確率と最適化

勾配法・ペナルティ関数法・内点法

数値最適化アルゴリズム

大規模な数理計画問題を解くためには、効率の良いアルゴリズムが必要。

無制約凸問題

- 勾配法

（制約付き）凸計画問題

- ペナルティ関数法
- 内点法
- 交互方向乗数法(ADMM)

目標

- 無制約凸計画問題

$$\text{minimize } f(x)$$

を考える(f は凸関数であり、$f : \mathbb{R}^n \to \mathbb{R}$)。

- 最適化アルゴリズムは、系列 $x^{(0)}, x^{(1)}, \ldots \in \mathbf{dom} f$ を生成する。ここで、

$$f(x^{(k)}) \to p^*, \quad k \to \infty$$

である。$x^{(0)}$ を初期点と呼ぶ。

仮定

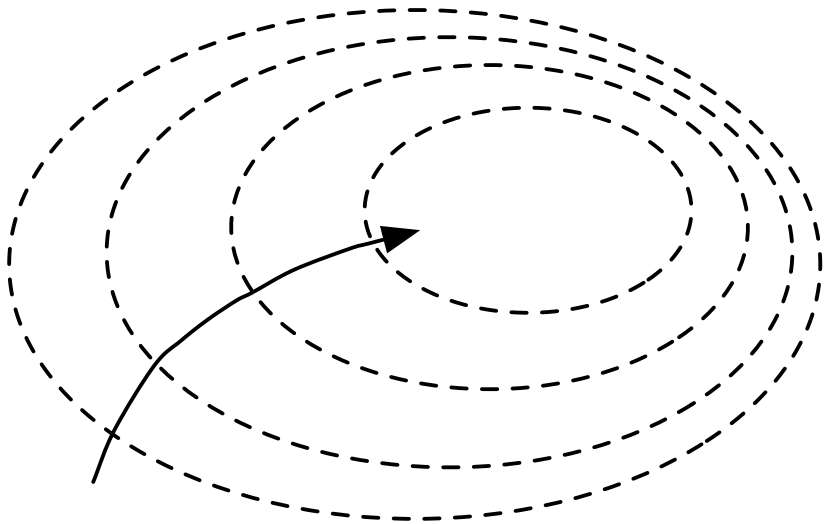
- $x^{(0)} \in \mathrm{dom}\, f$
- **サブレベル集合**

