Oct. 25, 2007

乱択近似
(randomized approximation)

1. 離散アルゴリズム

1-1. MCMC法における近似精度保証

マルコフ連鎖モンテカルロ法
(Markov chain **Monte Carlo**)

*来嶋 秀治
京都大学 数理解析研究所

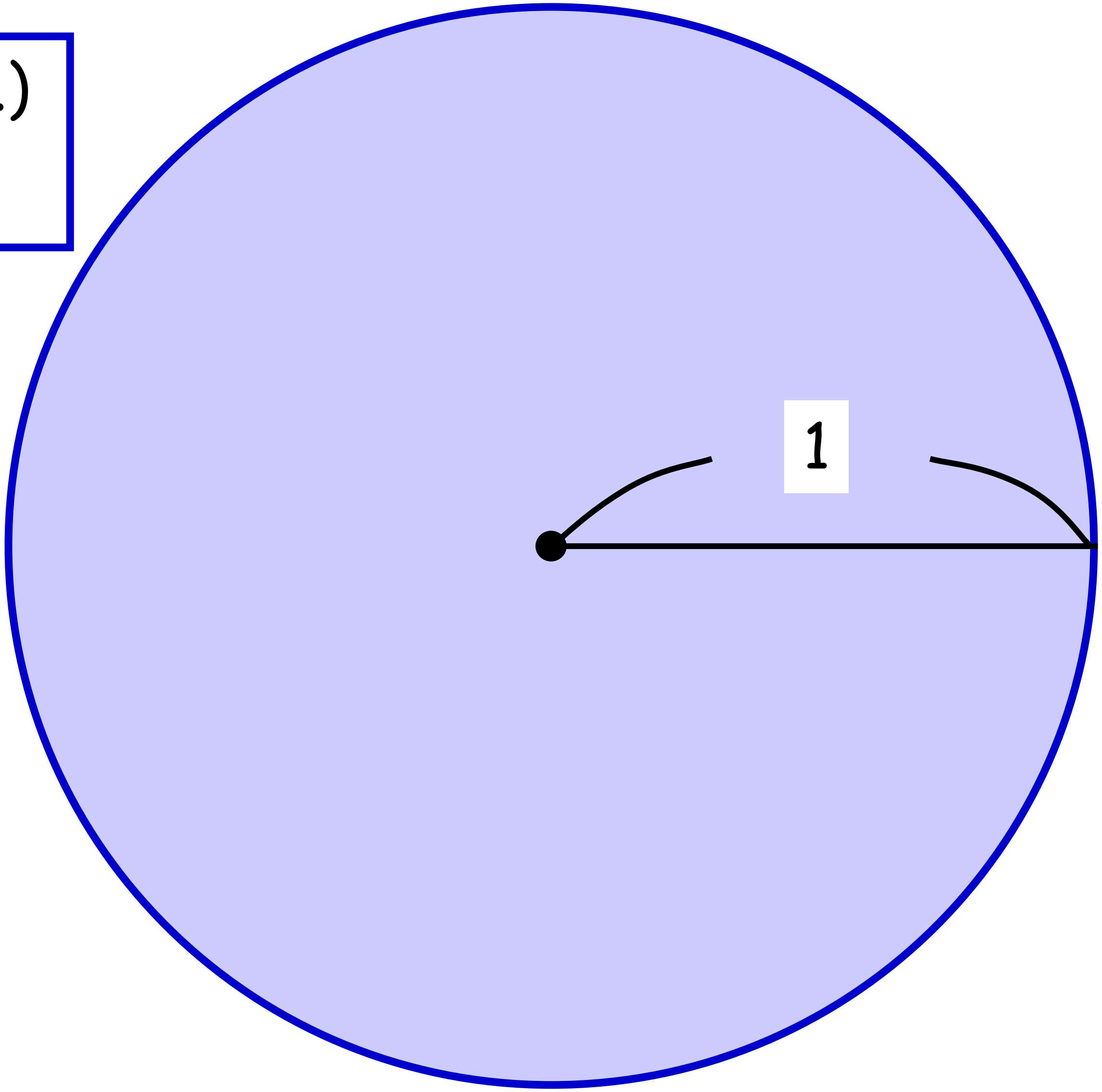
1: イントロダクション

モンテカルロ法とは

モンテカルロ法 - 円周率の計算を例に

円周率 π の値 (=3.141...)
を求めよ。

π = (半径1の円の面積)

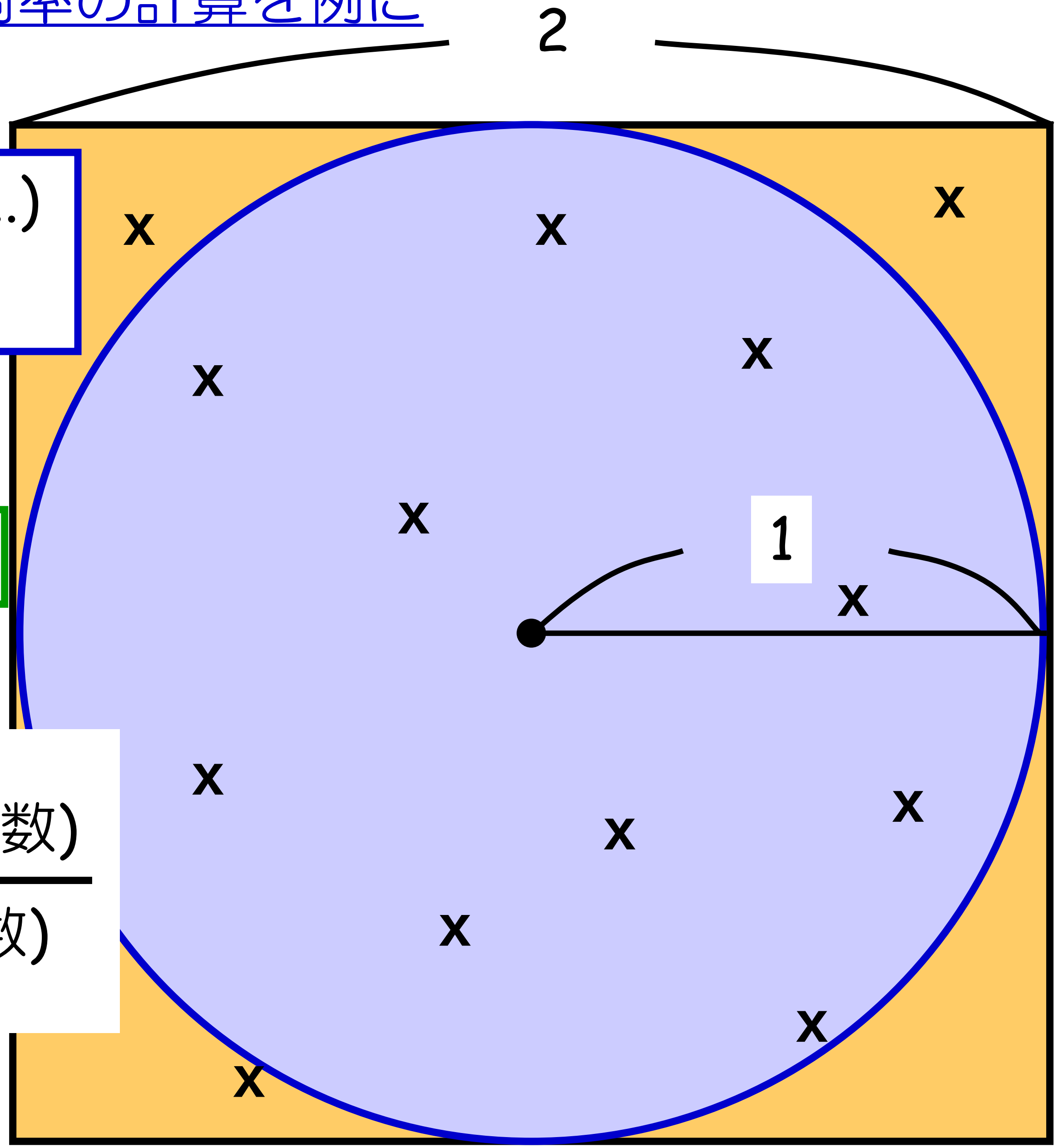


モンテカルロ法 - 円周率の計算を例に

円周率 π の値 (=3.141...)

