたX線光子のカウント数が n だったとき，統計分布が「平均 n，標準偏差 $\sqrt{n}$ の正規分布」に従うとする仮定の用いられる場合が多い。しかし現実に起こりうる「カウント数ゼロ」の場合に，論理的には決定的な不整合をもたらす。仮に標準偏差がカウント数の平方根に等しいとすれば，「カウント数ゼロ」は「標準偏差（誤差）ゼロ」を意味することになる。式 (6.3.A.3) の表現では「ゼロで割る」ことになり，通常の数値計算システムでは「実行時エラー (runtime error)」を発生する。

過去には，その矛盾を回避するために「カウント数ゼロのデータは除外する」という方法のとられる場合があった。しかし，観測されたカウント数がゼロであることは，「期待される観測強度が低い」という情報を含んでいるのに，このやり方では，その情報をまったく無視してしまうことになる。

Antoniadis ら (1990) は，リートベルト精密化 (Rietveld refinement) の際に，例えば初めの最適化ではゼロデータを除外するが，２回目以降の精密化では，ゼロデータの出現した標本点位置であっても，リートベルト精密化の結果推定された強度値の平方根を誤差値と仮定し直して最適化を繰り返す方法を提案した。リートベルト精密化で推定された強度値が正の値になり，このような精密化を繰り返して収束に至れば，論理的な不整合は回避できることになる。Antoniadis らは，このような解析手法を最尤推定 (maximum likelihood method) と称した。しかし，この手法を用いることによりリートベルト解析の結果が顕著に改善されるような実例が示されたわけではない。Antoniadis らの手法は，確かに当時有力であった粉末X線回折データ解析ソフトウェアの含む論理的な不整合を回避しうるものであったが，実際の粉末 XRD データの解析で効果が現れるのは，ごく特殊な場合に限られたことも推測される。（補足 6.3.A.2）

単純化のため単一ピークの形状分析の場合について詳しく検討する。すべての観測強度値 $\{Y_j\}$ が統計的に互いに独立であり，その観測強度値は，横軸の値 $\{X_j\}$ とパラメータ $\mathbf{a} = (a_0, \cdots, a_{m-1})$ の函数 $Y_j = f(X_j; \mathbf{a})$ としてモデル化されるとする。統計分布の標準偏差 $\{\epsilon_j\}$ が知られているとすれば，観測された強度値 $\{Y_0, \cdots, Y_{n-1}\}$ の実現する確率 $P(Y_0, \cdots, Y_{n-1}; \mathbf{a})$ は，

$$P(Y_0, \cdots, Y_{n-1}; \mathbf{a}) \approx \prod_{j=0}^{n-1} \frac{1}{\sqrt{2\pi}\epsilon_j} \exp\left\{-\frac{[Y_j - f(X_j; \mathbf{a})]^2}{2\epsilon_j^2}\right\} \tag{6.3.A.4}$$

と近似できる。この函数 $P(Y_0, \cdots, Y_{n-1}; \mathbf{a})$ は尤度函数 (likelihood function) あるいは単に「尤度」と呼ばれ，この函数を最大化するパラメータ $\mathbf{a}$ を求めることを「最尤推定」と呼ぶ。最尤推定を，文字の通りに「最も尤もらしい推定」と解釈しても良い。対数函数 $\ln x$ は単調増加関数なので，尤度を最大化することと尤度の

対数（対数尤度 logarithmic likelihood（ロガリズミック ライクリフッド））を最大化することとは，同じことである。対数尤度 $\ln P(Y_0, \cdots, Y_{n-1}; \mathbf{a})$ は

$$\ln P(Y_0, \cdots, Y_{n-1}; \mathbf{a}) \approx \sum_{j=0}^{n-1} \frac{1}{\sqrt{2\pi}\epsilon_j} - \sum_{j=0}^{n-1} \frac{[Y_j - f(X_j; \mathbf{a})]^2}{2\epsilon_j^2} \tag{6.3.A.5}$$

