2.8.7 ポテンシャル減少法

- ポテンシャル関数を用いて最適解への収束を保証する

$$\boxed{f_\rho(x,s) = \rho \ln(x^T s) + n \ln\left(\frac{x^T s}{n}\right) - \sum_{i=1}^n \ln x_i s_i, \quad \rho \text{ は定数}}$$

- 探索方向 $(\Delta x, \Delta y, \Delta s)$ を次の連立１次方程式から求める

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S^k & 0 & X^k \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} b - Ax^k \\ c - A^T y^k - s^k \\ \mu e - \frac{n+\sqrt{n}}{n} X^k s^k \end{pmatrix}$$

- ステップサイズ $\bar{\alpha}$ は以下の１次元最適化問題から求める

$$\begin{cases} \text{最小化：} & f_\rho(x^k + \alpha \Delta x, s^k + \alpha \Delta s) \\ \text{条件：} & (x^k + \alpha \Delta x, y^k + \alpha \Delta y, s^k + \alpha \Delta s) \in \mathcal{F}^0 \\ & 0 \le \alpha \le 1 \end{cases}$$

2.8.8 実装上の問題点

- 先程の数値実験からも実装の工夫はとても重要なことが分かる

問題名	CPLEX 1.0	CPLEX 2.2	CPLEX 5.0	CPLEX 7.1	アルゴリズム
car	1555	203	117	67	内点法
continent	365	110	100	47	主単体法
energy1	1217	46	32	22	内点法
energy2	10130	171	72	32	内点法
energy3	21797	153	113	82	内点法
fuel	5620	999	340	125	内点法
initial	3832	102	51	16	双対単体法
schedule	152404	252	132	48	内点法

高速化

- データの前処理 (退化対策，データのスケーリング等)
- データの疎性を利用 (疎行列に対する数値解法，計算の順番)
- データの使いまわし (複数回連立 1 次方程式を解く場合等)
- 高速な数値ライブラリー使用

- 並列化，マルチコアープロセッサの使用
- ハードウェアの効率化(プロセッサのキャッシュ等)

自分で組むより既存のソフトウェアを使う方が効率よい場合もある

2.9 ソフトウェア

- 有名な線形計画問題を解くソフトウェア

商用ソフトウェア

- **Gurobi (http://www.octobersky.jp/products/gurobi/gurobi.html)**
- **CPLEX (http://www-06.ibm.com/software/jp/websphere/ilog/optimization/core-products-technologies/cplex)**
- **NUOPT (http://www.msi.co.jp)**
- **MOSEK (http://www.mosek.com)**
- **LINGO (http://www.lindo.com)**
- **FICO Xpress (http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx)**

無料ソフトウェア

- **SoPlex (http://soplex.zib.de)**

- **CLP, COIN-OR プロジェクト (http://www.coin-or.org)**
- **その他のソフトウェアは例えば (http://plato.asu.edu/sub/nlores.html#LP-problem) を参照**

2.10 演習問題

問題1

$$\begin{cases} \text{最小化:} & -5x_1 - x_2 \\ \text{条件:} & 2x_1 + \frac{1}{2}x_2 \leq 8 \\ & x_1 + x_2 \geq 5 \\ & x_1 \geq 0,\ x_2 \geq 0 \end{cases}$$

に相当する図を描いて，最適解，最適値を求めよ．

問題2

線形計画問題において，(x, y, s) が Karush-Kuhn-Tucker 最適性条件を満たしているならば，x は主問題そして (y, s) は双対問題の最適解であることを示せ (３０ページ参照)．

問題3

３８ページにおいて，$\mathrm{rank}(A) = m$ であれば，任意の右辺ベクトルに対して連立１次方程式 (2.2) は唯一の解 $(\Delta x, \Delta y, \Delta s)$ を持つことを示せ．ヒント：$E = (S^0)^{-1/2}(X^0)^{1/2}$ を含むような方程式をたてて，$(\Delta x, \Delta s)$ が必ず唯一に求まることをまず示せ．Δy の唯一

性は $\mathrm{rank}(A) = m$ から求まる.

$\boxed{\text{問題 4}}$

定理 6 を証明せよ. ヒント : $\log(1-x) \leq -x, \quad x \leq 1$ の関係式を考慮しても良い.