を求めよ.

$$\pi = (\text{半径1の円の面積})$$



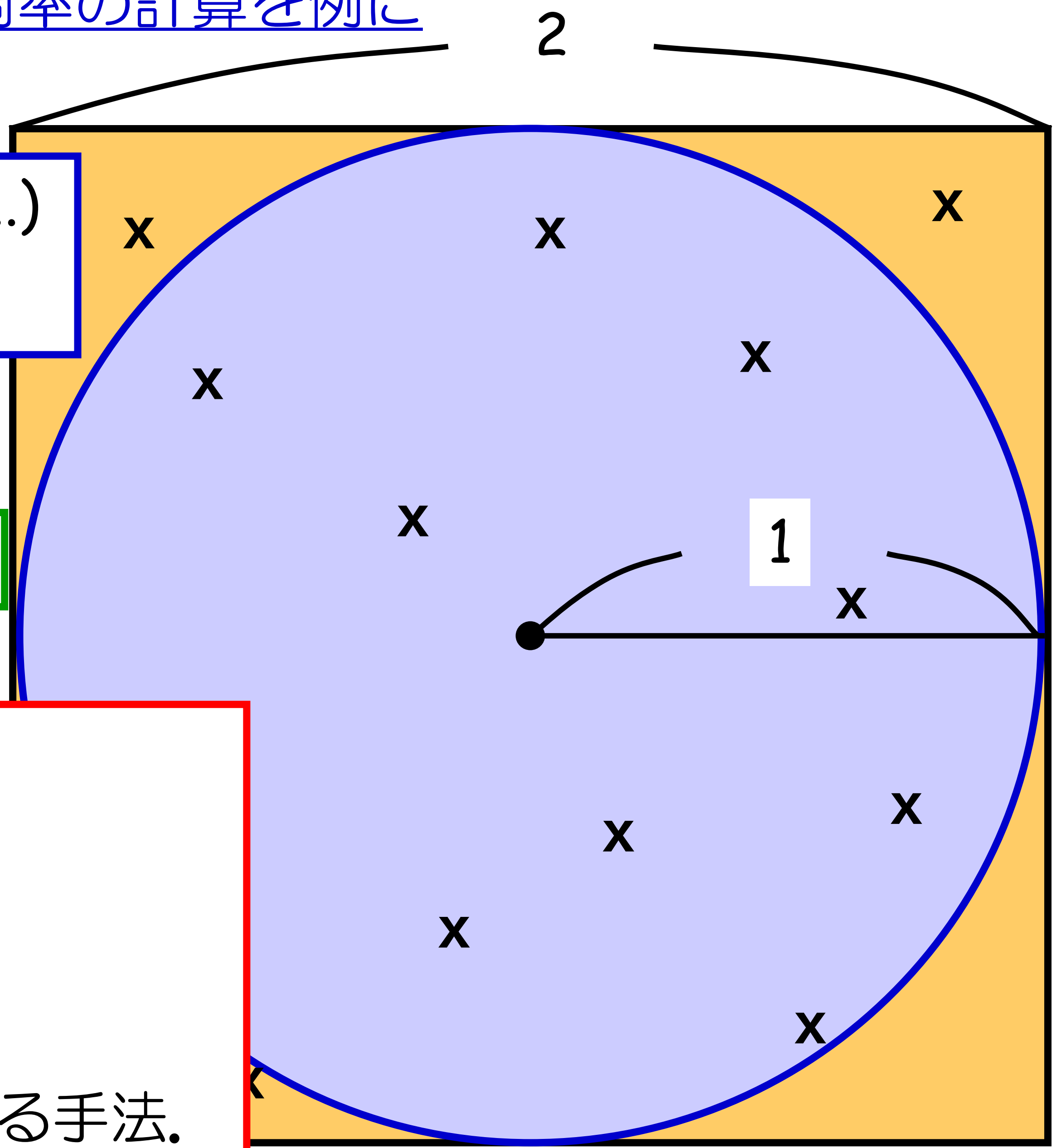
$$\pi = 4 \times \frac{(\text{円の中の点の数})}{(\text{打った点の数})}$$

モンテカルロ法 - 円周率の計算を例に

円周率 π の値 (=3.141...)

を求めよ.

π = (半径1の円の面積)



モンテカルロ法

乱数を使用して,

- 比
- 平均

などを近似的に計算する手法.