$$S = \{x : f(x) \leq f(x^{(0)})\}$$

が閉集合となっている。

降下法

降下法は、勾配法、ニュートン法を含む一般的な最適化原理。



- $\nabla f(x) = 0$ **を満たす点を探す。**

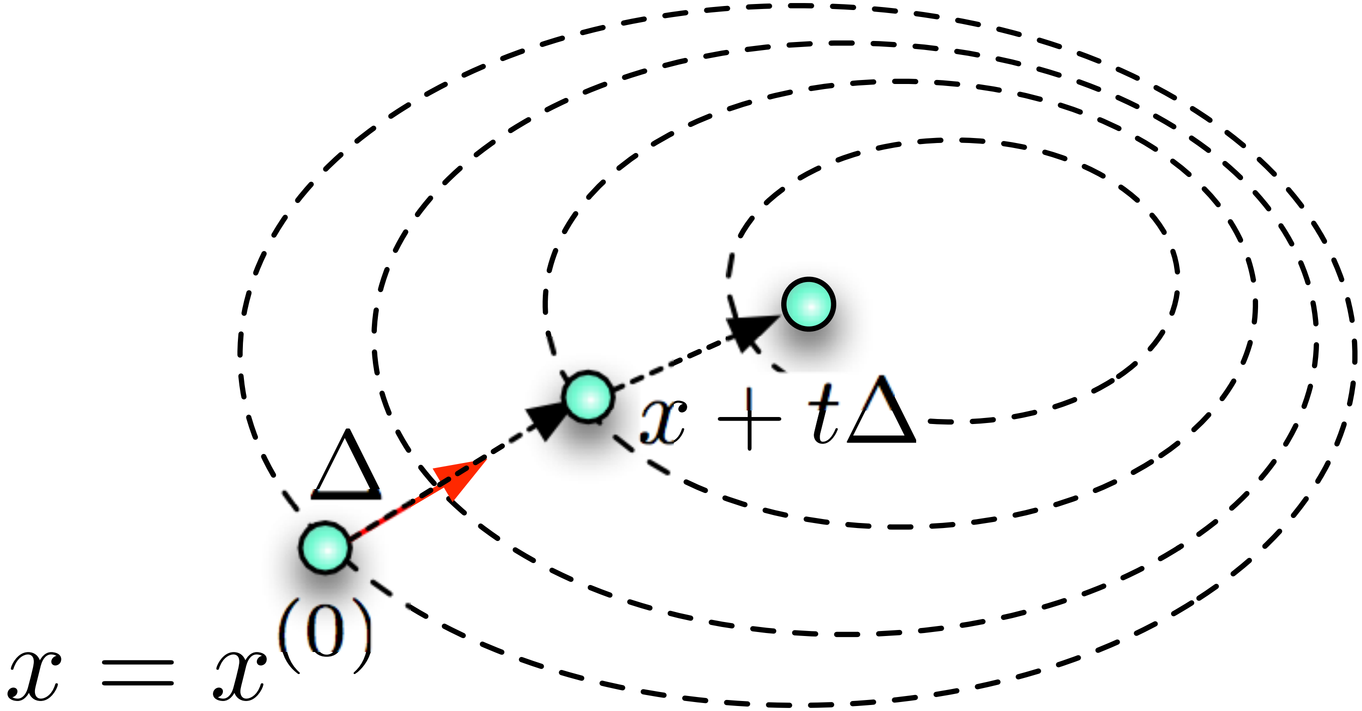
降下法(descent method)の手順

探索点を$x = x^{(0)}$としたのち、下記を繰り返す。

- xにおける探索方向ベクトルΔを求める
- 直線探索を行い、ステップ乗数$t(t > 0)$を求める。
- $x := x + t\Delta$として、探索点を更新する
- 停止条件が満たされた場合に実行終了。

注: 機械学習分野では、直線探索を行わず、定数αに対して$x := x + \alpha\Delta$と更新を行う。

降下法の実行過程



探索方向Δ

第k反復における探索方向を$\Delta^{(k)}$、ステップ乗数を$t^{(k)}$とすると

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta^{(k)}$$

となる。

$$f(x^{(0)}) > f(x^{(1)}) > f(x^{(2)}) > \cdots$$

となるように$\Delta^{(k)}$、$t^{(k)}$を選びたい。

直線探索

正確な直線探索

$$t = \arg\min_{t>0} f(x + t\Delta)$$

- 1次元の目的関数(tの関数)の最小化問題
- 場合によっては解析的に求まる。

勾配法 (gradient descent method)

