

最尤推定法

尤度関数とは:

事象 (x_k) が起こる確率を、既知のパラメータ (a_k) の確率密度関数

$$P(X = x_i | a_k) = \prod_i \left\{ \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left[-\frac{\varepsilon_i(a_k)^2}{2\sigma_i^2} \right] \right\} = \prod_i \left(\frac{1}{\sqrt{2\pi}\sigma_i} \right) \cdot \exp \left[-\sum_i \frac{\varepsilon_i(a_k)^2}{2\sigma_i^2} \right]$$

などとする、逆に $X = (x_i)$ がわかっているとし、

パラメータ (a_k) がどれだけ尤もらしいか (尤度) を表す確率密度関数とみなし、

上記の確率密度関数を変数 (a_k) の関数として

尤度関数 $P(a_i) = P(x_i | a_i)$ という。

最尤推定法

誤差 $\varepsilon_i = f(x_i, a_i) - y_i$ が分散 σ_i の正規分布に従うとする。データ (x_i, y_i) に対するパラメータ (a_i) の尤度関数は

$$P(a_i) = \prod_i \left(\frac{1}{\sqrt{2\pi}\sigma_i} \right) \cdot \exp \left[-\sum_i \frac{\varepsilon_i^2}{2\sigma_i^2} \right]$$

尤度を最大化するパラメータを求めるのが「最尤推定法」。

$$\max P(a_i) = \max \ln P(a_i) = \min \sum_i \frac{\varepsilon_i^2}{\sigma_i^2}: \text{最小二乗法に一致する}$$

線形最小二乗法 $y = a + bx$ の場合: 行列表示

$$f(x_i) = a + bx$$

酒井英行訳、M.C.Barford著、実験精度と誤差、丸善

$$\begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum y_i x_i \end{pmatrix}$$

$$\begin{pmatrix} n & \langle x \rangle \\ \langle x \rangle & \langle x^2 \rangle \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \langle y \rangle \\ \langle xy \rangle \end{pmatrix}$$

$$S_{xx} = \langle x^2 \rangle: x \text{の分散}$$

$$S_{xy} = \langle xy \rangle: x \text{と} y \text{の共分散}$$

x_i にはほとんど誤差がなく、誤差は y_i から来るものとする、

$$a_n = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} = \frac{\sum (x_i - \langle x \rangle)(y_i - \langle y \rangle)}{\sum (x_i - \langle x \rangle)^2} \quad (3.9)$$

$$\frac{\sum x^2 \sum y - \sum x \sum xy}{\sum x^2 \sum y - \sum x \sum xy}$$

線形最小二乗法 $y = a + bx$ の誤差

酒井英行訳、M.C.Barford著、実験精度と誤差、丸善

$$f(x_i) = a + bx_i$$

$$\varepsilon_i = f(x_i) - (a + bx_i) \quad \text{目的関数 } S = \sum \varepsilon_i^2 \text{ を最小化}$$

$$\begin{pmatrix} n & s_x \\ s_x & s_{xx} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} s_y \\ s_{xy} \end{pmatrix} \quad s_x = \sum x_i, s_y = \sum y_i, s_{xx} = \sum x_i^2, s_{xy} = \sum x_i y_i$$

$\langle A \rangle$ は A の平均

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} s_{xx} & -s_x \\ -s_x & n \end{pmatrix} \begin{pmatrix} s_y \\ s_{xy} \end{pmatrix} \quad \Delta = ns_{xx} - s_x^2 = n \sum (x_i - \langle x \rangle)^2$$

$$b = \frac{ns_{xy} - s_x s_y}{\Delta} = \frac{\sum (x_i - \langle x \rangle)(y_i - \langle y \rangle)}{\sum (x_i - \langle x \rangle)^2} \quad a = \frac{s_{xx} s_y - s_x s_{xy}}{\Delta} = \langle y \rangle - b \langle x \rangle$$

$$\sigma_y^2 = \langle y^2 \rangle - \langle y \rangle^2 - \frac{(\langle xy \rangle - \langle x \rangle \langle y \rangle)^2}{\langle x^2 \rangle - \langle x \rangle^2} = \frac{1}{n^2} \left[ns_{yy} - s_y^2 - \frac{(ns_{xy} - s_x s_y)^2}{ns_{xx} - s_x^2} \right]$$

$$\text{標準誤差: } S_a = \frac{\sigma_y \sqrt{\langle x^2 \rangle}}{\sqrt{(n-2)(\langle x^2 \rangle - \langle x \rangle^2)}}$$

$$S_b = \frac{\sigma_y}{\sqrt{(n-2)(\langle x^2 \rangle - \langle x \rangle^2)}}$$

$$\text{相関係数: } r = s_{xy} / \sqrt{s_{xx} s_{yy}}$$

LSQ: Polynomial

線形最小二乘法: 多項式

$$f(x) = \sum_{k=0}^n a_k x^k \quad S = \sum_{i=1}^N \left(y_i - \sum_{k=0}^n a_k x_i^k \right)^2 \quad \frac{dS}{da_l} = -2 \sum_{i=1}^N x_i^l \left(y_i - \sum_{k=0}^n a_k x_i^k \right) = 0$$

$$\sum_{k=0}^n \sum_{i=1}^N a_k x_i^{k+l} = \sum_{i=1}^N y_i x_i^l \quad (l = 0, 1, \dots, N)$$

$$\begin{pmatrix} n & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^N \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & & \sum x_i^{N+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & & \sum x_i^{N+2} \\ \vdots & & & \ddots & \\ \sum x_i^N & \sum x_i^{N+1} & \sum x_i^{N+2} & & \sum x_i^{2N} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \\ \vdots \\ \sum y_i x_i^N \end{pmatrix}$$