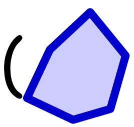
2次元領域の面積計算

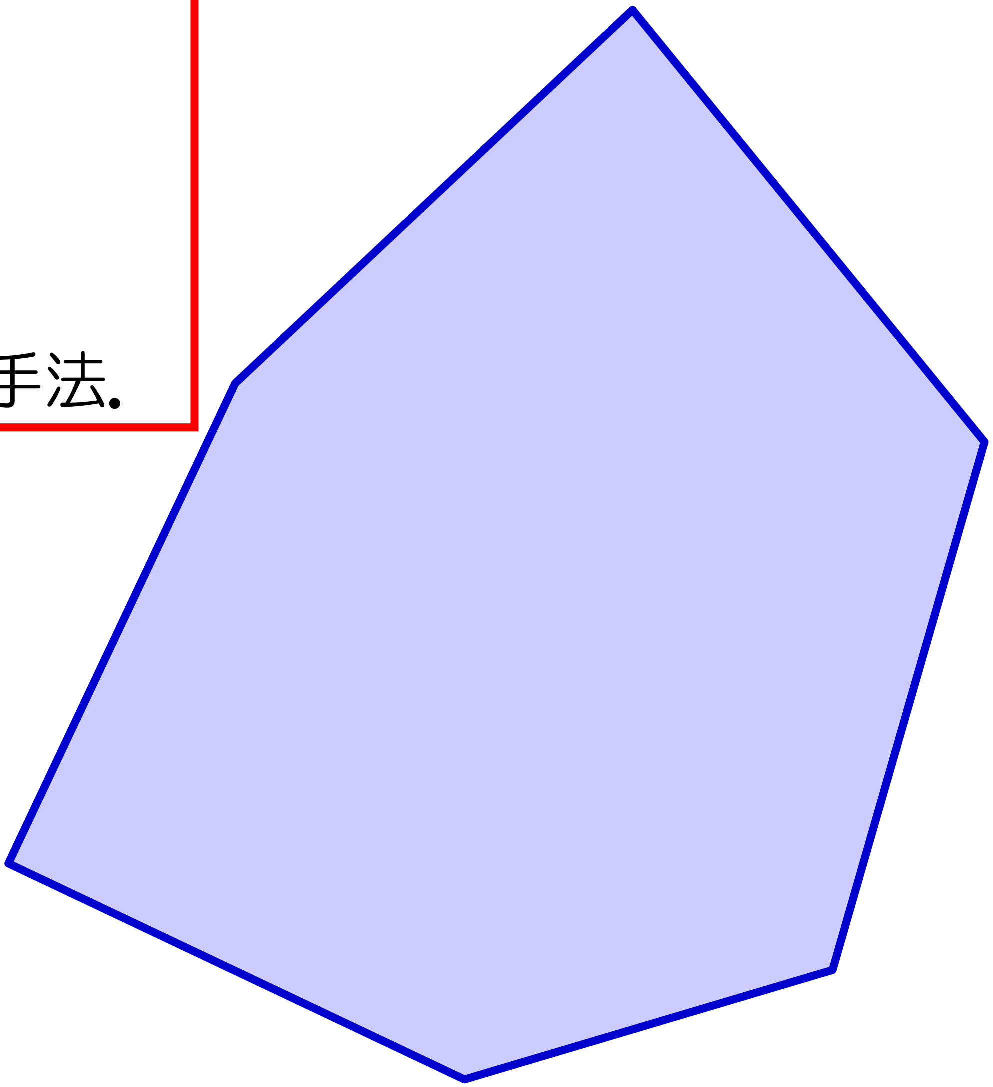
モンテカルロ法

乱数を使用して,

- 比
- 平均

などを近似的に計算する手法.

( の面積) =



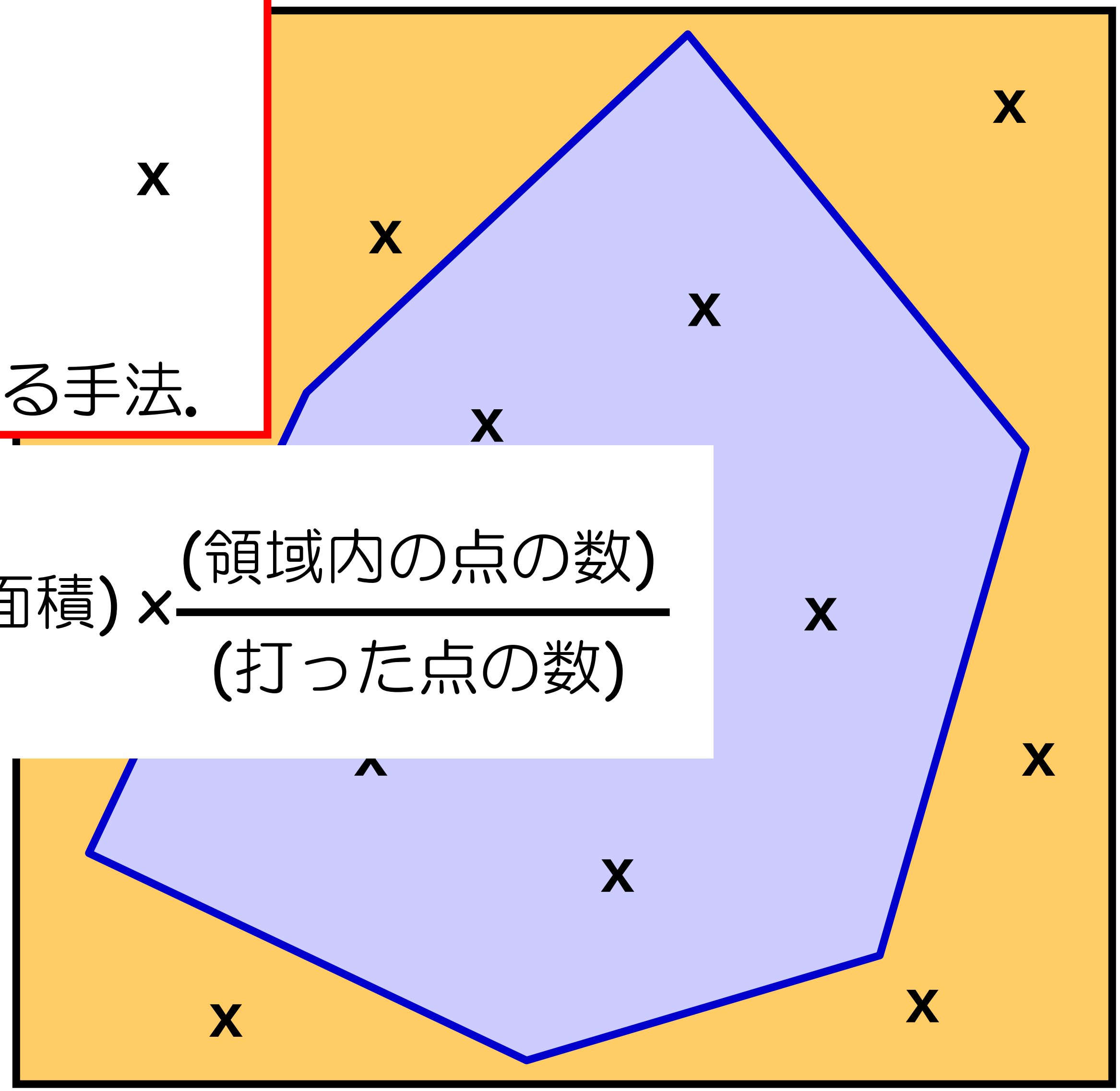
2次元領域の面積計算

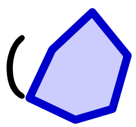
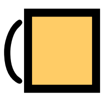
モンテカルロ法

乱数を使用して,

- 比
- 平均

などを近似的に計算する手法.