勾配法は、探索方向Δとして、負勾配ベクトル $-\nabla f(x)$を利用する降下法である。

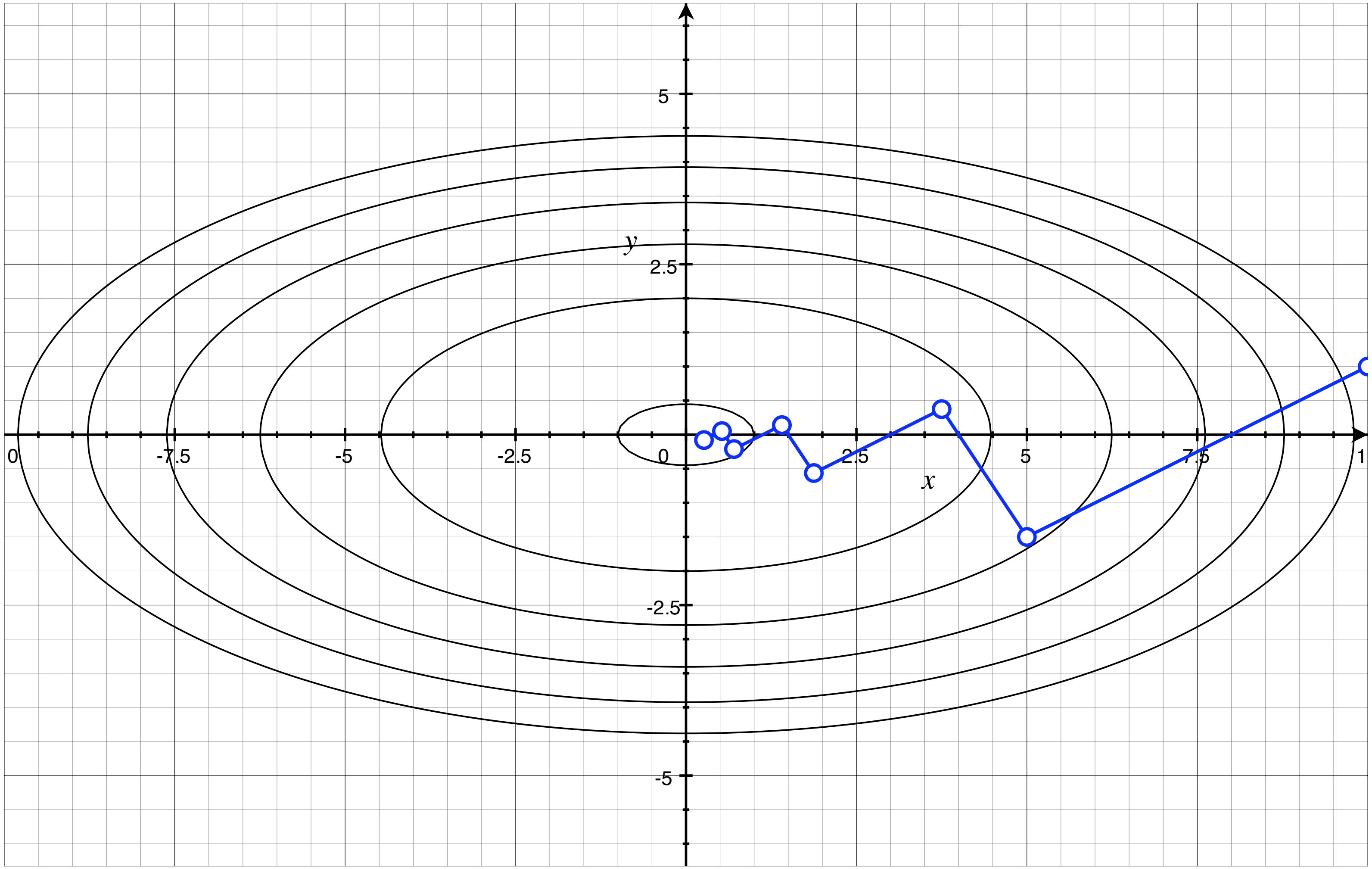
- 最適解への収束性を持つ
- 探索点は等高線に常に直交するように移動する。

勾配法の手順

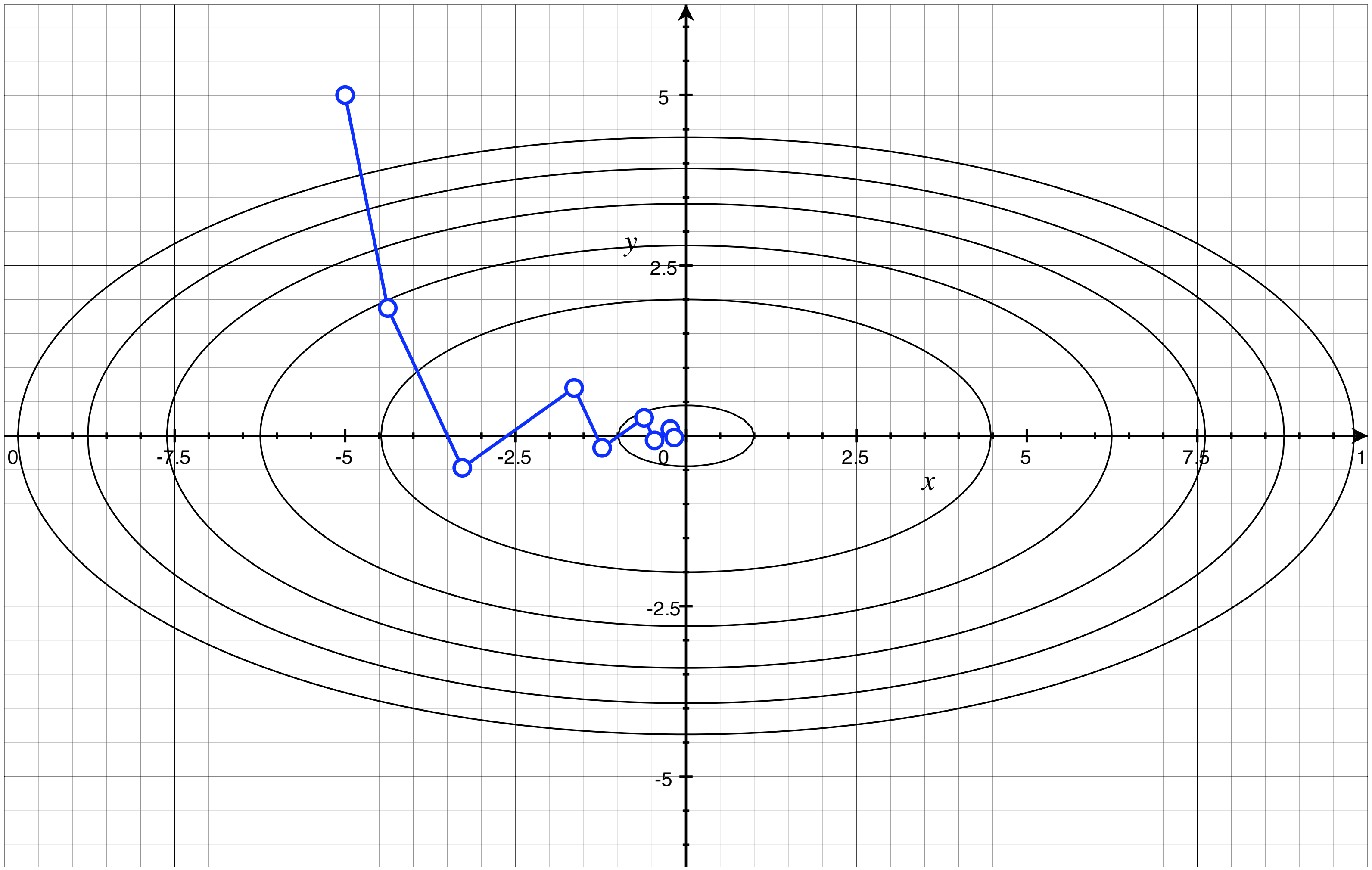
探索点を $x = x^{(0)}$ としたのち、下記を繰り返す。

- $\Delta = -\nabla f(x)$ とする。
- $||\nabla f(x)||_2 < \epsilon$ で実行終了。
- 直線探索を行い、ステップ乗数 $t (t > 0)$ を求める。
- $x := x + t\Delta$ として、探索点を更新する