線形最小二乗法：一般関数の場合

$$f(x) = \sum_{k=1}^n a_k f_k(x) \quad S = \sum_{i=1}^N \left(y_i - \sum_{k=1}^n a_k f_k(x_i) \right)^2$$
$$\frac{dS}{da_l} = -2 \sum_{i=1}^N f_l(x_i) \left(y_i - \sum_{k=1}^n a_k f_k(x_i) \right) = 0$$

$$\begin{pmatrix} \sum f_1(x_i)f_1(x_i) & \sum f_1(x_i)f_2(x_i) & \sum f_1(x_i)f_3(x_i) & \cdots & \sum f_1(x_i)f_N(x_i) \\ \sum f_2(x_i)f_1(x_i) & \sum f_2(x_i)f_2(x_i) & \sum f_2(x_i)f_3(x_i) & & \sum f_2(x_i)f_N(x_i) \\ \sum f_3(x_i)f_1(x_i) & \sum f_3(x_i)f_2(x_i) & \sum f_3(x_i)f_3(x_i) & & \sum f_3(x_i)f_N(x_i) \\ \vdots & & & \ddots & \\ \sum f_N(x_i)f_1(x_i) & \sum f_N(x_i)f_2(x_i) & \sum f_N(x_i)f_3(x_i) & & \sum f_N(x_i)f_N(x_i) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} \sum y_i f_1(x_i) \\ \sum y_i f_2(x_i) \\ \sum y_i f_3(x_i) \\ \vdots \\ \sum y_i f_N(x_i) \end{pmatrix}$$

**If $f(x)$ is linear with respect to fitting parameters,
final solution is obtained by one matrix operation**

係数に関して線形であれば、1度の行列計算で最終解が得られる

ex. $f(x) = a + b \log x + c / x$

$$f(x, y) = a + bxy + cy / x$$

関数の解: Newton-Raphson法 (Newton法)

$f(x) = 0$ の解を求める

$$f(x_0 + dx) = f(x_0) + dx f'(x_0) \sim 0$$

$$\Rightarrow x_1 = x_0 + dx = x_0 - f(x_0) / f'(x_0)$$

計算では $f'(x_0)$ を差分計算で置き換えられる
割線法 (セカント法、はさみうち法):

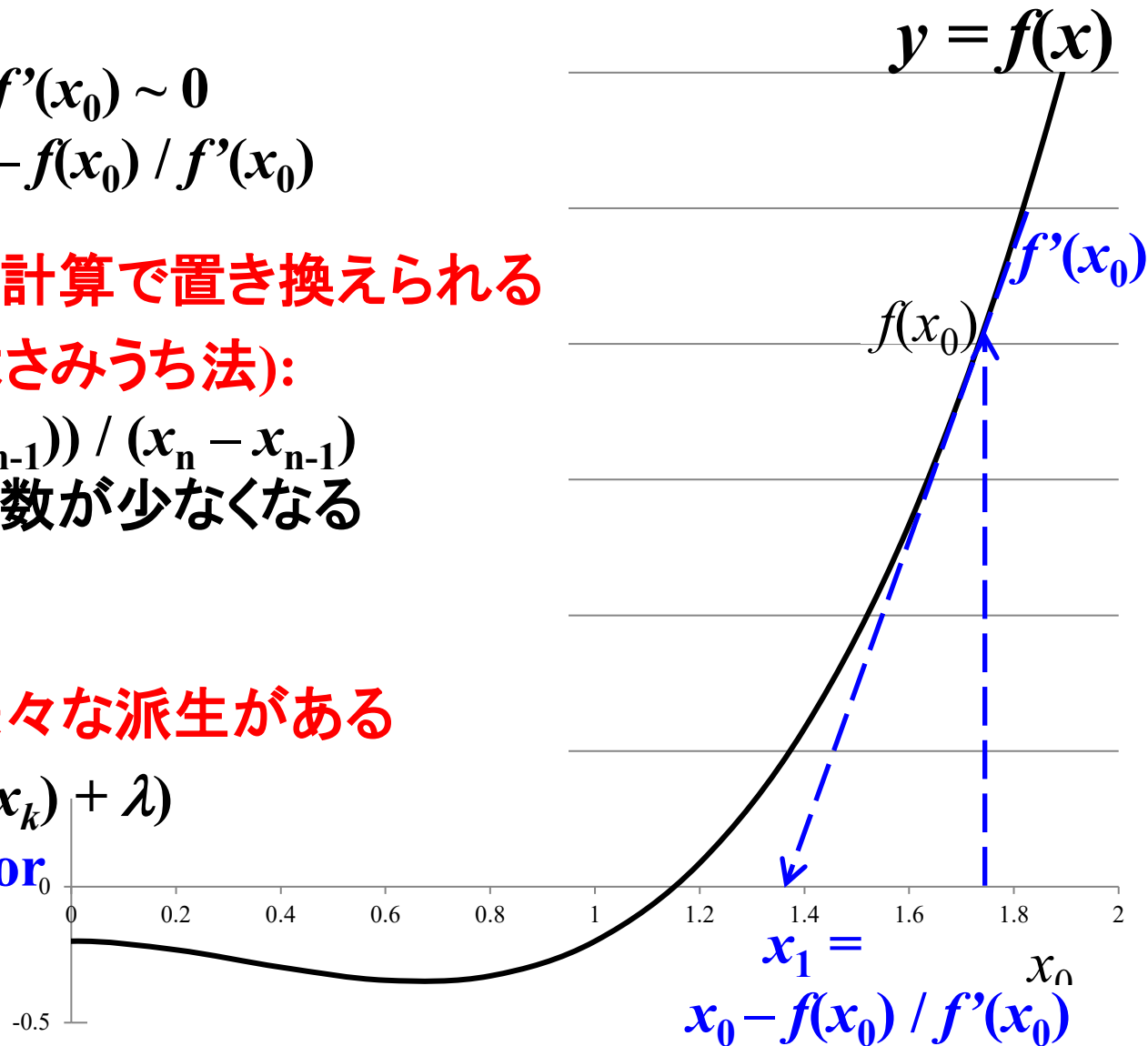
$$f'(x) = (f(x_n) - f(x_{n-1})) / (x_n - x_{n-1})$$