( の面積) = ( の面積) × $\dfrac{(\text{領域内の点の数})}{(\text{打った点の数})}$

2次元領域の面積計算

モンテカルロ法

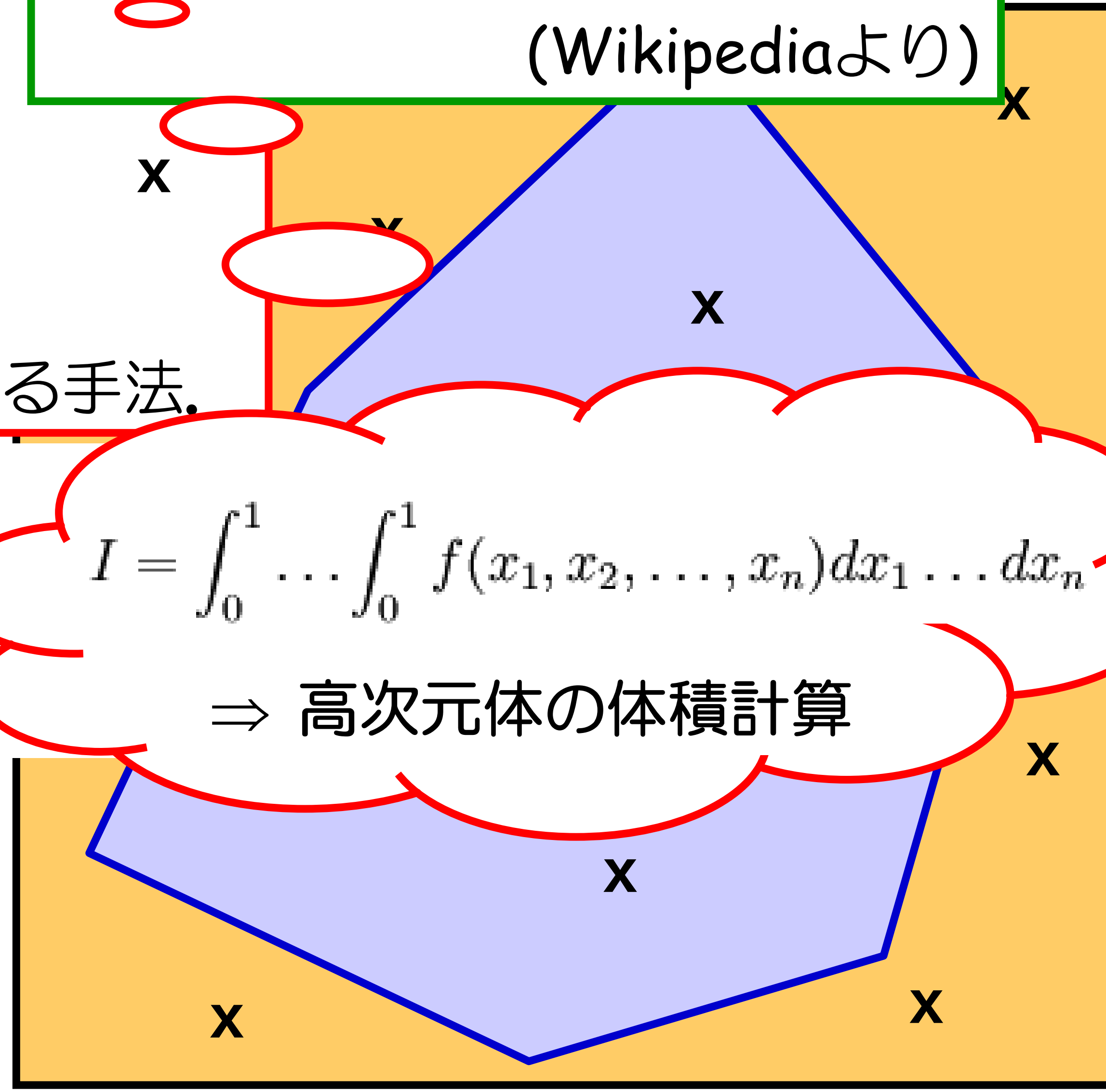
乱数を使用して,

- 比
- 平均

などを近似的に計算する手法.

重積分に強いという特徴がある.

(Wikipediaより)



$$I = \int_0^1 \cdots \int_0^1 f(x_1, x_2, \ldots, x_n) dx_1 \ldots dx_n$$

$\Rightarrow$ 高次元体の体積計算

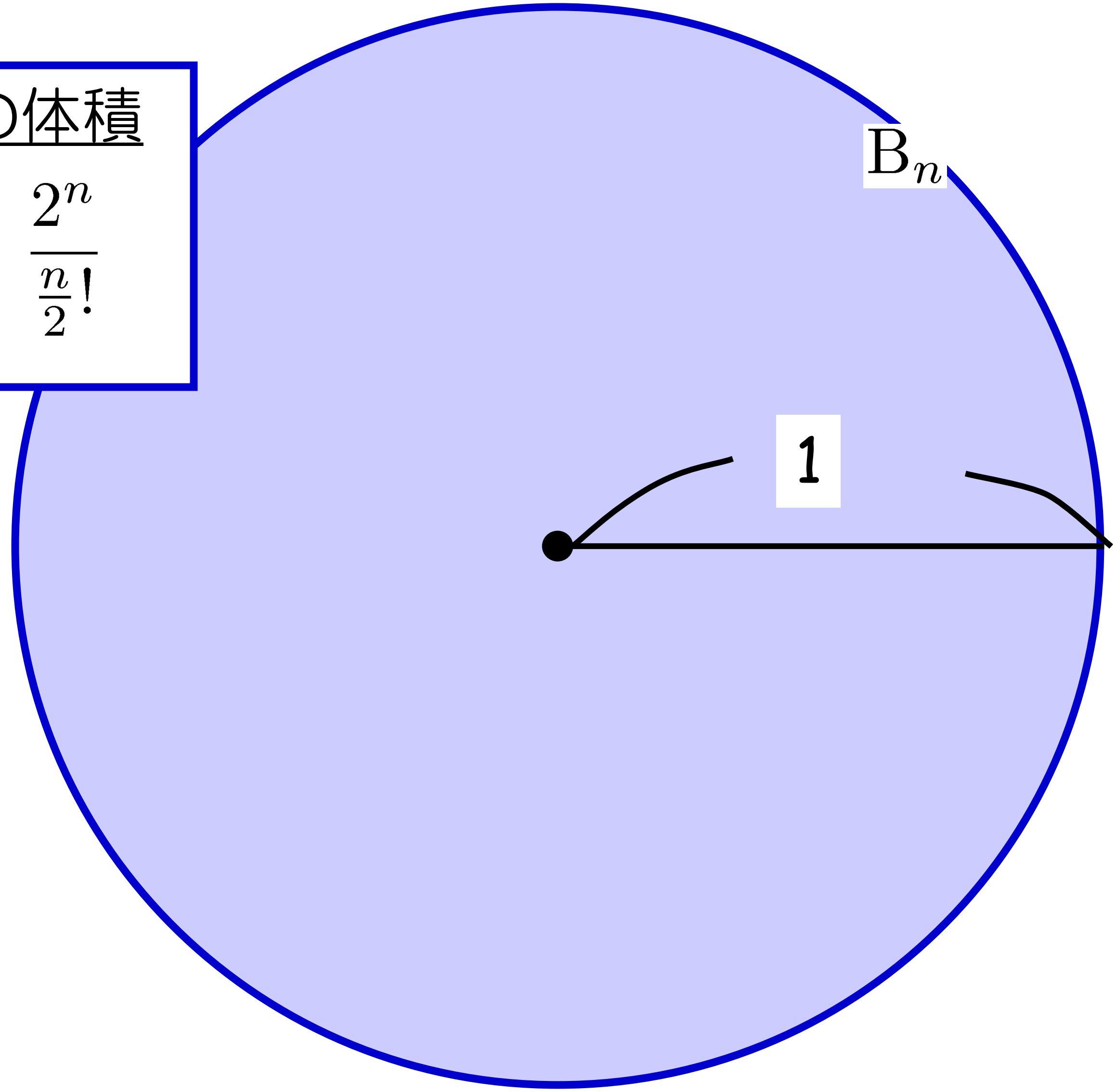
(の面積) = (の

高次元の呪い - 球を例に

n次元球 B_n の体積

n次元ユークリッド球の体積

$$\mathrm{V}(\mathrm{B}_n) = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} < \frac{2^n}{\frac{n}{2}!}$$