Chapter 3

制約なし非線形計画問題 (unconstrained nonlinear programming problem)

3.1 概要

ここでは以下の問題を考える.

$$\boxed{\begin{cases} \text{最小化：} & \boldsymbol{f}(\boldsymbol{x}) \\ \text{条件：} & \boldsymbol{x} \in \mathbb{R}^n \end{cases}}$$

ただし，$f : \mathbb{R}^n \to \mathbb{R}$，(必ずしもそうである必要はないが) 2回連続微分可能であるとする．

以下の数値手法を解説する．

- シンプレックス法 (simplex method)
- 直線探索法 (line search methods)
- 直交探索法
- パターン探索法
- 最急降下法 (steepest descent method)
- Nesterov の 1 次法 (Nesterov's first-order methods)
- ニュートン法 (Newton method)
- 準ニュートン法 (quasi-Newton methods)
- 記憶制限準ニュートン法 (limited memory quasi-Newton methods)
- 共役勾配法 (conjugate gradiente methods)

3.2 シンプレックス法 [KOWALIK1968]

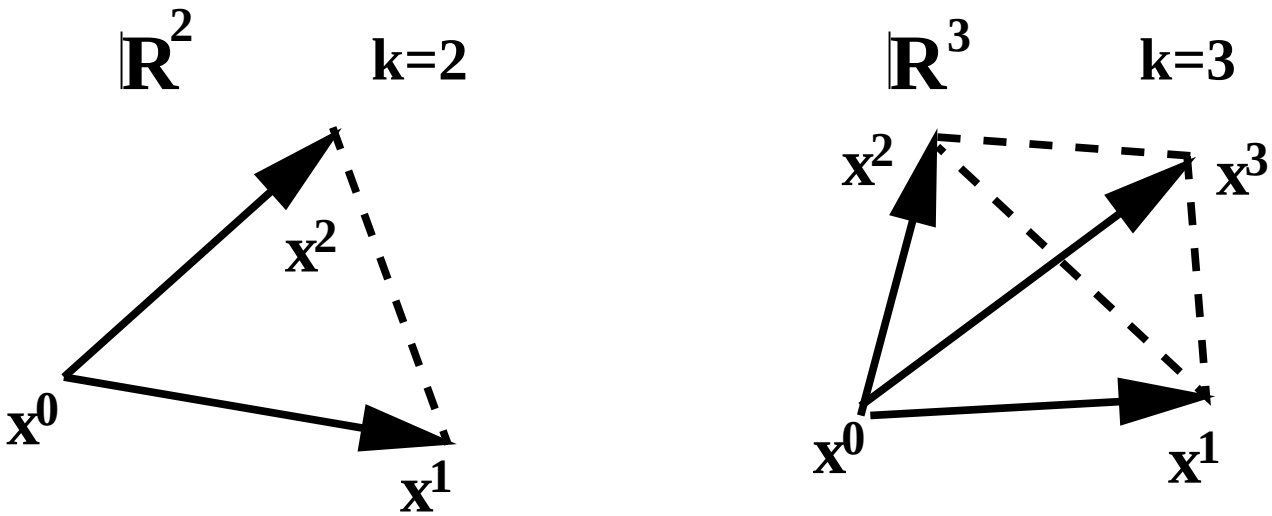
- 線形計画問題に対するシンプレックス法（単体法）とは異なる.
- 関数値だけが分かれば実装できる.

＜＜基本的な考え方＞＞

- ２次元 $(n = 2)$ では三角形，３次元 $(n = 3)$ では四面体を巧妙に反転，拡大，縮小させて１反復で次元に比例した回数の関数評価で最小化を行う.
- 最小点から遠いところでは加速するようになっている.
- 最小点から近いところでは減速し，高精度な近似解を求めるようになっている.

定義 1

$(k \leq n)$ $x^0, x^1, \ldots, x^k \in \mathbb{R}^n$ がアフィン独立であるとは $x^1 - x^0, x^2 - x^0, \ldots, x^k - x^0 \in \mathbb{R}^n$ が線形独立である時をいう.



定義 2

S が $k+1$ $(\leq n+1)$ のアフィン独立な頂点の多面体であるとき, S を $k-$シンプレックスであるという.

<<シンプレックス法>>

- $x^0 \in \mathbb{R}^n$を１つの頂点として持つ$n-$シンプレックスを用いて実数値関数$f : \mathbb{R}^n \to \mathbb{R}$を最小化する．

<<初期シンプレックス>>

$$S^0 = (x^0, x^1, \ldots, x^n) = x^0 E + \begin{pmatrix} 0 & \delta & \sigma & \cdots & \sigma \\ 0 & \sigma & \delta & \cdots & \sigma \\ 0 & \sigma & \sigma & \cdots & \sigma \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \sigma & \sigma & \cdots & \delta \end{pmatrix} \in \mathbb{R}^{n \times (n+1)}$$