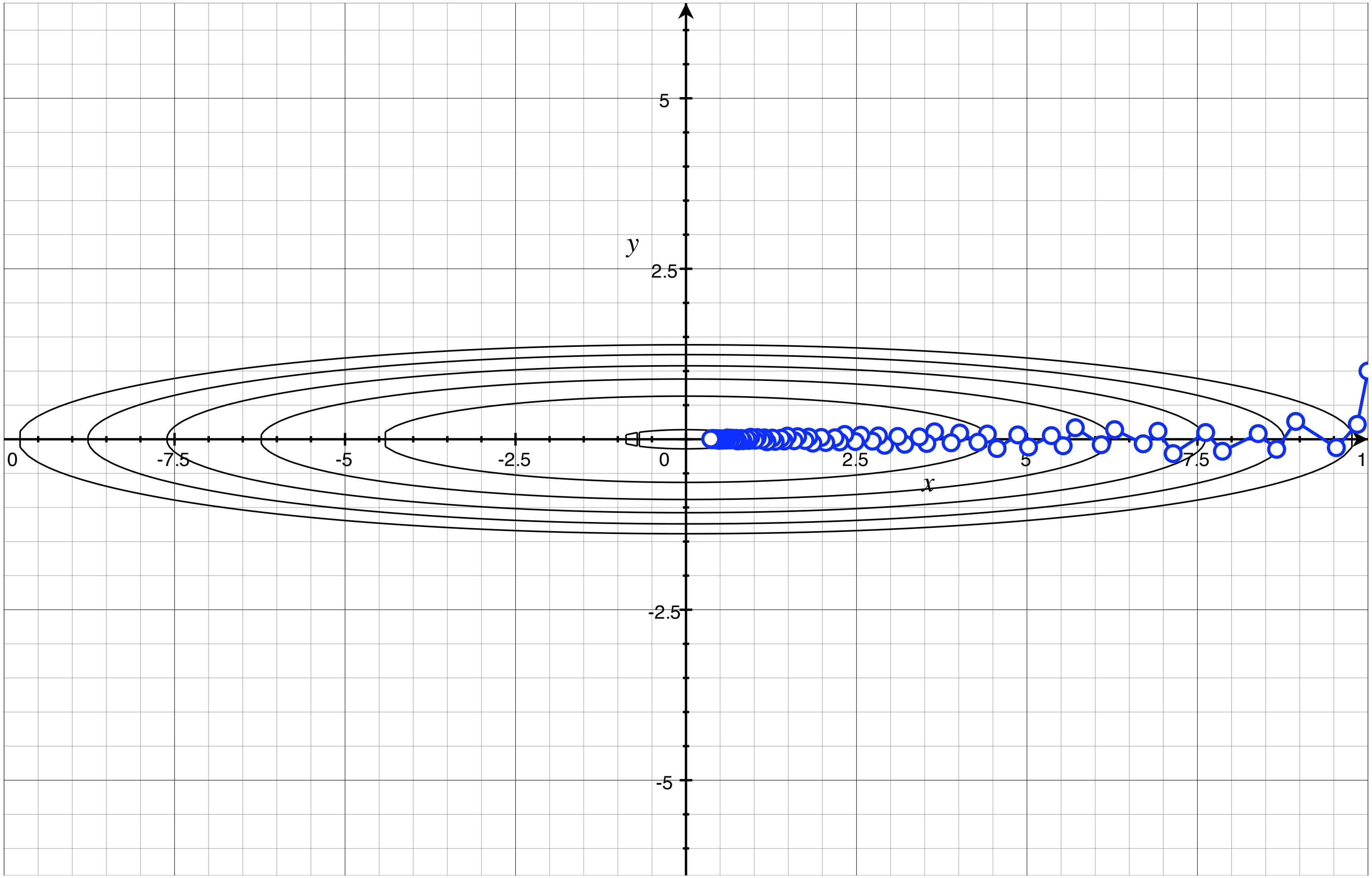
$$f(x) = x_1^2 + \theta x_2^2,\ \ \theta = 5,\ \ x^{(0)} = (10, 1)$$



$$f(x) = x_1^2 + \theta x_2^2,\ \ \theta = 5,\ \ x^{(0)} = (-5, 5)$$



$$f(x) = x_1^2 + \theta x_2^2, \;\; \theta = 50, \;\; x^{(0)} = (10, 1)$$



勾配法の収束速度

- 目的関数の形状によっては、勾配法の収束は非常に遅くなる。
- 勾配法は単純だが、多くの応用では収束が遅くなるケースが多く、実用に耐えない場合もある。
- 収束速度は目的関数のヘッセ行列の “条件数” と深く関係する。