高次元の呪い - 球を例に

n次元球 B_n の体積

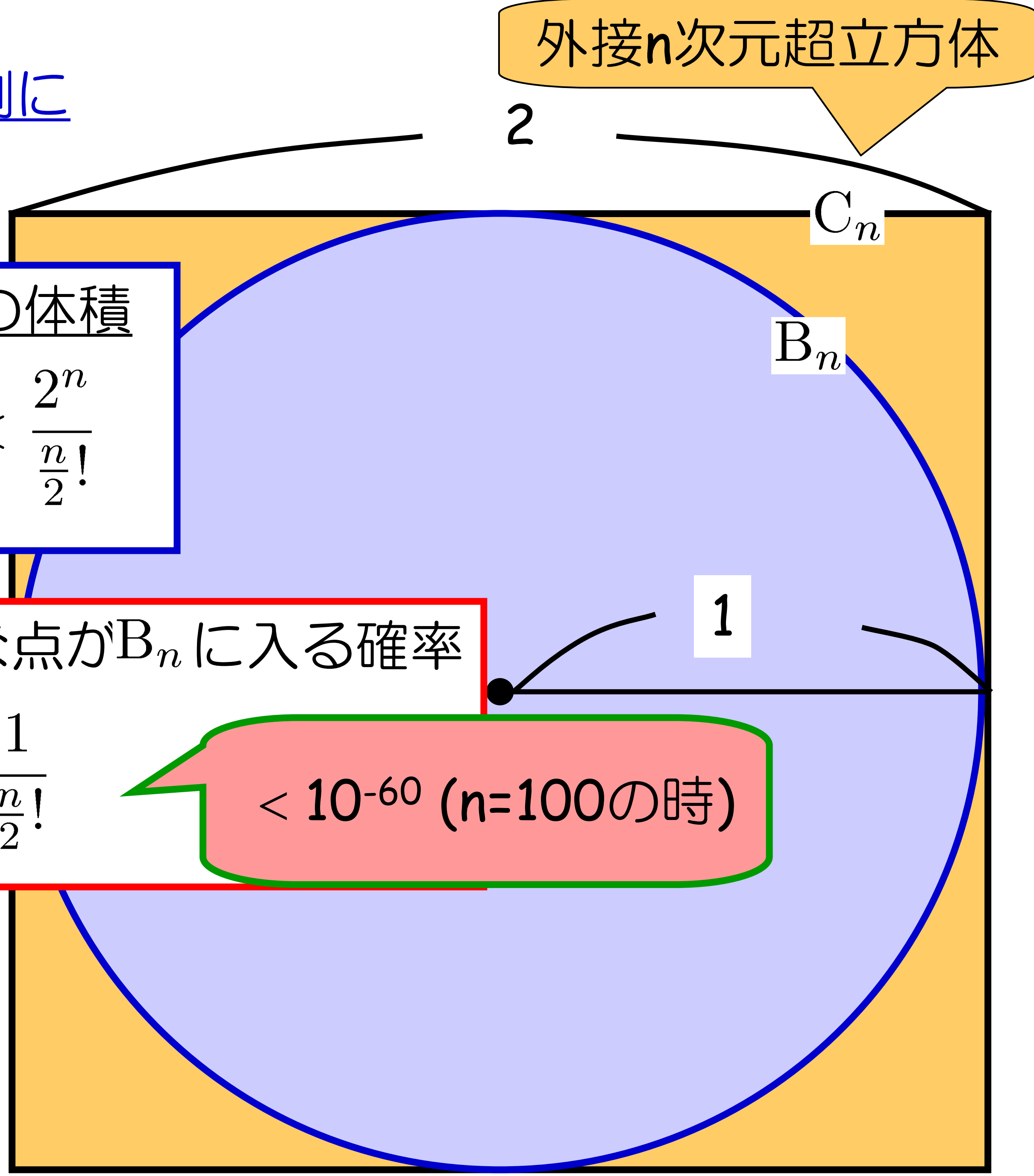
n次元ユークリッド球の体積

$$\mathrm{V}(\mathrm{B}_n) = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} < \frac{2^n}{\frac{n}{2}!}$$

C_n 上の一様ランダムな点がB_n に入る確率

$$\frac{\mathrm{V}(\mathrm{B}_n)}{\mathrm{V}(\mathrm{C}_n)} = \frac{\mathrm{V}(\mathrm{B}_n)}{2^n} < \frac{1}{\frac{n}{2}!}$$

$< 10^{-60}$ (n=100の時)



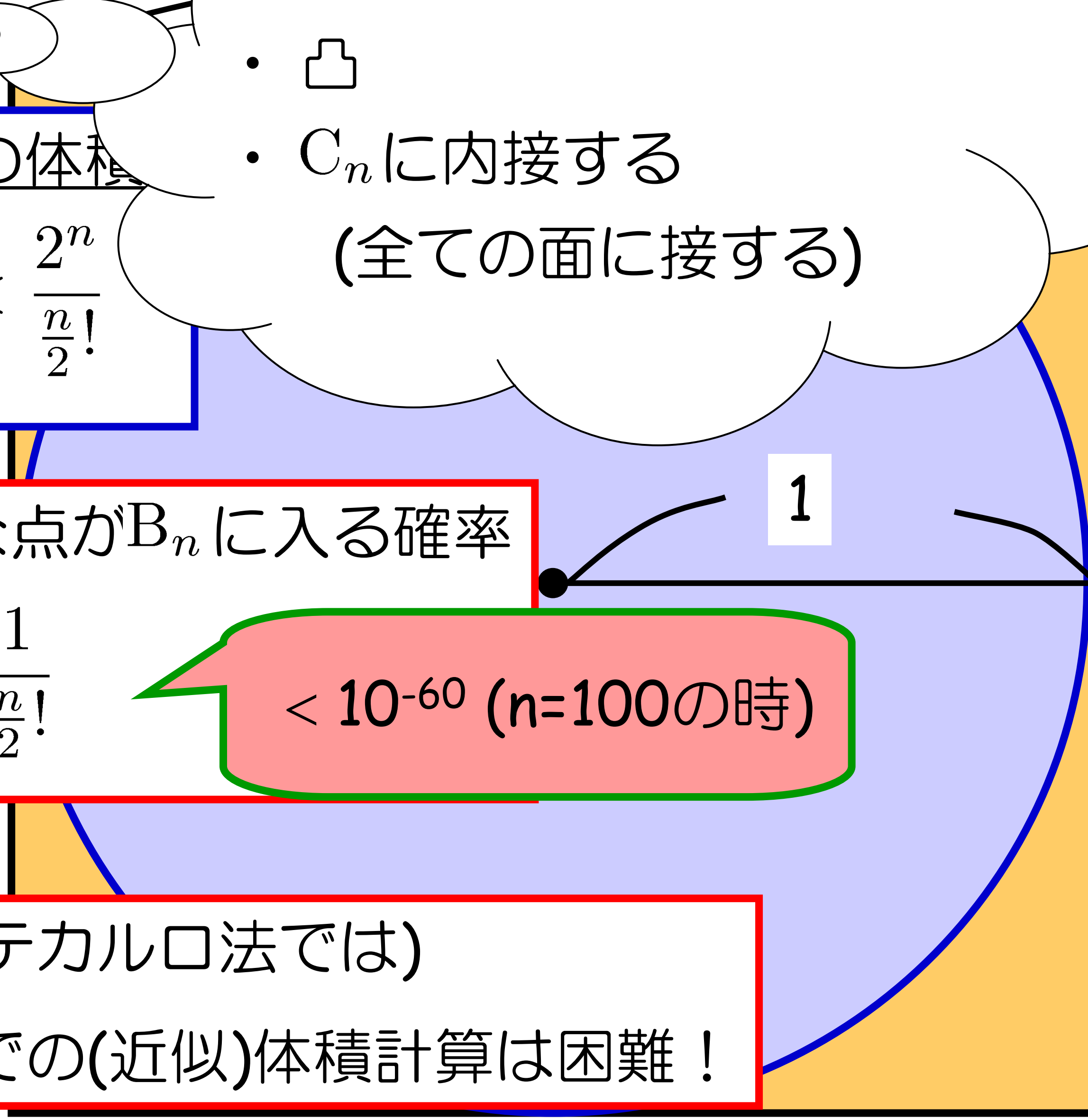
高次元の呪い - 球を例に

n次元球 B_n の体積

n次元ユークリッド球の体積

$$\mathrm{V}(\mathrm{B}_n) = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} < \frac{2^n}{\frac{n}{2}!}$$