ただし，

$$\delta = \lambda \frac{\sqrt{n+1} + n - 1}{\sqrt{2}n}, \quad \sigma = \lambda \frac{\sqrt{n+1} - 1}{\sqrt{2}n},$$

$\lambda > 0$は<u>スケーリング・ファクター</u>である．Eは全て１要素の行列．

例：

$n = 2$のとき，$\delta \approx 0.97$, $\sigma \approx 0.26$

$n = 3$のとき，$\delta \approx 0.94$, $\sigma \approx 0.24$

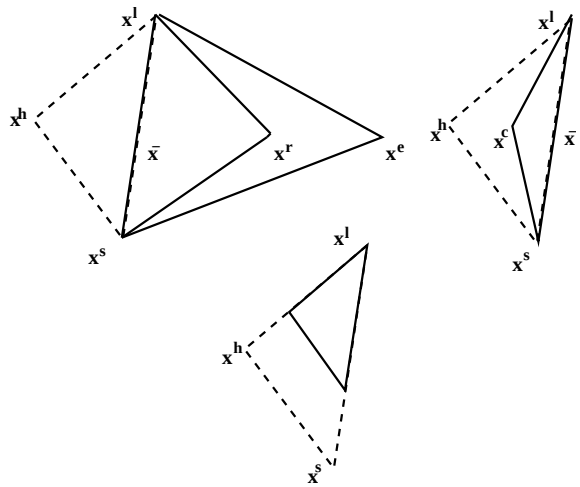
<<シンプレックス法の基本操作>>

各反復にて次の頂点を求める.

$x^h = \mathrm{argmax}_i f(x^i)$	最大の関数値を実現する頂点
$x^s = \mathrm{argmax}_{i \neq h} f(x^i)$	２番目に最大の関数値を実現する頂点
$x^\ell = \mathrm{argmin}_i f(x^i)$	最小の関数値を実現する頂点
$\bar{x} = \frac{\sum_{i \neq h} x^i}{n}$	最大の関数値を実現する頂点を除いた重心

次の操作が定義されている.

鏡映	(reflection)	$x^r = \bar{x} + \alpha(\bar{x} - x^h), \ \ \alpha > 0$
拡張	(expansion)	$x^e = \bar{x} + \gamma(x^r - \bar{x}), \ \ \gamma > 0$
収縮	(contraction)	$x^c = \bar{x} + \beta(x^h - \bar{x}), \ \ \beta \in (0, 1)$
縮小	(reduction)	全ての頂点を x^ℓ の方向に縮小する



$x^h = \mathrm{argmax}_i f(x^i)$ 最大の関数値を実現する頂点
$x^s = \mathrm{argmax}_{i \neq h} f(x^i)$ ２番目に最大の関数値を実現する頂点
$x^\ell = \mathrm{argmin}_i f(x^i)$ 最小の関数値を実現する頂点
$\bar{x} = \frac{\sum_{i \neq h} x^i}{n}$ 最大の関数値を実現する頂点を除いた重心

<<シンプレックス法：アルゴリズム>>

Step 0: シンプレックス S^0 にて，$x^h, x^s, x^\ell, \bar{x}$ を求める．最小解の候補：x^ℓ，最小値の候補：$f(x^\ell)$，$k := 0$ とおく．

Step 1: x^r を求めて，$f(x^r)$ を計算する．

Step 2: $f(x^\ell) \leq f(x^r) \leq f(x^s)$ ならば，x^h を x^r で置き換え，S^{k+1} を更新し，$k := k+1$ として Step 1 にもどる．

Step 3: $f(x^r) < f(x^\ell)$ ならば，$f(x^e)$ を計算する．

Step 3-a: $f(x^e) < f(x^\ell)$ ならば，x^h を x^e で置き換え，S^{k+1} を更新し，$k := k+1$ として Step 1 にもどる．

Step 3-b: そうでなければ，x^h を x^r で置き換え，S^{k+1} を更新し，$k := k+1$ として Step 1 にもどる．

Step 4: $f(x^\ell) < f(x^r) < f(x^h)$ ならば，x^h を x^r で置き換え，収縮を行うため x^c を求め，$f(x^c)$ を計算する．

Step 4-a: $f(x^c) < f(x^h)$ ならば，x^h を x^c で置き換え，S^{k+1} を更新し，$k := k+1$ として Step 1 にもどる．