を使う。 $f(x)$ の計算回数が少なくなる

発散を抑える工夫で様々な派生がある

$$x_{k+1} = x_k - f(x_k) / (f'(x_k) + \lambda)$$

λ : Damping Factor

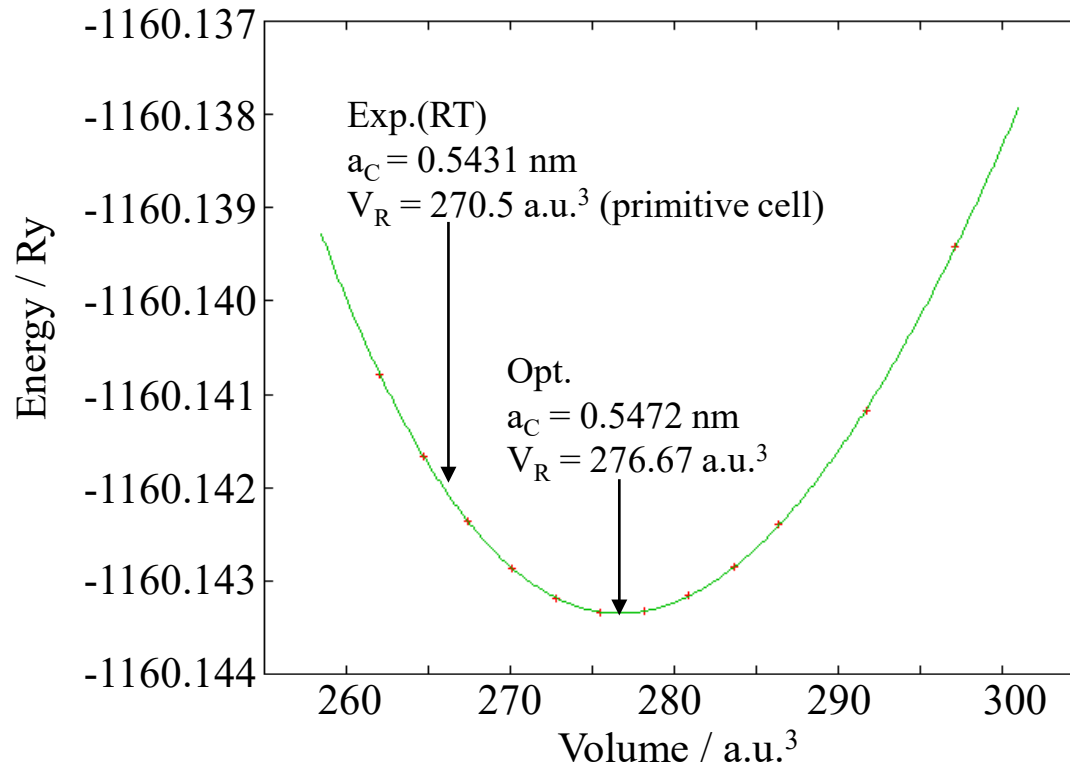


NL optimization of crystal structure:

Illustrative approach

安定構造: 図解による解法

Calculate total energy by quantum calculations by varying a lattice parameter
ex. Si