⇒ ふっくらした高次元体
- 凸
- C_nに内接する
  (全ての面に接する)



C_n上の一様ランダムな点がB_nに入る確率

$$\frac{\mathrm{V}(\mathrm{B}_n)}{\mathrm{V}(\mathrm{C}_n)} = \frac{\mathrm{V}(\mathrm{B}_n)}{2^n} < \frac{1}{\frac{n}{2}!}$$

$< 10^{-60}$ (n=100の時)

⇒ (素朴なモンテカルロ法では)

多項式時間での(近似)体積計算は困難！

数え上げ，積分の計算困難性

n次元凸体の体積の計算 => #P困難

#P: 数え上げ問題に対する計算量クラス.

#P困難 ⊂ NP困難

乱択近似
(randomized approximation)

FPRAS (Fully Polynomial-time Randomized Approximation Scheme)
(全多項式時間乱択近似計算法)

cf. FPTAS

A: 真の値

Z: 近似値

$$\Pr\left[\frac{|Z-A|}{A} \leq \varepsilon\right] \geq 1-\delta$$

MCMC法による解決

マルコフ連鎖を用いたランダム生成法 + モンテカルロ法

MCMC法における近似精度保証 - Table of Contents

<u>MCMC法</u>

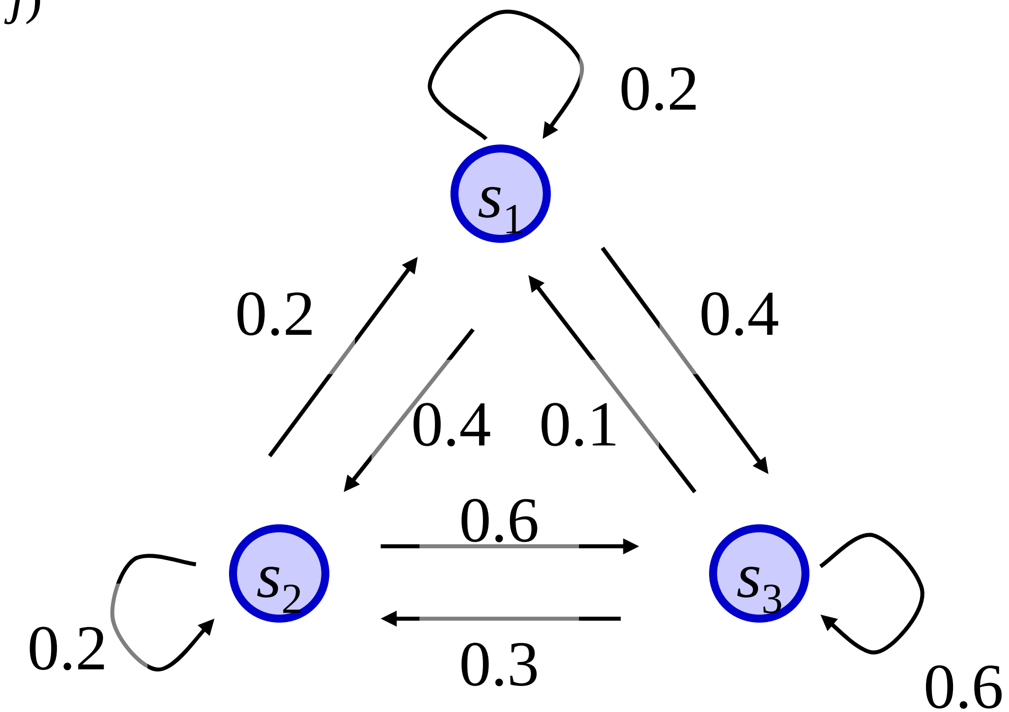
マルコフ連鎖を用いたランダム生成法 + モンテカルロ法

2: マルコフ連鎖を用いたランダム生成法

マルコフ連鎖の定常分布

- マルコフ連鎖 M（エルゴード的）
  - 状態空間 : Ω（有限）
  - 推移規則 : 推移確率行列 P によって推移する

$$P_{ij} = \Pr(\text{推移 } i \to j) = P(i, j)$$



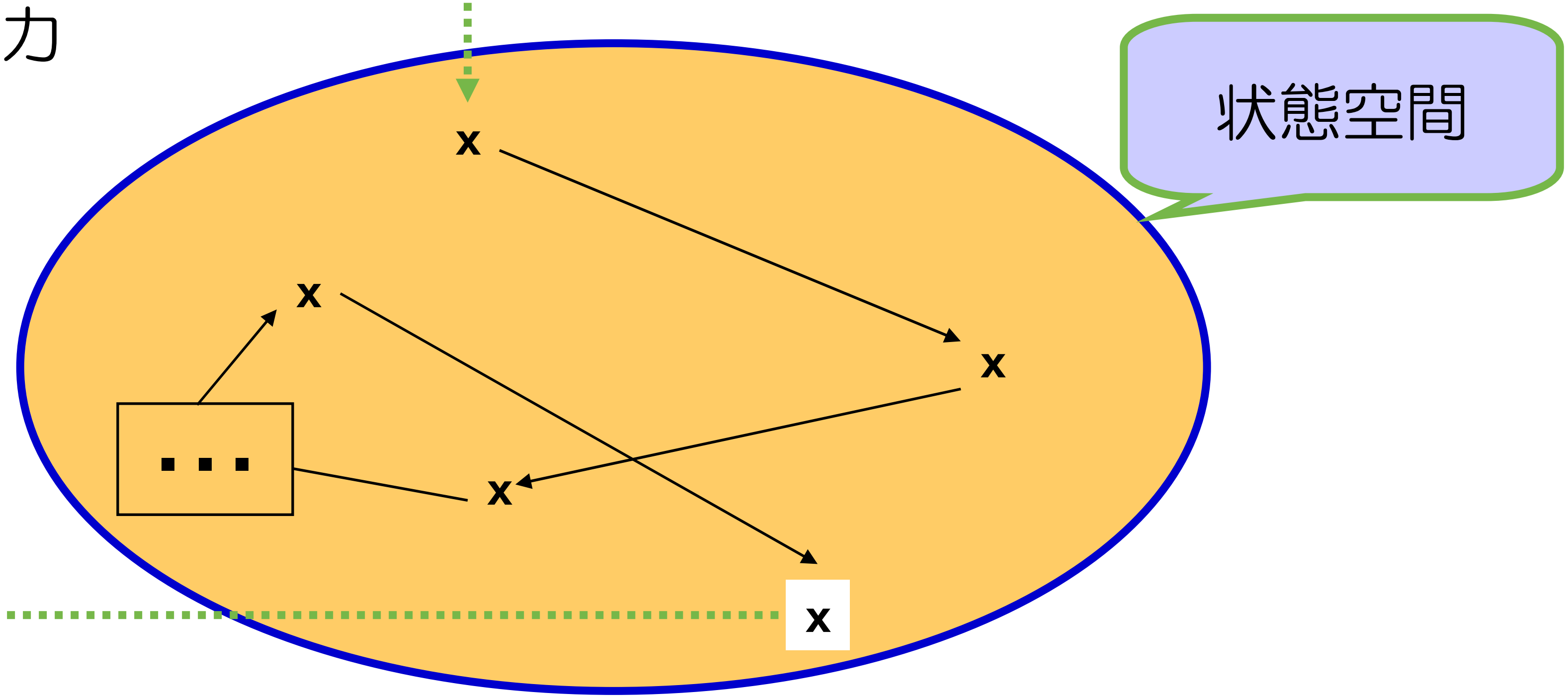
状態遷移図

マルコフ連鎖の極限分布

任意の初期状態

推移を繰り返す

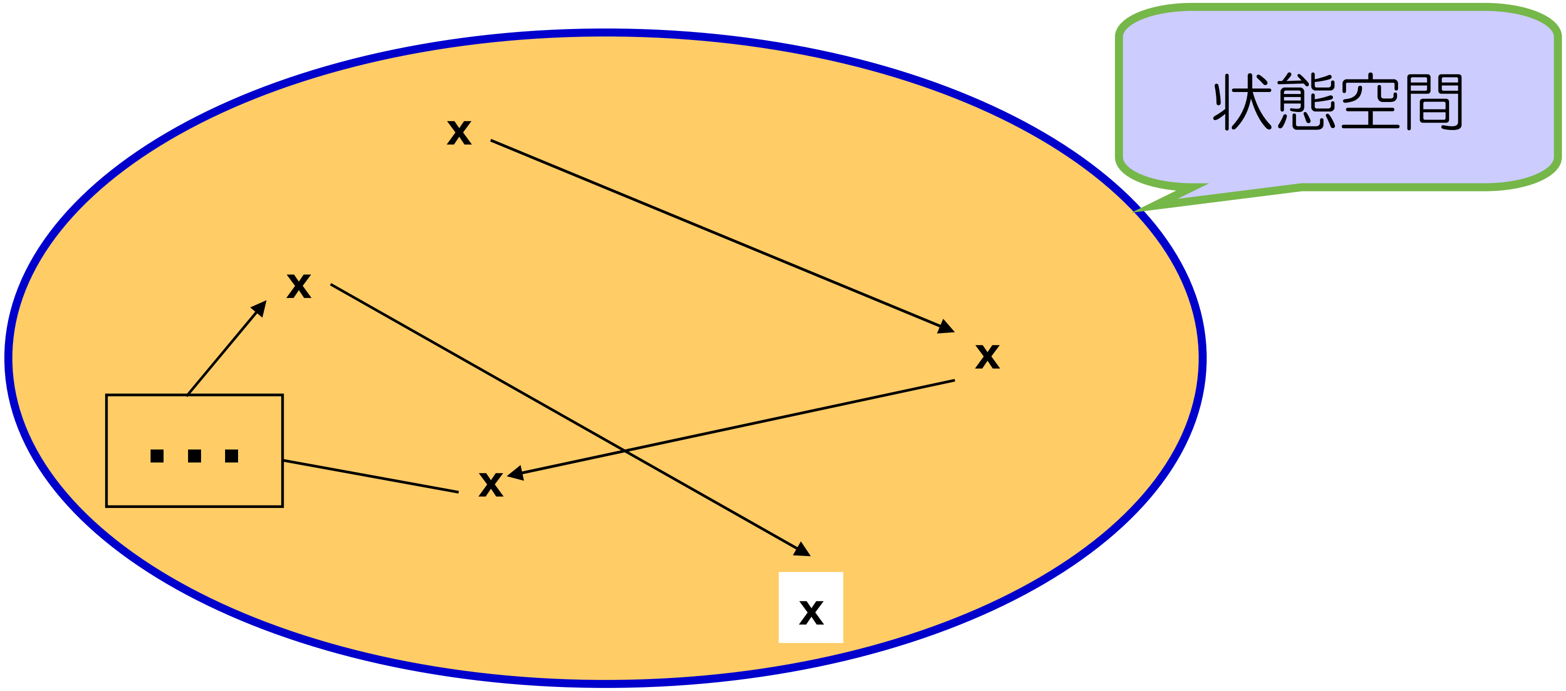
状態を出力



漸近的に定常分布

「マルコフ連鎖を用いたサンプリング法」のアイデア

1. 目的の分布を定常分布にもつマルコフ連鎖を設計する。
2. 十分な回数推移させて、定常分布からサンプリングする。



漸近的に定常分布

MCMC法における近似精度保証 - Table of Contents

3: 高次元凸体の体積計算

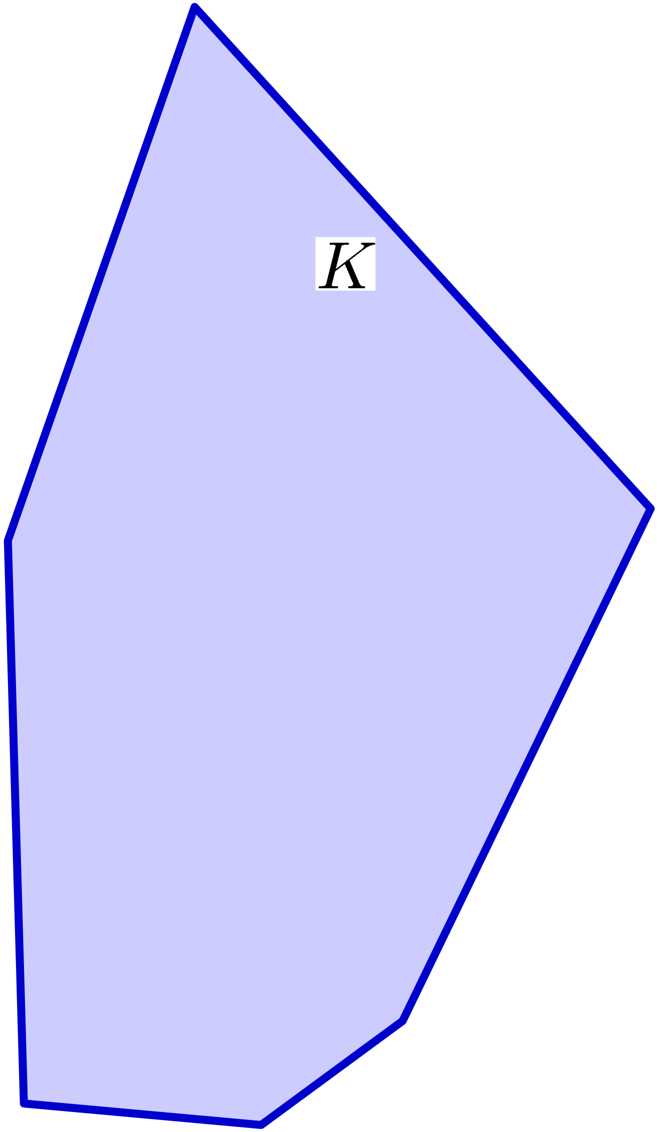
[Dyer, Frieze, Kannan 1991]