Step 5: 縮小を行って，S^{k+1} を更新し，$k := k+1$ として Step 1 にもどる．

3.3 直線探索法

＜基本的な考え方＞

- 一般的によく使われる手法．
- 出発点 $\bar{x} \in \mathbb{R}^n$ と関数を局所的に減少させる探索方向 $d \in \mathbb{R}^n$ $(d \neq 0)$ が与えられているとする．つまり，十分小さな $\bar{\epsilon} > 0$ が存在し，

$$f(\bar{x} + \bar{\epsilon} d) < f(\bar{x})$$

  が成り立っていると仮定する．

- 次の 1 次元半直線の最適化問題を近似的に解きたい．

$$\begin{cases} \text{最小化：} & \phi(\alpha) \equiv f(\bar{x} + \alpha d) \\ \text{条件：} & \alpha > 0 \end{cases} \tag{3.1}$$

3.3.1 黄金分割法 (golden section method)

- 以下の２次方程式の正の根を考える．

$$\tau^2 + \tau - 1 = 0, \quad \tau = \frac{-1 \pm \sqrt{5}}{2}, \quad \tau_+ = \frac{-1 + \sqrt{5}}{2} \approx 0.618$$

＜黄金分割法：アルゴリズム＞

Step 0: 問題 (3.1) の近似解が存在すると思われる区間 $[a, b]$ と “解の精度” $\epsilon > 0$ を決める．

Step 1: $k := 1$

$$\begin{aligned} \alpha_1 &= a + 0.382(b - a) \\ \beta_1 &= a + 0.618(b - a) \end{aligned}$$

を求め，$\phi(\alpha_1)$，$\phi(\beta_1)$ を計算する．$a_1 := a, \quad b_1 := b$.

Step 2: もし $\phi(\alpha_k) > \phi(\beta_k)$ ならば Step 3へ．もし $\phi(\alpha_k) \leq \phi(\beta_k)$ ならば Step 4へ．

Step 3-a: もし $b_k - \alpha_k < \epsilon$ ならば終了し，β_k を近似解として出力．

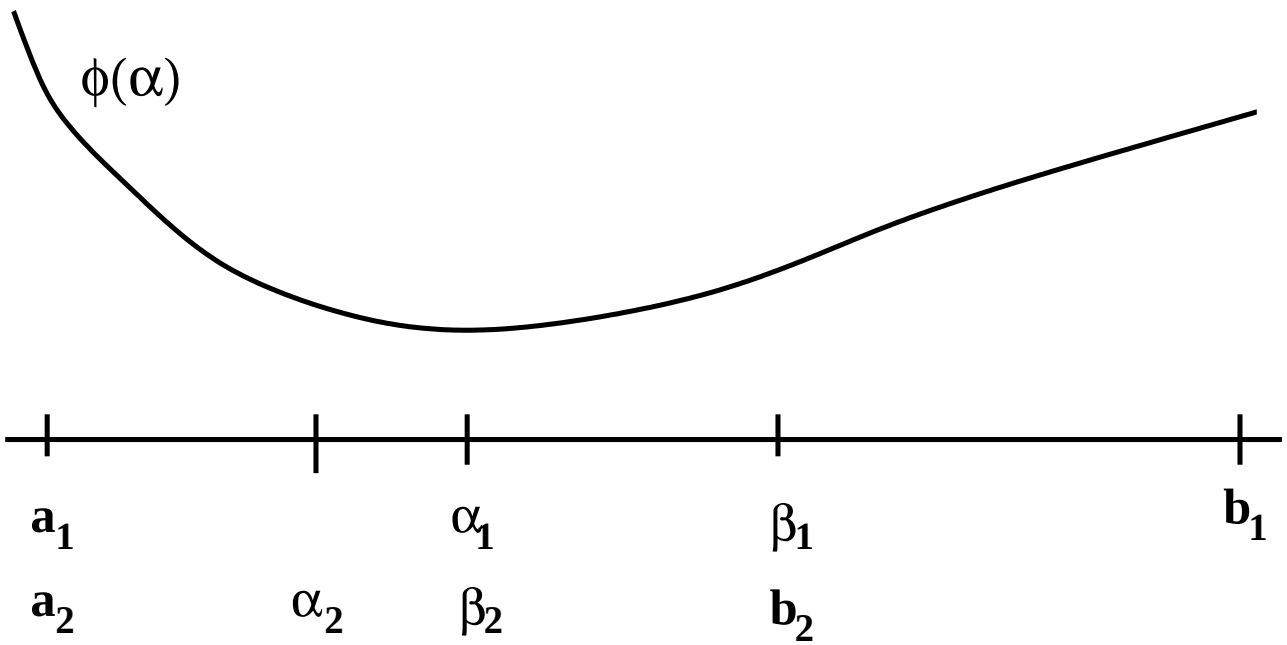
Step 3-b: そうでなければ，

$a_{k+1} := \alpha_k$, $b_{k+1} := b_k$, $\alpha_{k+1} := \beta_k$, $\phi(\alpha_{k+1}) := \phi(\beta_k)$, $\beta_{k+1} := \alpha_{k+1} + 0.618(b_{k+1} - a_{k+1})$, $\phi(\beta_{k+1})$を計算し，$k := k + 1$，Step 2へ.