と表され，式 (6.3.A.5) の右辺第 1 項はパラメータ $\mathbf{a}$ によらない定数項であるから，最尤推定は「重み付き残差平方和」$\sum_{j=0}^{n-1} [Y_j - f(X_j; \mathbf{a})]^2 / 2\epsilon_j^2$ を最小化することと概ね同じことになる。「XRD データ解析の分野」で，重み付き最小平方法 (weighted least-squares method) が好んで使われることには，このような背景が存在する。

粉末 XRD データの解析に最尤推定が効果的に機能する実例は，Antoniadis (1990) らとはまったく異なる文脈で示された (Ida & Izumi, 2011)。 (↩)

（補足 6.3.A.1）ポアソン分布のキュムラント（↩）

ポアソン分布のモーメント母函数 (moment generating function) $M(\theta)$ は，

$$M(\theta; \lambda) \equiv \sum_{n=0}^{\infty} \mathrm{e}^{n\theta} P(n; \lambda) = \sum_{n=0}^{\infty} \mathrm{e}^{n\theta} \frac{\lambda^n}{n!} \mathrm{e}^{-\lambda} = \mathrm{e}^{-\lambda} \sum_{n=0}^{\infty} \frac{\left(\lambda \mathrm{e}^{\theta}\right)^n}{n!} = \exp\left(\lambda \mathrm{e}^{\theta} - \lambda\right)$$

と表される。キュムラント母函数 (cumulant generating function) $K(\theta)$ は，

$$K(\theta, \lambda) \equiv \ln M(\theta) = \lambda \mathrm{e}^{\theta} - \lambda$$

と表され，1 階以上の偏微分が

$$\frac{\partial}{\partial\theta} K(\theta, \lambda) = \lambda\, \mathrm{e}^{\theta} \underset{\theta \to 0}{\longrightarrow} \lambda$$

$$\frac{\partial^2}{\partial\theta^2} K(\theta, \lambda) = \lambda\, \mathrm{e}^{\theta} \underset{\theta \to 0}{\longrightarrow} \lambda$$

$$\vdots$$

などとなることから，1 以上の任意の k について

$$\frac{\partial^k}{\partial\theta^k} K(\theta, \lambda) = \lambda\, \mathrm{e}^{\theta} \underset{\theta \to 0}{\longrightarrow} \lambda$$

となることを確認できる。 (↩)

（補足 6.3.A.2）粉末 XRD データの統計誤差（↩）

通常の実験室型 XRD 装置のように，真空管中の金属ターゲットへ高電圧（数万ボルト）で加速した電子線を照射することによって発生するX線を用いるか，直流磁場と交流電場を協調させて光速近くまで電子を加速するシンクロトロン加速器の放出する軌道放射光を使うかによらず，多くの粉末 XRD データ解析ソフトウェアでは「観測されるX線回折強度が，ランダムに配向した多数の結晶粒からの回折に由来すること」を前提とすることが通例であった。回折X線を写真フィルムに照射して検出していた20 世紀中頃の時期までは，この前提が問題とされることはなかったと推測される。

1940–1950 年頃にガイガー計数器 (Geiger counter) やシンチレーション計数器 (scintillation counter) などの計数型X線検出器が実用化された。その普及が進むことにともなって，粉末 XRD 測定のために用いる粉末試料は，丁寧に細かく粉砕する必要があることなども経験的に知られるようになった。

光子計数法 (photon counting method) により測定されたX線強度がポアソン統計に従い，カウント数の平方根に相当する統計誤差をともなうこと自体は正しい。この統計誤差は**計数統計誤差** (counting statistical error) と呼ばれる。一方で，現実的な測定条件で粉末 XRD 測定を行うと，ランダムに配向した結晶子（結晶粒）のうち，偶然回折条件条件を満たす配向をとり，観測される回折X線強度に寄与しうる結晶子（結晶粒）の数は限られたものになることが粉末 XRD データの統計的な誤差のうち主要な要因になる場合が少なくないことも知られるようになった (Alexander et al., 1948; Wolff, 1958; Wolff et al., 1959)。この統計誤差は**粒子統計誤差** (particle statistical error) あるいは標本化誤差 (sampling error) と呼ばれることがある。しかし，素粒子論分野で用いられる「粒子統計」という語とも，主に医薬分野・生物統計学分野などで用いられる標本誤差 (sampling error) とも意味する内容が異なり，曖昧(あいまい)な用語の使用が混乱を招いてきた傾向もあるように思われる。（↩）