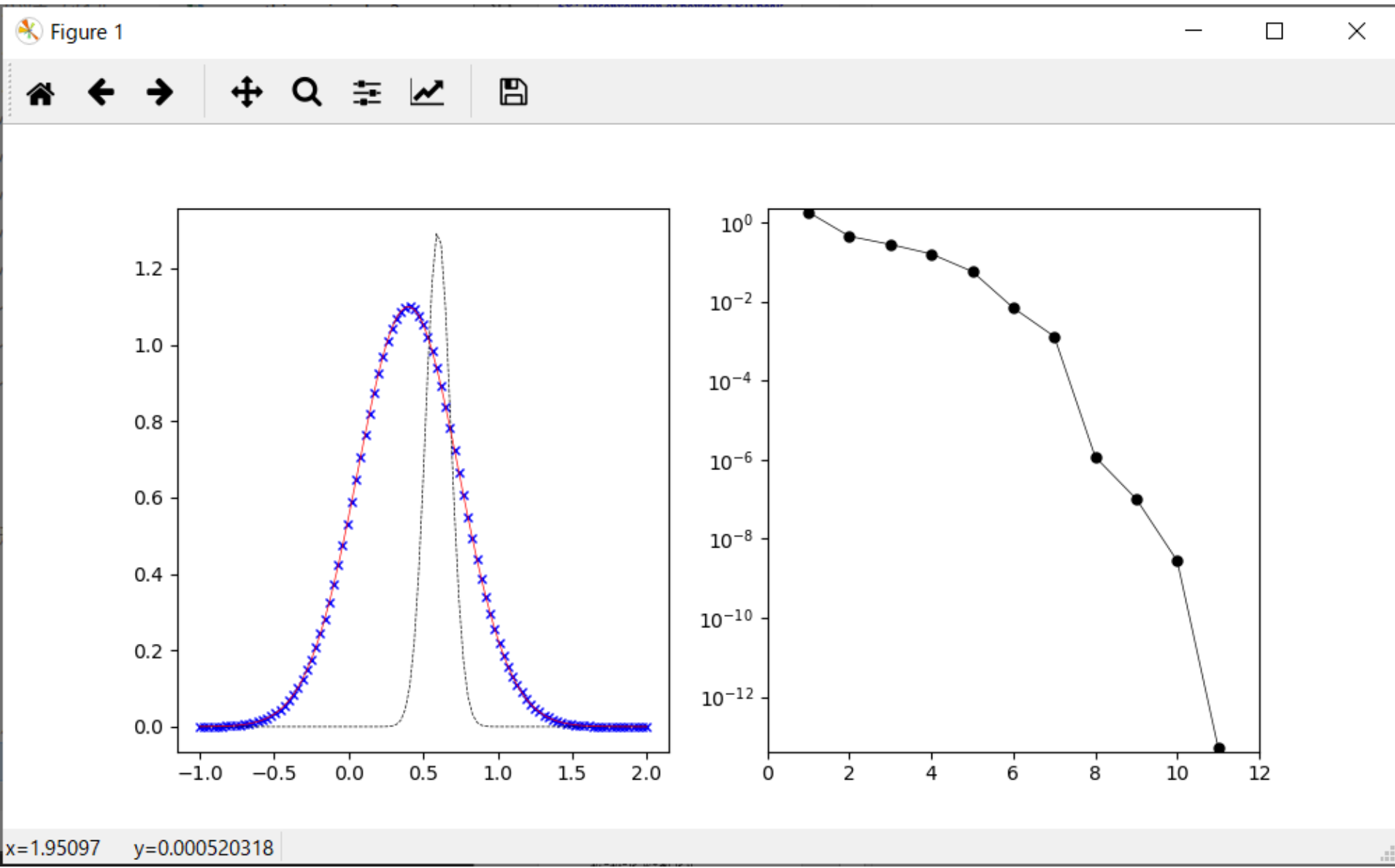


$$E = E_{\min} + 1/2B_0(V/V_0)^2$$

$$B_0 \text{ (GPa)} = 87.57 \text{ GPa (exp: 97.88 GPa)}$$

Ex.: Deconvolution of powder XRD peak

peakfit-scipy-minimize.py



分光解析に使われるプロファイルモデル

Lorentz関数

$$I_L(x) = \frac{1}{1 + [(x - x_0)/w]^2} \quad w: \text{半値半幅}$$

Gauss関数

$$I_G(x) = \frac{1}{a_w w \pi^{1/2}} \exp\left\{-\left[(x - x_0)/(a_w w)\right]^2\right\}$$
$$a_w = (\ln 2)^{-1/2} = 0.832554611$$

Voigt関数:

Lotentz型の固有スペクトルに
他の要因のGauss型の広がり重なる
畳み込み積分 (Convolution) であらわされる

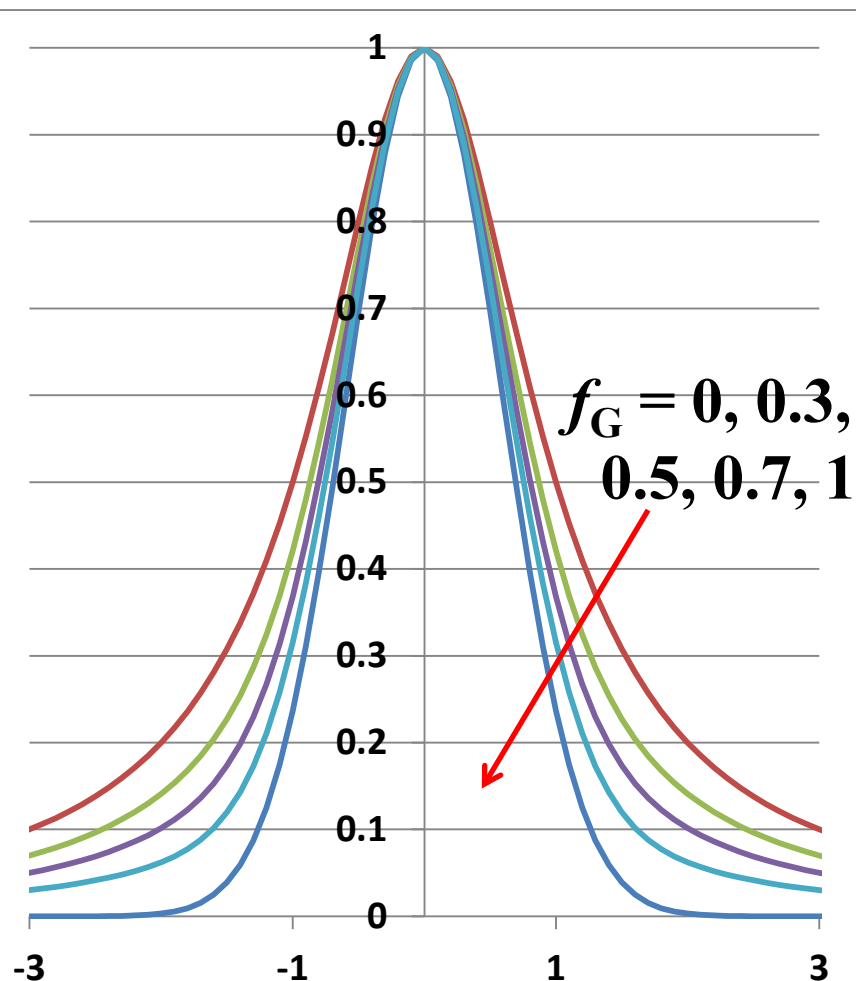
$$I_V(x) = \int_{-\infty}^{\infty} I_G(x') I_L(x - x') dx'$$
$$= \frac{a_V}{\pi} \int_{-\infty}^{\infty} \frac{\exp(-x'^2)}{a_V^2 + (x - x')^2} dx'$$

Pseudo-Voigt関数:

Voigt関数の簡略版

$$I_{PV}(x) = f_G I_G(x) + (1 - f_G) I_L(x)$$

f_G : Gauss関数分率



多変数関数の最適化: Newton-Raphson法

多変数への拡張: 最小化関数 $F(x_l)$ の最小値を求める

$$f_k(x_l) = \partial F(x_l) / \partial x_k = 0$$

繰り返し計算: $f_k(x_l + \delta x_l) \sim f_k(x_l) + \sum_{k'} \delta x_{k'} \partial f_k(x_l) / \partial x_{k'} = 0$

$$x_{l,1} = x_{l,0} - (\partial f_k(x_l) / \partial x_{k'})^{-1} (f_k) = x_{l,0} - (F''_{kk'})^{-1} (F'_k)$$

$$F''_{kk'} = \frac{\partial^2 F(\mathbf{x})}{\partial x_k \partial x_{k'}}$$

Hessian (ヘッセ) 行列

(ヘッセ行列の固有値をヘッシアンと呼ぶ)

Hessian行列は正定値であるとは限らない (極大値、鞍点)

$\Rightarrow F''$ が降下方向を与えるとは限らない

F'' を正定値行列で置き換え、発散を抑える

$$x_{l,1} = x_{l,0} - (F''_{kk'} + \lambda I)^{-1} (F'_k)$$

λ : Dumping Factor

Quasi-Newton method (準Newton法)

矢部博, 工学基礎 最適化とその応用, 数理工学社 (2006)

最小化関数 $F(x_l)$

繰り返し計算: $x_l^{(i+1)} = x_l^{(i)} - (\partial^2 F / \partial x_k \partial x_{k'})^{-1} (\partial F / \partial x_k)$

$F''_{kk'} = \partial^2 F / \partial x_k \partial x_{k'}$: Hessian (ヘッセ) 行列

Newton法の問題:

- (1) Hessian行列は二次行列のため、計算に時間がかかる
- (2) Hessian行列の固有値は負になることもある $\Rightarrow$ 極大値を探索
- (3) 発散しやすい

準Newton法:

- (1,2) Hessian行列の計算を過去の一次微分の数を使って近似
- (3) 探索方向 $-(\partial^2 F / \partial x_k \partial x_{k'})^{-1} (\partial F / \partial x_k)$ に沿って
線形探索を行う

Davidon-Fletcher-Powell (DFP) 法

矢部博, 工学基礎 最適化とその応用, 数理工学社 (2006)

$$F(x_l^{(k)} + \alpha d) = F(x_l^{(k)}) + \nabla F(x_l^{(k)})^T d + \frac{1}{2} d^T B^{(k)} d \sim 0$$

$B^{(k)} d = -\nabla F(x_l^{(k)})$ で探索方向 d を決める

DFP法: 準Newton法のはじまり

$$s^{(k)} = x^{(k+1)} - x^{(k)}, y^{(k)} = \nabla F(x_l^{(k+1)}) - \nabla F(x_l^{(k)})$$

$$\begin{aligned} B^{(k+1)} &= B^{(k)} + \frac{(y^{(k)} - B^{(k)} s^{(k)}) \cdot y^{(k)T} + y^{(k)} \cdot (y^{(k)} - B^{(k)} s^{(k)})^T}{s^{(k)T} \cdot y^{(k)}} \\ &\quad - \frac{s^{(k)T} \cdot (y^{(k)} - B^{(k)} s^{(k)})}{(s^{(k)T} \cdot y^{(k)})^2} y^{(k)} \cdot y^{(k)T} \\ &= B^{(k)} - \frac{B^{(k)} s^{(k)} \cdot y^{(k)T} + y^{(k)} \cdot (B^{(k)} s^{(k)})^T}{s^{(k)T} \cdot y^{(k)}} + \left(1 + \frac{s^{(k)T} B^{(k)} s^{(k)}}{s^{(k)T} \cdot y^{(k)}} \right) \end{aligned}$$