勾配法のまとめ

- 探索方向として負勾配ベクトルを利用する降下法の一種
- 目的関数の形状によっては収束が非常に遅くなる場合がある。
- 一次収束性を持つ(直線探索を行う場合)
- 大局的収束性を持つ(直線探索を行う場合)
- 直線探索を行わない場合は、ステップ係数の設定を慎重に行う必要がある

ペナルティ関数法

- 制約付き最小化問題を解くための最適化手法
- 制約条件に対して、制約充足しない点にペナルティを与える「ペナルティ関数」を利用
- メリット関数は、最適化対象の目的関数とペナルティ関数の和
- 勾配法を利用してメリット関数を最小化する

ペナルティ関数法

メリット関数

$$F(x) = f(x) + tp(x)$$

- 本当に最小化したい対象は $f(x)$
- $p(x)$ がペナルティ関数(制約条件が破れると正の値を取る)
- $F(x)$ を勾配法を利用して最小化する

ペナルティ関数の例

等式制約

$$h(x) = 0 \to p(x) = (h(x))^2$$

不等式制約

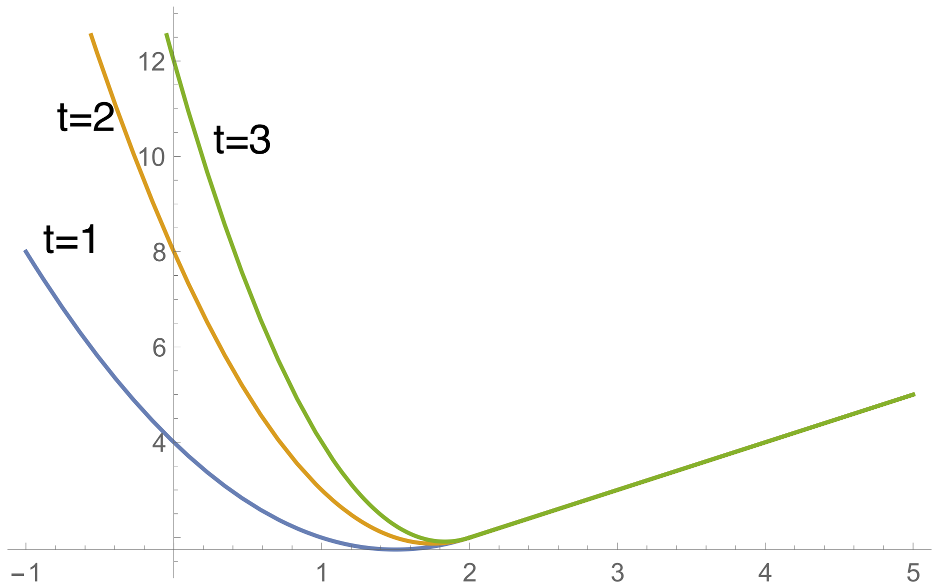
$$g(x) \leq 0 \to p(x) = (\max[0, g(x)])^2$$

ペナルティ関数法の例

次の凸最適化問題を考える:

$$\text{minimize } x \text{ subject to } -x+2 \leq 0$$

$$\rightarrow \text{minimize } x + t\left(\max\{0, -x+2\}\right)^2$$



クイズ

次の凸計画問題をペナルティ関数法を利用して解きたい。その場合のメリット関数(二乗ペナルティ関数を利用)を書け。

$$\text{minimize } x^2 + y^2$$

$$\text{subject to}$$

$$x^2 + y^2 = 1,$$

$$y \geq -x + 1$$

クイズ解答

$$
\begin{aligned}
F(x,y) &= x^2 + y^2 \\
&\quad + t(x^2 + y^2 - 1)^2 \\
&\quad + (\max\{0, -x - y + 1\})^2
\end{aligned}
$$