（補足 6.3.B）準ニュートン法と非ニュートン法による曲線当て嵌め解析（↩）

Scipy ライブラリでは，個別ピーク形状分析のような曲線当て嵌め (curve fitting) 解析に使うことのできるメソッドとして，scipy.optimize.curve_fit() メソッドと scipy.optimize.leastsq() メソッド，scipy.optimize.least_squares() メソッド，scipy.optimize.minimize() メソッドが提供される。このうち curve_fit() メソッドと leastsq() メソッド，least_squares メソッドでは，準ニュートン法に属する Levenberg-Marquardt アルゴリズムを選択できる。minimize() メソッドでは多様な最適化アルゴリズムを選択可能であり，準ニュートン法系のアルゴリズムである BFGS (Broyden-Fletcher-Goldfarb-Shanno) アルゴリズムや非ニュートン法系の Nelder-Mead 法を選択することもできる。一方で，curve_fit() と minimize() メソッドでは最適化函数の最大呼び出し回数を明示的に指定できない。least_squares() メソッドはヘッセ (Hesse) 行列あるいはその逆行列を参照できない。

Nelder-Mead 法のような非ニュートン法では最適化の際に Hesse 行列を用いないが，数値微分を用いて数値的に Hesse 行列を計算することはできる。

位置 $\{x_j\}$ で実測強度値 $\{y_j\}$ が観測されたとする。また，各強度値の統計誤差が $\{\epsilon_j\}$ と表されるとする。強度値がモデル関数 $f(x;\mathbf{p})$ によって，

$$y_j \approx f(x_j;\mathbf{p})$$

と近似できるとする。ここで $\mathbf{p} = \{p_1, p_2, \cdots, p_m\}$ はモデル関数 $f(x;\mathbf{p})$ の含むパラメータを表す。

重み付き最小自乗最適化 (least squares optimization) では，函数

$$\chi^2(\mathbf{p}) = \sum_{j=1}^{n} \frac{\left[y_j - f(x_j;\mathbf{p})\right]^2}{\epsilon_j^2}$$

が最小になるときのパラメータ $\mathbf{p}$ の値を求める。

$\chi^2(\mathbf{p})$ が極値をとるときの $\mathbf{p}$ の値を求めるためには，方程式

$$\frac{\partial \chi^2(\mathbf{p})}{\partial \mathbf{p}} = \mathbf{0} \quad \Leftrightarrow \quad \begin{pmatrix} \dfrac{\partial \chi^2(\mathbf{p})}{\partial p_1} \\ \dfrac{\partial \chi^2(\mathbf{p})}{\partial p_2} \\ \vdots \\ \dfrac{\partial \chi^2(\mathbf{p})}{\partial p_m} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

と解けば良い。初期値 $\mathbf{p}_0$ に関して，一般的なニュートン・ラフソン法 (Newton-Raphson method) の形式をそのまま当てはめれば，極位置は

$$\mathbf{p} = \mathbf{p}_0 - \frac{\left[\dfrac{\partial \chi^2(\mathbf{p})}{\partial \mathbf{p}}\right]_{\mathbf{p}=\mathbf{p}_0}}{\left\{\dfrac{\partial}{\partial \mathbf{p}}\left[\dfrac{\partial \chi^2(\mathbf{p})}{\partial \mathbf{p}}\right]\right\}_{\mathbf{p}=\mathbf{p}_0}}$$

のように表現されると予想される。

準ニュートン法 (quasi-Newton method) は，最適化のプロセスの中で，以下の式で定義される ヘッセ行列（ヘシアン）(Hesse matrix; Hessian) $\mathbf{H}$

$$\mathbf{H} = \begin{pmatrix} \dfrac{\partial^2 \chi^2(\mathbf{p})}{\partial p_1^2} & \dfrac{\partial^2 \chi^2(\mathbf{p})}{\partial p_1 \partial p_2} & \cdots & \dfrac{\partial^2 \chi^2(\mathbf{p})}{\partial p_1 \partial p_m} \\ \dfrac{\partial^2 \chi^2(\mathbf{p})}{\partial p_2 \partial p_1} & \dfrac{\partial^2 \chi^2(\mathbf{p})}{\partial p_2^2} & \cdots & \dfrac{\partial^2 \chi^2(\mathbf{p})}{\partial p_2 \partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 \chi^2(\mathbf{p})}{\partial p_m \partial p_1} & \dfrac{\partial^2 \chi^2(\mathbf{p})}{\partial p_m \partial p_2} & \cdots & \dfrac{\partial^2 \chi^2(\mathbf{p})}{\partial p_m^2} \end{pmatrix}$$