Step 4-a: もし$\beta_k - a_k < \epsilon$ならば終了し，α_kを近似解として出力.

Step 4-b: そうでなければ，

$a_{k+1} := a_k$, $b_{k+1} := \beta_k$, $\beta_{k+1} := \alpha_k$, $\phi(\beta_{k+1}) := \phi(\alpha_k)$, $\alpha_{k+1} := \alpha_{k+1} + 0.382(b_{k+1} - a_{k+1})$, $\phi(\alpha_{k+1})$を計算し，$k := k + 1$，Step 2へ.



3.3.2　２次関数補間法

＜基本的な考え方＞

- 最小化したい関数 $\phi(\alpha)$ が <u>最小解の近傍では２次関数による近似が良い</u> と仮定する．
- 関数 $\phi(\alpha)$ の最小解が区間 $[\alpha_1, \alpha_3]$ に入っているとして，$\alpha_2 \in [\alpha_1, \alpha_3]$ とする．
- これらの３点で２次関数 $q(\alpha) = a\alpha^2 + b\alpha + c$ と $\phi(\alpha)$ が一致するとする．
- $q(\alpha)$ の最小解 $\bar{\alpha}$ は以下のようにして求められる．

$$\bar{\alpha} = -\frac{b}{2a} = \frac{\alpha_1 + \alpha_2}{2} + \frac{(\phi(\alpha_1) - \phi(\alpha_2))(\alpha_2 - \alpha_3)(\alpha_3 - \alpha_1)}{2((\alpha_2 - \alpha_3)\phi(\alpha_1) + (\alpha_3 - \alpha_1)\phi(\alpha_2) + (\alpha_1 - \alpha_2)\phi(\alpha_3))} \tag{3.2}$$

＜２次関数補間法：アルゴリズム＞

Step 0: 問題 (3.1) の近似解が存在すると思われる区間 $[\alpha_1, \alpha_3] \ni \alpha_2$ と “解の精度” $\epsilon > 0$ を決める．$\phi(\alpha_1), \phi(\alpha_2), \phi(\alpha_3)$ を計算する．

Step 1: 式 (3.2) より，$\bar{\alpha}$ を求める．

Step 2: もし $(\bar{\alpha} - \alpha_1)(\bar{\alpha} - \alpha_3) \geq 0$ ならば，Step 3へ，そうでなければ Step 4へ．

Step 3: $\alpha_1, \alpha_2, \alpha_3, \bar{\alpha}$ から $\alpha_1 < \alpha_2 < \alpha_3$ を再定義し，$\phi(\alpha_1), \phi(\alpha_2), \phi(\alpha_3)$ を計算する．Step 1へ．

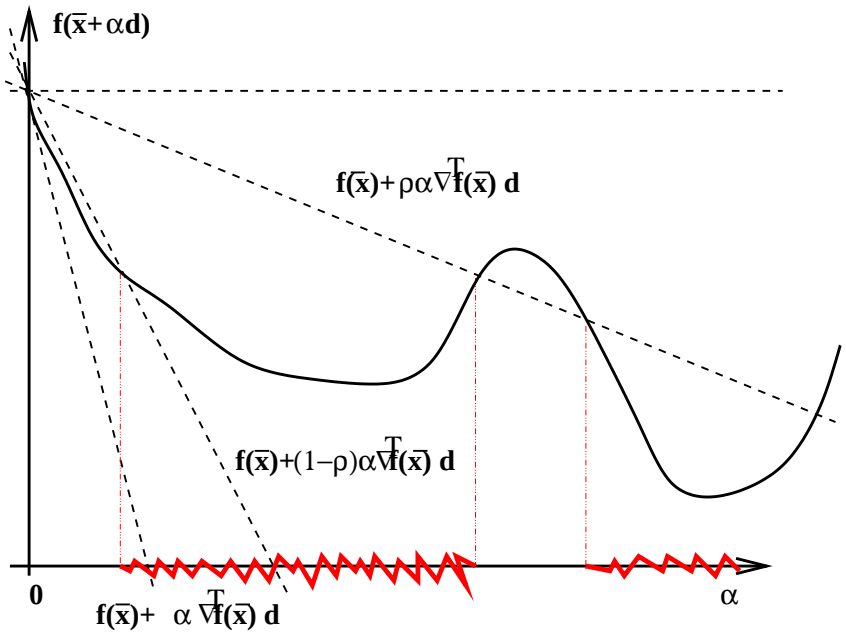
Step 4: もし $|\bar{\alpha} - \alpha_2| < \epsilon$ ならば終了．そうでなければ Step 3へ．