パラメータtの選択について

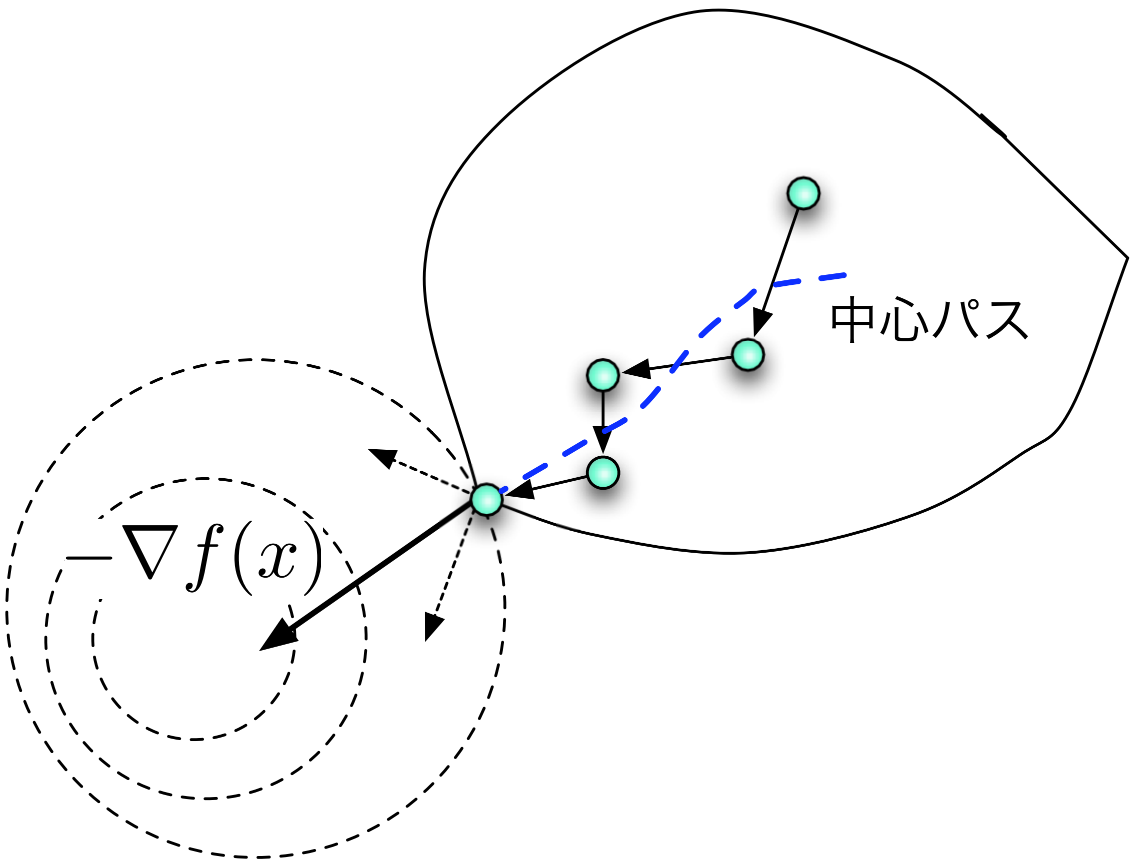
- メリット関数を最小化した結果が真の目的関数の最小化の結果と近くなるためには、tは十分に大きい必要がある
- しかしtが大きいと「悪条件問題」となり、勾配法の収束は非常に遅くなる
- tは小さい値からスタートし、次第に大きい値に更新していく手法がよく利用される(例えば、$t_i = \gamma t_{i-1}$)

内点法(interior point method)

内点法は、凸計画問題を解くための最適化アルゴリズムである。

注：ニュートン法、勾配法は微分可能な目的関数 f について $\nabla f(x) = 0$ となる点を見つけるアルゴリズムである。凸計画問題の解は必ずしも $\nabla f(x) = 0$ を満たす点とは限らない!

内点法の考え方



仮定

等式条件のない凸計画問題を考える。

$f_0, f_1, \ldots, f_m$ **は凸関数かつ微分可能。**

$$\begin{aligned} &\text{minimize} \ f_0(x) \\ &\text{subject to} \ f_i(x) \leq 0, \quad i = 1, 2, \ldots, m \end{aligned}$$

(等式条件がある場合にも容易に拡張できる。)

問題の変換

”形式的に”無制約形として書き直す。

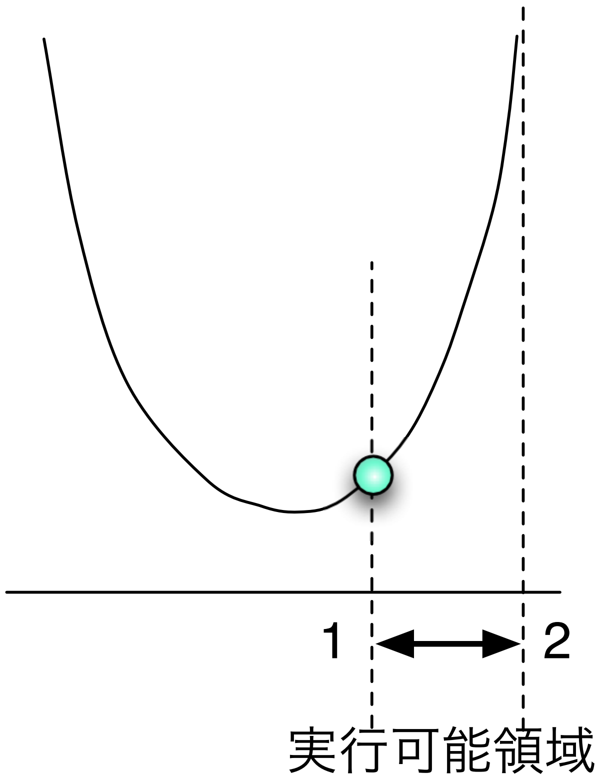
$$\text{minimize} \;\; f_0(x) + \sum_{i=1}^{m} I_-(f_i(x))$$