の逆行列 $\mathbf{H}^{-1}$ を勾配 (gradient) $[\partial \chi^2(\mathbf{p})/\partial \mathbf{p}]_{\mathbf{p}=\mathbf{p}_0}$ にかけて，移動方向と移動の大きさ $(\mathbf{p} - \mathbf{p}_0)$ を決める手順を含む。

最適化プロセスが完了したとき，ヘッセ行列の逆行列の２倍 $\mathbf{C} = 2\mathbf{H}^{-1}$ はパラメータ $\mathbf{p}$ の共分散行列に対応し，パラメータ間の相関が弱い時は，行列 $\mathbf{C}$ の対角項の平方根がパラメータ誤差に相当する。ただし，scipy.optimizeモジュール中の leastsq() や least_squaress()，minimize() などの最適化メソッドはヘッセ行列の逆行列の２倍でなく，ヘッセ行列の逆行列を出力し，それが「パラメータ共分散行列」と誤認される傾向があるように思われる。

Python から利用できる科学技術計算ライブラリである scipy ライブラリの scipy.optimize モジュールに含まれるメソッドのうち，モンテ・カルロ計算を含む "basinhopping()"，差分進化アルゴリズム "differential_evolution()"，擬似焼き鈍しアルゴリズム "dual_annealing()" などは「大域的最適化アルゴリズム global optimization」と呼ばれる。進化論的アルゴリズム (evolutional alghorithm) に分類される「差分進化アルゴリズム」と，進化論的アルゴリズムには分類されないが大域的最適化アルゴリズムの一つである「擬似焼き鈍しアルゴリズム」はヘッセ行列を使わない。「局所的最適化アルゴリズム」に分類されるうちで，ネルダー・ミード (Nelder-Mead) アルゴリズムもヘッセ行列を使わず，非ニュートン法系アルゴリズムに分類される。

しかし，非ニュートン法系アルゴリズムで最適化が行われたとしても，数値計算によりヘッセ行列を求め実験誤差からパラメータ誤差への伝播を評価することは可能である。そのことの理解されない場合が多いことにも，注意をした方が良いかと思われる。（↩）

（補足 6.3.C）フォークト函数（↩）

フォークト函数 (Voigt function) $f_{\mathrm{Voigt}}(x;\sigma,w)$ は，標準偏差 σ の Gauss 型函数

$$f_{\mathrm{Gauss}}(x;\sigma) \equiv \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{6.3.C.1}$$

と半値半幅 w の Lorentz 型函数

$$f_{\mathrm{Lorentz}}(x;w) \equiv \frac{1}{\pi w}\left(1+\frac{x^2}{w^2}\right)^{-1} \tag{6.3.C.2}$$

との畳込

$$f_{\mathrm{Voigt}}(x;\sigma,w) \equiv f_{\mathrm{Gauss}}(x;\sigma) * f_{\mathrm{Lorentz}}(x;w) \equiv \int_{-\infty}^{\infty} f_{\mathrm{Gauss}}(x-y;\sigma)\, f_{\mathrm{Lorentz}}(y;w)\,\mathrm{d}y \tag{6.3.C.3}$$