Broyden-Fletcher-Goldfarb-Shanno (BFGS) 法

矢部博, 工学基礎 最適化とその応用, 数理工学社 (2006)

BFGS法: 準Newton法でも最も有効と認められている

$$s^{(k)} = x^{(k+1)} - x^{(k)}, y^{(k)} = \nabla F(x_l^{(k+1)}) - \nabla F(x_l^{(k)})$$

$$B^{(k+1)} = B^{(k)} - \frac{B^{(k)} s^{(k)} (B^{(k)} s^{(k)})^T}{s^{(k)T} B^{(k)} s^{(k)}} + \frac{y^{(k)} y^{(k)T}}{s^{(k)T} y^{(k)}}$$

アルゴリズム:

STEP 0: 初期値 $x^{(0)}$ 、初期行列 $B^{(0)}$ (通常は単位行列) を与える。

STEP 1: $B^{(k)} d = -\nabla F(x_l^{(k)})$ から探索方向 $d^{(k)}$ を求める

STEP 2: 直線探索によって、 $d^{(k)}$ 方向のステップ幅 $\alpha^{(k)}$ を決める

STEP 3: $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$ とする

STEP 4: 収束したら計算終了。

収束条件が満たされていなければSTEP 5へ

STEP 5: $s^{(k)}, y^{(k)}$ を計算し、 $B^{(k+1)}$ を求め、STEP 1へ

最急降下法 (Steepest Descend) 法

矢部博, 工学基礎 最適化とその応用, 数理工学社 (2006)

残差関数の傾きのみから最小値を探索する。勾配法としては最も単純。

- SD法: ベクトル $-(df/dx_i)dx_i$ 方向へ進めば S^2 は小さくなる

$$x_i^{(i+1)} = x_i^{(i)} - \alpha(df/dx_i)$$

α は適当に決める。

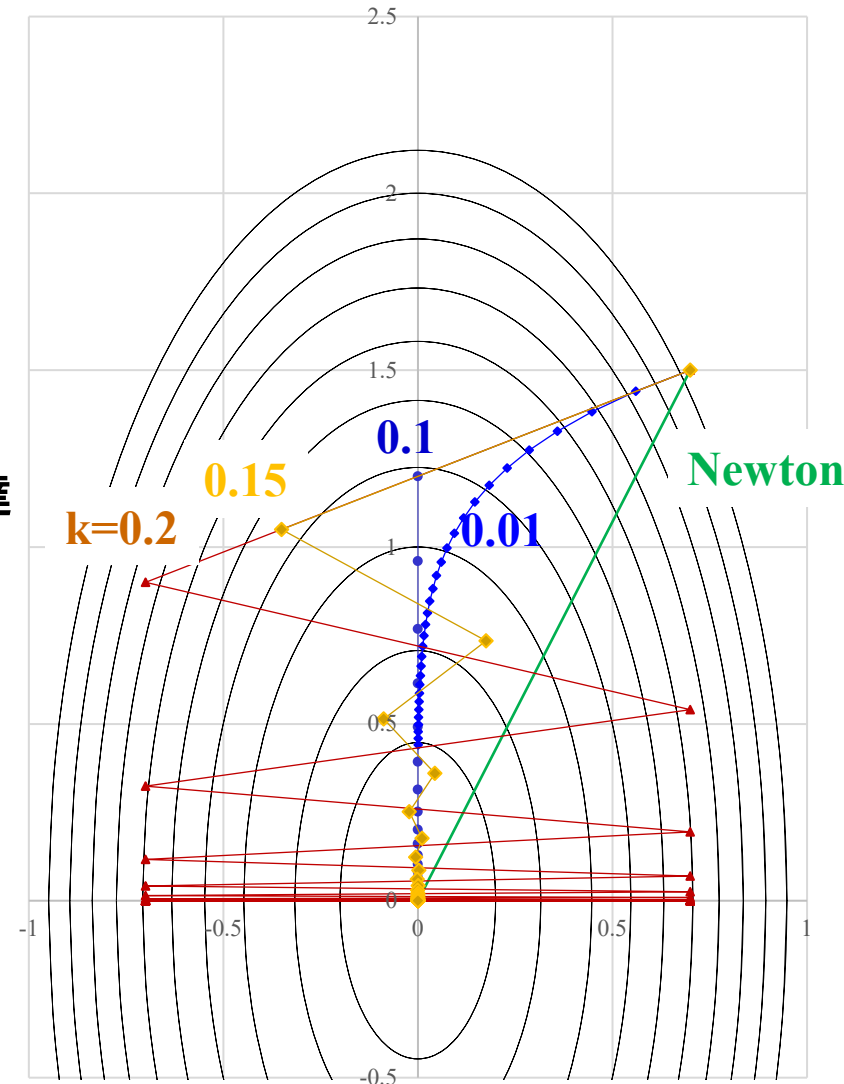
右の例:

$S^2 = f(x_i) = 5x_1^2 + x_2^2$, 初期値 $x_1 = 0.7, x_2 = 1.5$

- Newton法:
楕円問題の場合は一度目の計算で最適値に到達

- SD法:
 $\alpha = 0.3$: 発散 (グラフにプロットしていない)
0.2, 0.15: 振動しながら収束
0.1: 最初の1度目で x_1 の最適値に到達
0.01: 振動せず、緩やかに収束

「 S^2 が大きく非対称な場合、最急勾配方向は
最小値方向とは大きく異なることがある」
=> 共役勾配法



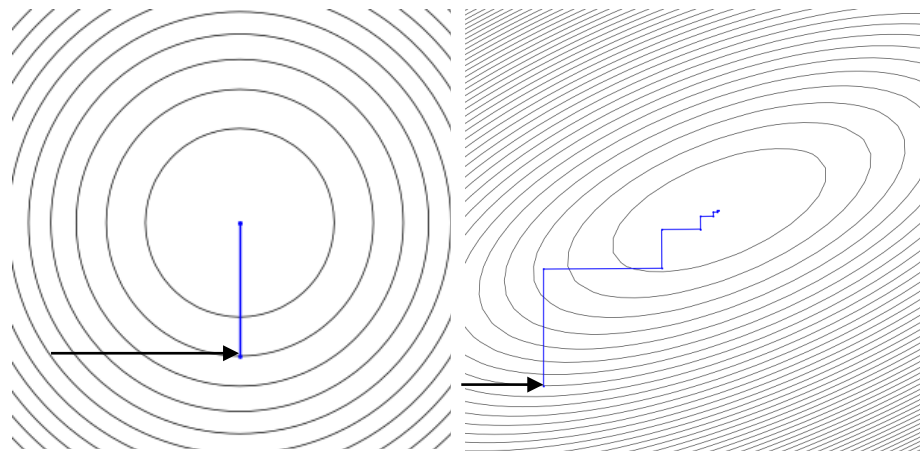
共役勾配 (Conjugate Gradient) 法

矢部博, 工学基礎 最適化とその応用, 数理工学社 (2006)

行列 A に対してベクトル u, v が $u^t A v = 0$ であるとき、 u と v は互いに共役の関係にあるという

- 共役な探索方向に沿って正確な直線探索を実行していけば、有限回の反復で2次関数の最小解に到達することが期待される

等高線が円の場合、変数個数回の探索で最小値に到達できるが...



座標変換により、楕円を円に変換すればよい

$$u^T P^T P v = u^T A v = 0$$

u と $A v$ 互いに共役なベクトル

- 初期値 x_0 を与える
- 初期探索方向 d を再急降下方向にとる

$$d = -\nabla f$$

- 直接探索法に従って α_k を決め、

$$x_{k+1} = x_k + \alpha_k d_k$$

を計算する

- 探索方向を

$$d_k = d_{k-1} - \frac{d_{k-1} \cdot \nabla f(x_k)}{d_{k-1} \cdot d_{k-1}} \nabla f(x_k)$$

に更新する

- 3,4を繰り返して収束させる

4.では、 d_k は d_i ($i = 1, \dots, k-1$) のすべてに直交するため、このループは有限回しか実行できない $\Rightarrow$ 時々 d_k をリセット

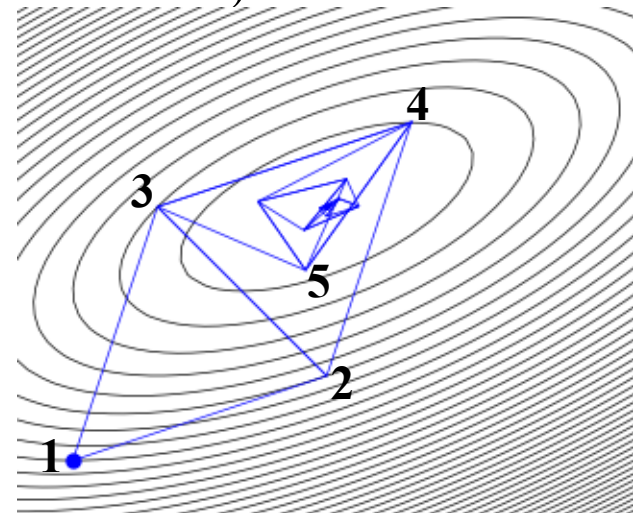
単体 (Simplex) 法 (Amoeba法)

南茂夫 編著、科学計測のための波形データ処理、CQ出版 (1986年)

単体 (Simplex): n 次元空間で $(n+1)$ 個の頂点を作る図形

$F(x_i)$ の最小値を求める

1. $(n+1)$ 個の初期値で頂点 $F(x_i)$ ($i = 1, 2, \dots, n+1$) を
 $F(x_i) > F(x_{i'})$ ($i < i'$) となるように並べ替える
2. 最大値を取る頂点 x_i 以外が作る重心を $x_G = \sum_{i=2} x_i / n$ とする
3. 直線 $x_1 - x_G$ 上で新しい点を次の順に求めて F を計算する
 - (i) 鏡映 : $x_R = x_1 + \alpha(x_G - x_1)$ (たとえば $\alpha = 2$)
 - (ii) 拡大 : $x_E = x_1 + \beta(x_G - x_1)$ (たとえば $\beta > 2$)
 - (iii) 縮小鏡映 : $x_{CR} = x_1 + \gamma(x_G - x_1)$ (たとえば $1.0 < \gamma < 2.0$)
 - (iv) 縮小 : $x_{CW} = x_1 + \delta(x_G - x_1)$ (たとえば $0 < \delta < 1.0$)
4. (i)~(iv)のうちで最初に $F(x) < F(x_1)$ を満たす点を
 x_1 と交換する。
この操作を繰り返す