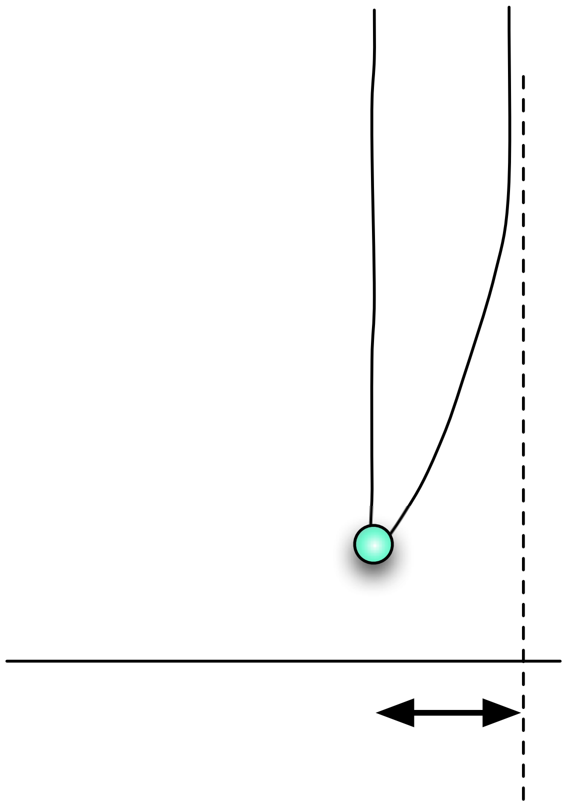
ここで、

$$I_-(u) = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0 \end{cases}$$

$$\begin{aligned} \text{minimize } & x^2 \\ \text{subject to } & \\ 1 - x &\leq 0 \\ x - 2 &\leq 0 \end{aligned}$$

$$\begin{aligned} \text{minimize } x^2 \quad &+ \quad I_-(1-x) \\ &+ \quad I_-(x-2) \end{aligned}$$