として定義される。

フォークト関数 $f_{\mathrm{Voigt}}(x;\sigma,w)$ は，尺度化複素補誤差関数 (scaled complex complementary error function)（ファディーヴァ関数） (Faddeeva function) $f_{\mathrm{Faddeeva}}(x;\sigma,w)$ の関数値の実部と等価であり，$f_{\mathrm{Faddeeva}}(x;\sigma,w)$ 関数の値は，冪級数展開と連分数展開を組み合わせて高効率に高精度な値が得られることは 1990 年には知られていた (Poppe & Wijers, 1990a, 1990b)。このことを数式で表現すれば，以下の式 (1.4)–(1.7) のようになる。

$$f_{\mathrm{Voigt}}(x;\sigma,w) = \frac{\mathrm{Re}[f_{\mathrm{Faddeeva}}(z)]}{\sqrt{2\pi}\sigma} \tag{1.4}$$

$$z \equiv \frac{x+\mathrm{i}w}{\sqrt{2}\sigma} \tag{1.5}$$

$$f_{\mathrm{Faddeeva}}(z) \equiv \mathrm{e}^{-z^2}\mathrm{erfc}(-\mathrm{i}z) \tag{1.6}$$

$$\mathrm{erfc}(x) \equiv \frac{2}{\sqrt{\pi}}\int_x^{\infty} \mathrm{e}^{-t^2}\mathrm{d}t \tag{1.7}$$

非商用目的では無料で利用できる SciPy ライブラリ (Virtanen et al., 2020) に含まれる scipy.special.voigt_profile() メソッドを使えば，フォークト関数は容易に計算できる。 (↩)

参考文献

Alexandar, L., Klug, H. P, & Kummer, E. (1948). “Statistical factors affecting the intensity of x-rays diffracted by crystalline powders”, Journal of Applied Physics, **19**(8), 742-753. DOI: https://doi.org/10.1063/1.1698200. (↩)

Antoniadis, A., Berruyer, J. & Filhol, A. (**1990**). “Maximum-likelihood methods in powder diffraction refinements”, Acta Crystallographica A**46**, 692–711. DOI: https://doi.org/10.1107/S0108767390004500. (↩)

Byrd, R. H., Lu, P., Nocedal, J. & Zhu, C. (**1995**). “A limited memory algorithm for bound constrained optimization”, SIAM Journal on Scietific Cumputing, **16**(5), 1190–1208. DOI: https://doi.org/10.1007%2FBF01582063. (↩)

Ida, T. & Kimura, K. (**1999**). “Effect of sample transparency in powder diffractometry with Bragg-Brentano geometry as a convolution”, Journal of Applied Crystallography, **32**, 982–991. DOI: https://dx.doi.org/10.1107/S0021889899008894. (↩)

Ida, T. & Izumi, F. (**2011**). “Application of a theory for particle statistics to structure refinement from powder diffraction data”, Journal of Applied Crystallography, **44**, 921–927. DOI: https://doi.org/10.1107/S0021889811031013. (↩)

Kaiser, D. L. & Watters, Jr., R. L. (**2010**). “Certificate, Standard Reference Material 640d, Si powder, Line Position and Line Shape Standard for Powder Diffraction”, National Institute of

Standards & Technology. URL: https://tsapps.nist.gov/srmext/certificates/archives/640d.pdf (viewed on Apr. 18, 2024) (↩)

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (**1992**) *Numerical Recipes in C: The Art of Scientific Computing, 2nd edition*, Cambridge University Press (Cambridge, New York, Port Chester, Melbourne, Sydney) [ISBN: 0-521-43108-5] (↩)

Rietveld, H. M. (**1969**). "A profile refinement method for nuclear and magnetic structures", Journal of Applied Crystallography, **2**, 65–71. DOI: https://doi.org/10.1107/S0021889869006558. (↩)

Wolff, P. M. de (**1958**). "Particle statistics in X-ray diffractometry: I. general theory", Applied Scientific Research, Section B, **7**, 102–112. Google Scholar: link. (↩)

Wolff, P. M. de, Taylor, J. M. & Parrish, W. (**1959**). "Experimental study of effect of crystallite size statistics on X-ray diffractometer intensities", Journal of Applied Physics, **30**, 63–69. DOI: https://doi.org/10.1063/1.1734976. (↩)

Zhu, C., Byrd, R. H. & Lu, H. (**1997**). "L-BFGS-B: algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization", ACM Transactions on Mathematical Software, **23**(4), 550–560. DOI: https://doi.org/10.1145%2F279232.279236. (↩)

菊田惺志 (**1992**). 「物理工学実験 15 X線回折・散乱技術 上」東京大学出版会. ISBN: 978-4130630351 (↩)