3.3.3 Armijo-Goldsteinルール

- 直線探索法のなかでは次の Wolfe 条件とともにもっともポピュラーな方法である．
- 最小化する関数が微分可能である必要がある．

パラメータ $\rho \in (0, \frac{1}{2})$ をもちいて、次の不等式を満すステップ・サイズ α を求める．

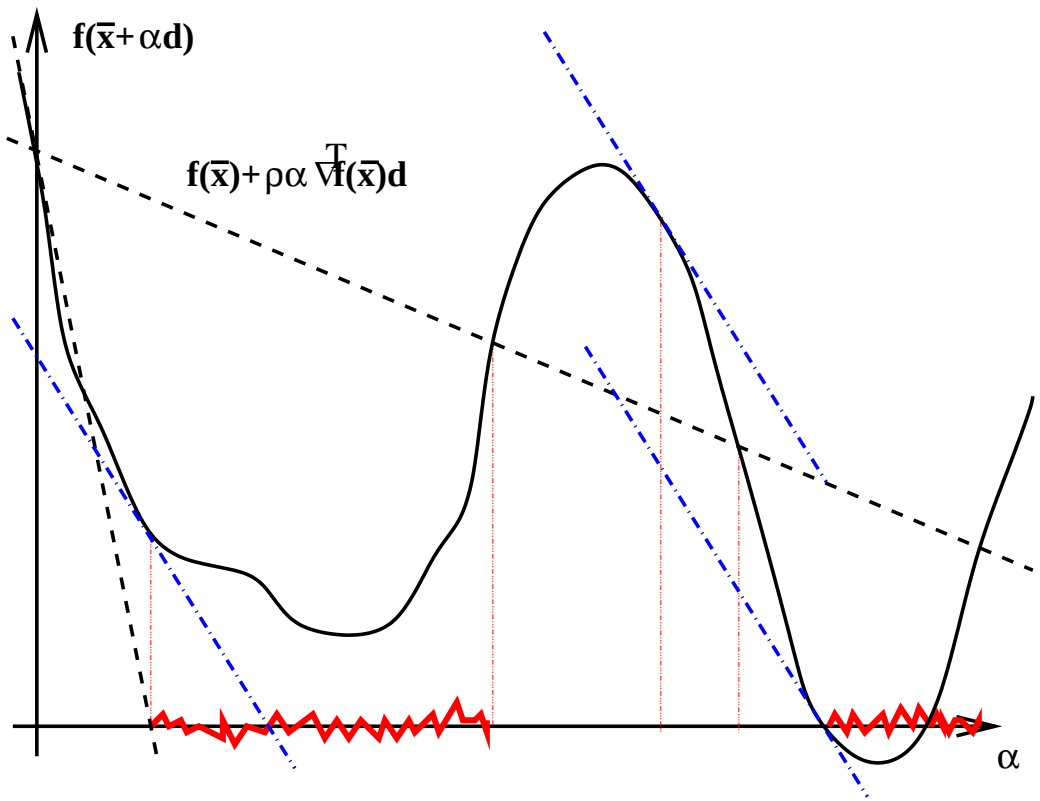
$$f(\bar{x}) + (1-\rho)\alpha \nabla f(\bar{x})^T d \leq f(\bar{x} + \alpha d) \leq f(\bar{x}) + \rho\alpha \nabla f(\bar{x})^T d$$