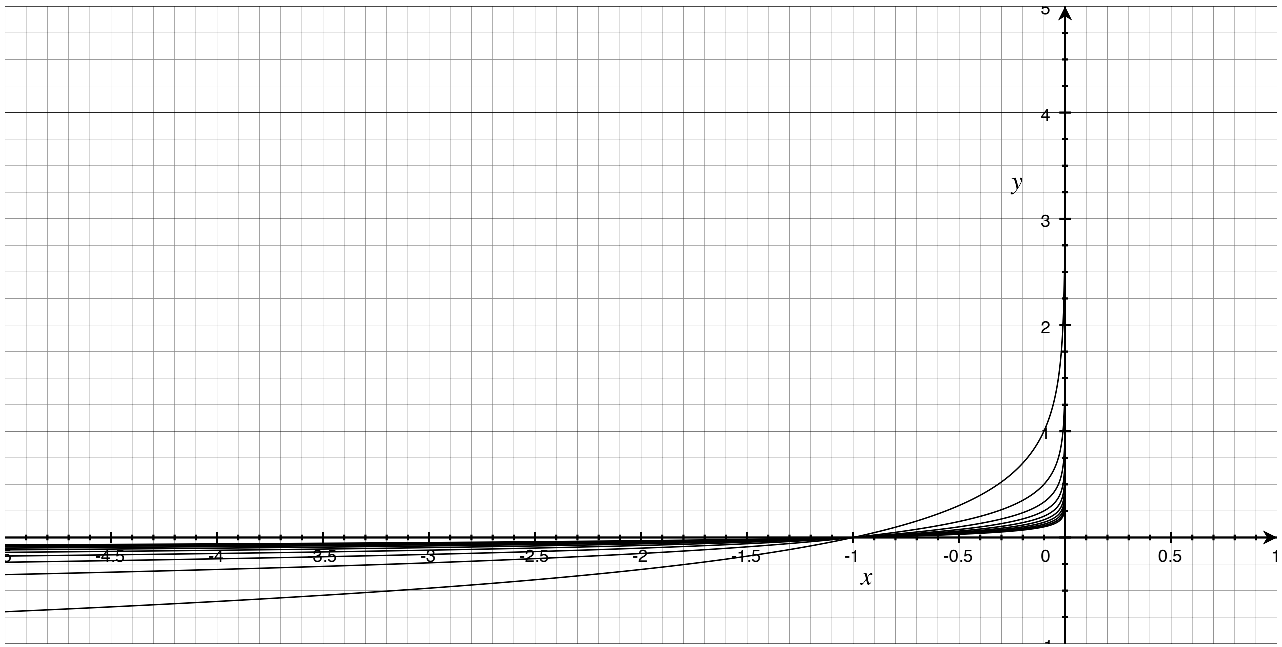




対数バリア関数(log-barrier function)

$I(u)$ を対数バリア関数で近似 $(t > 0)$:

$$b(u) = -\frac{1}{t}\ln(-u)$$



$t = 1, 2, \ldots, 10$

近似無制約化問題

$$\text{minimize } f_0(x) - \frac{1}{t}\sum_{i=1}^{m} \ln(-f_i(x))$$

- **目的関数は微分可能かつ凸関数**
- $t \to \infty$ **で無制約形に漸近**

メリット関数：

$$\psi^{(t)}(x) = tf_0(x) - \sum_{i=1}^{m} \ln(-f_i(x))$$

中心点と中心パス

中心点

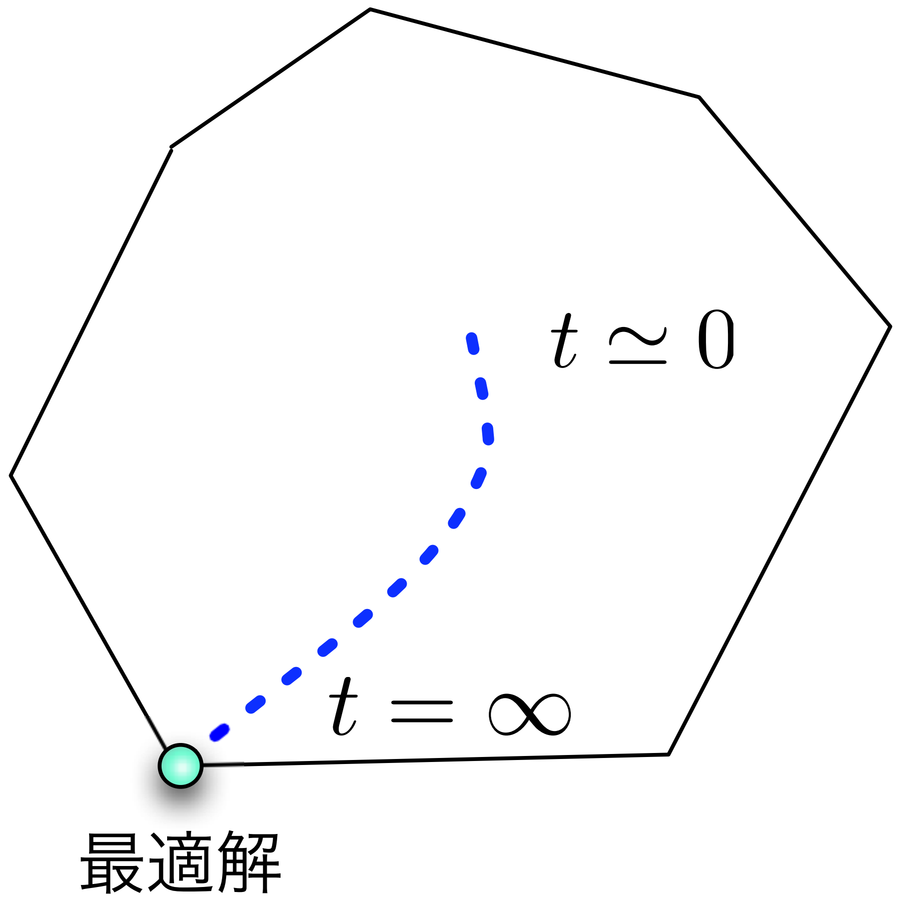
$$x^*(t) = \arg\min_{x \in F} \psi^{(t)}(x)$$

F は実行可能領域。

中心パス(central path)

$$\{x^*(t) : t > 0\}$$

中心パスの例



主パス追跡内点法の手順

$x_0 \in F^*$ (F^* は実行可能領域の境界以外の部分)、
$t = t_0 > 0$、$\mu > 0$、$\epsilon > 0$

$x = x_0$ としたのち、以下を反復する。

1. センタリング：初期点を x とし、ニュートン法を利用して $x^*(t)$ を求める。
2. 探索点更新：$x = x^*(t)$ とする。
3. 停止条件：$m/t < \epsilon$ が成り立てば終了。
4. t の更新：$t := \mu t$

主パスの追跡