非線形最小化問題の解法

$F(x)$ の最小値(最大値)を求める

勾配法

- **Newton-Raphson法:**
二次微分行列を使って最小値を探索
- **準Newton法**
二次微分行列を一次微分ベクトルから近似して最小値を探索
- **最急降下法 (Steepest Descent):**
一次微分から傾きの方向のみで最小値を探索
- **共役勾配 (Conjugate Gradient) 法:**
変数の変位ベクトルの共役方向へ最小値を探索
- **Marquart法**
 $f_j(x_i)$ の一次微分の行列を使って最小値を探索

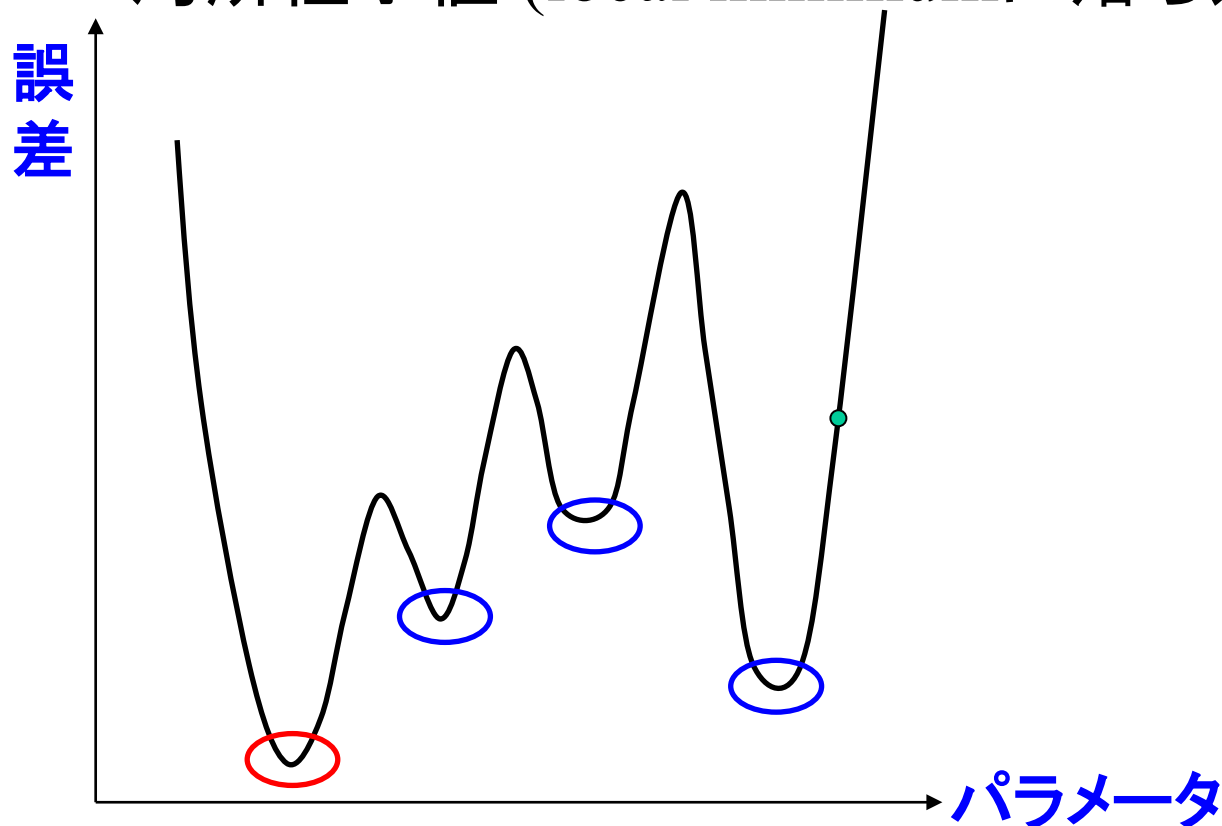
直接探索法

- **単体 (Simplex) 法 (Amoeba法)**
一定のルールに従い、試行錯誤で最小値を探索

非線形 (多値) 方程式の注意

- ・ 解が複数ある場合も
- ・ ほとんどの場合、1度の計算で最適解を求めることは無理
 - ・ 収束したことを確認する
 - ・ 大域最小値を求める

⇔ 局所極小値 (local minimumに落ち込む)



非線形最適化アルゴリズムの傾向

	A	B
収束速度	×	○
収束安定性	○	×
安定収束範囲	○	×
使い方	第一段階	第二段階

A: 単体 (Simplex) 法

A,B: 共役勾配法 (Conjugate Gradient: CG)

B: 最急降下法 (Steepest Descent: SD)

B: Newton-Raphson法 ・準Newton法

- Davidson-Fletcher-Powell (DFP)

- Broyden-Fletcher-Goldfarb-Shanno (BFGS)