3.3.4 Wolfe条件

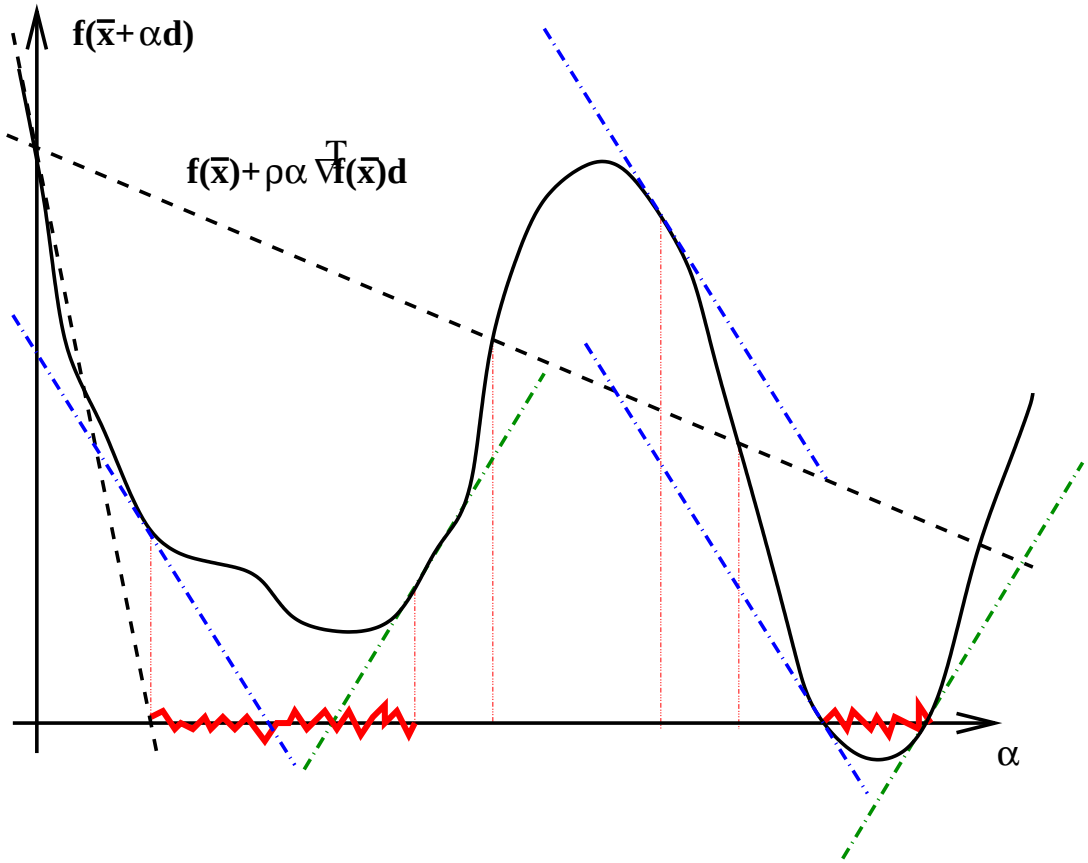
● パラメータ $0 < \rho_1 < \rho_2 < 1$ をもちいて、次の不等式を満す α が Wolfe条件 を満すステップ・サイズと呼ばれている.

$$
\begin{aligned}
f(\bar{x} + \alpha d) &\leq f(\bar{x}) + \rho_1 \alpha \nabla f(\bar{x})^T d \\
\nabla f(\bar{x} + \alpha d)^T d &\geq \rho_2 \nabla f(\bar{x})^T d
\end{aligned}
$$



● パラメータ $0 < \rho_1 < \rho_2 < 1$ をもちいて、次の不等式を満す α が strong Wolfe 条件を満すステップ・サイズと呼ばれている.

$$
\begin{aligned}
f(\bar{x} + \alpha d) &\leq f(\bar{x}) + \rho_1 \alpha \nabla f(\bar{x})^T d \\
|\nabla f(\bar{x} + \alpha d)^T d| &\leq \rho_2 |\nabla f(\bar{x})^T d|
\end{aligned}
$$



3.4 直交探索法

＜基本的な考え方＞

- お互いに直交する方向ベクトルに従って直線探索を行う．

＜直交探索法：アルゴリズム＞

Step 0: d^i $(i = 1, 2, \ldots, n)$ を直交する探索方向，x^0 を初期点，$\epsilon > 0$ を停止基準，$k := 0$ を反復回数とする．

Step 1: $i := 1$ として，$x_i^k := x^0$ とする．

Step 2: $\alpha \in \mathbb{R}$ に対する以下の <u>直線探索を近似的に解く</u>．

$$\{\text{最小化：}\ f(x_i^k + \alpha d^i);\ \text{条件：}\ \alpha \in \mathbb{R}$$

この解を α_i^k として，$x_{i+1}^k := x_i^k + \alpha_i^k d^i$ とおく．

Step 3: $i < n$ であれば，$i := i + 1$ として，step 2に戻る．

Step 4: $x^{k+1} := x_n^k$ とおき，終了条件 $\|x^{k+1} - x^k\| < \epsilon$ を満たしているならば終了．

Step 5: $k := k + 1$ としてstep 1に戻る．

3.5 パターン探索法 [KOWALIK1968]

＜基本的な考え方＞

- 直交探索を使って，現在の点の近くで目的関数が減少する “谷の方向” を求める．
- 上記の “谷の方向” を使って直線探索を行う．

＜パターン探索法：アルゴリズム＞

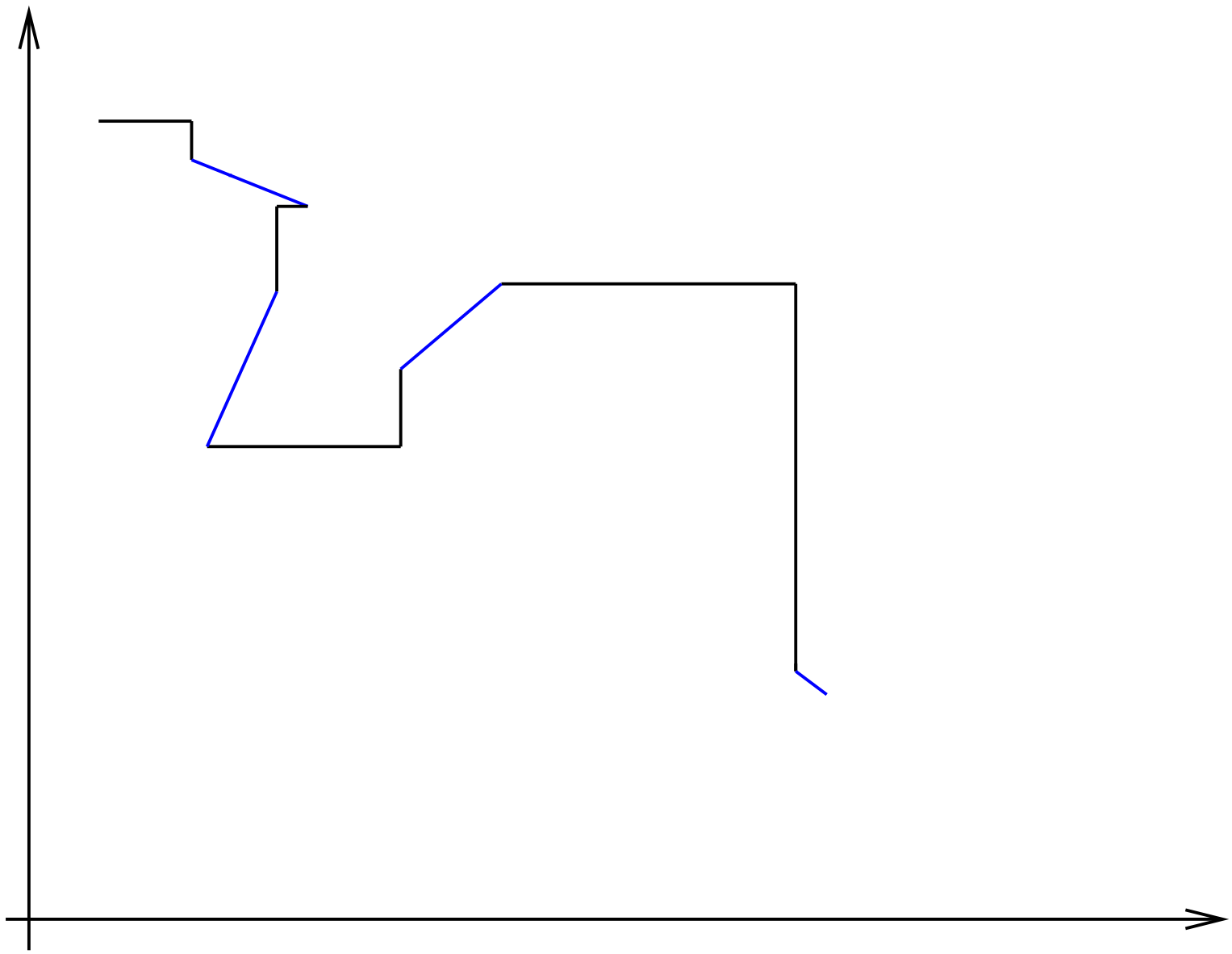
Step 0: d^i $(i = 1, 2, \ldots, n)$ を直交する探索方向，x^0 を初期点，$\epsilon > 0$ を停止基準，$k := 0$ を反復回数とする．

Step 1: d^i $(i = 1, 2, \ldots, n)$ を使って，直交探索を 1 反復行い，x_T^k を求める．

Step 2: $p^k = x_T^k - x^k$ を探索方向として，$\alpha \in \mathbb{R}$ に対する以下の直線探索を近似的に解く．

$$\{\text{最小化：}\ f(x_T^k + \alpha p^k);\ \text{条件：}\ \alpha \in \mathbb{R}$$

この解を α^k として，$x^{k+1} := x_T^k + \alpha^k p^k$ とおく．

Step 3: 終了条件 $\|x^{k+1} - x^k\| < \epsilon$ を満たしているならば終了．

Step 5: $k := k + 1$ として step 1 に戻る．