再帰的なモンテカルロ法

問題: 高次元凸体の体積計算

K: n次元凸体

体積V(K)を計算



仮定: ふっくらとした凸体

K: n次元凸体

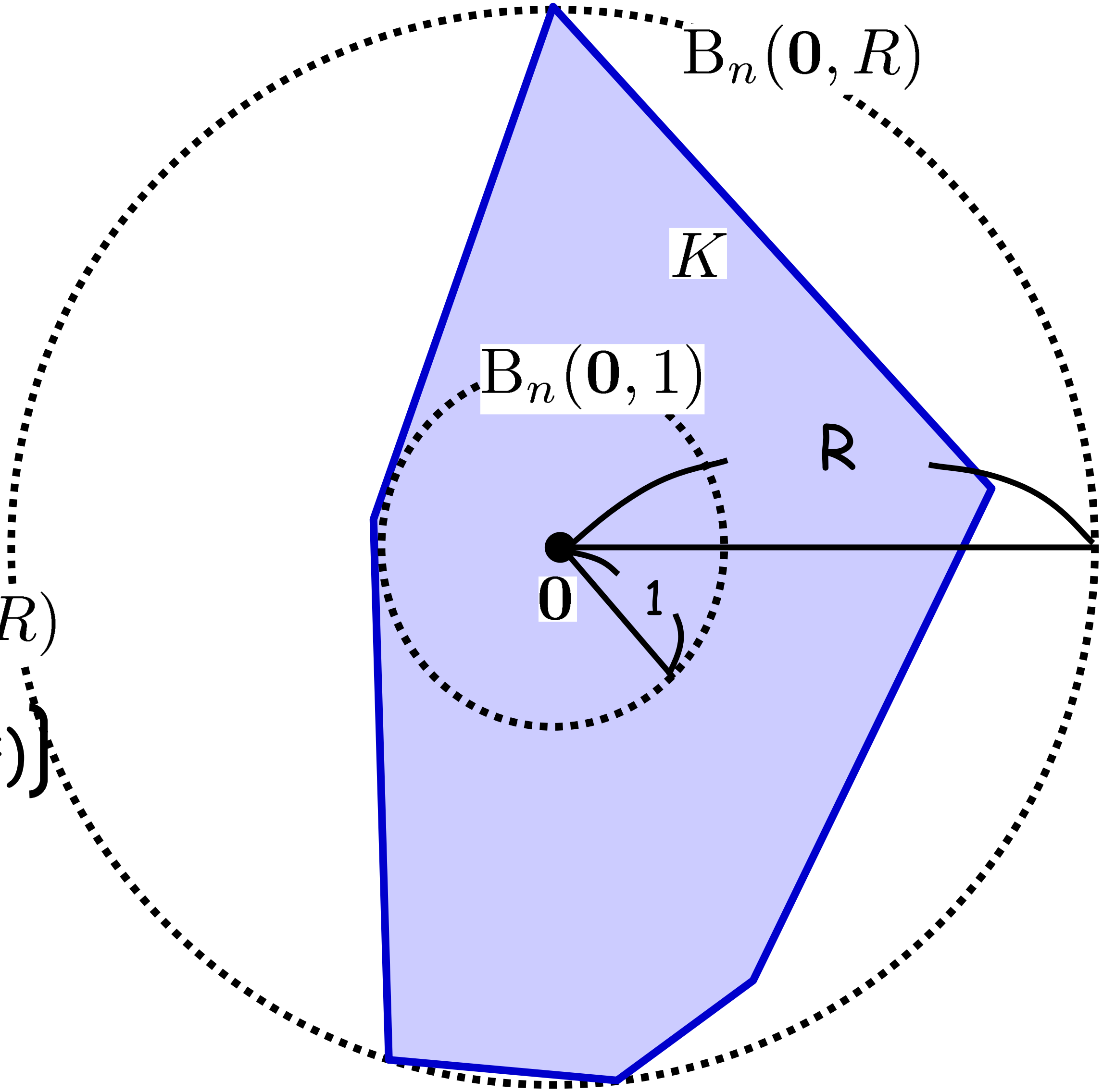
体積V(K)を計算

仮定

Kはふっくらとしている

$$\mathrm{B}_n(\mathbf{0}, R) \subseteq K \subseteq \mathrm{B}_n(\mathbf{0}, R)$$

$[R = O(n^c)]$



[Kannan, Lovasz, Simonovitz 1993]

K上の一様ランダム生成 - Ball walk

Bal-walk

X: 現在の点

X': 次の時刻の点

Step 1.

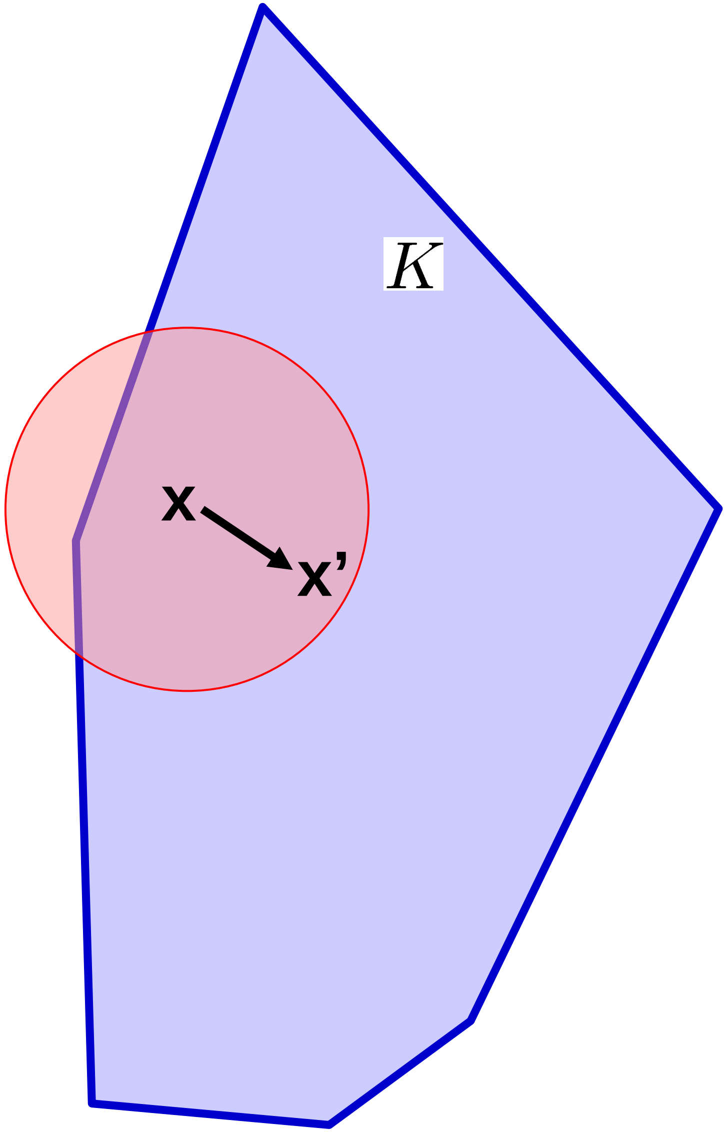
Y : $\mathrm{B}_n(X, a)$ 中の

一様ランダムな点を選ぶ.

Step 2.

$$X' = \begin{cases} Y & (Y \in K) \\ X & (Y \notin K) \end{cases}$$

定常分布はK上の一様分布



K上の一様ランダム生成

(ふっくらとした凸体に対して)

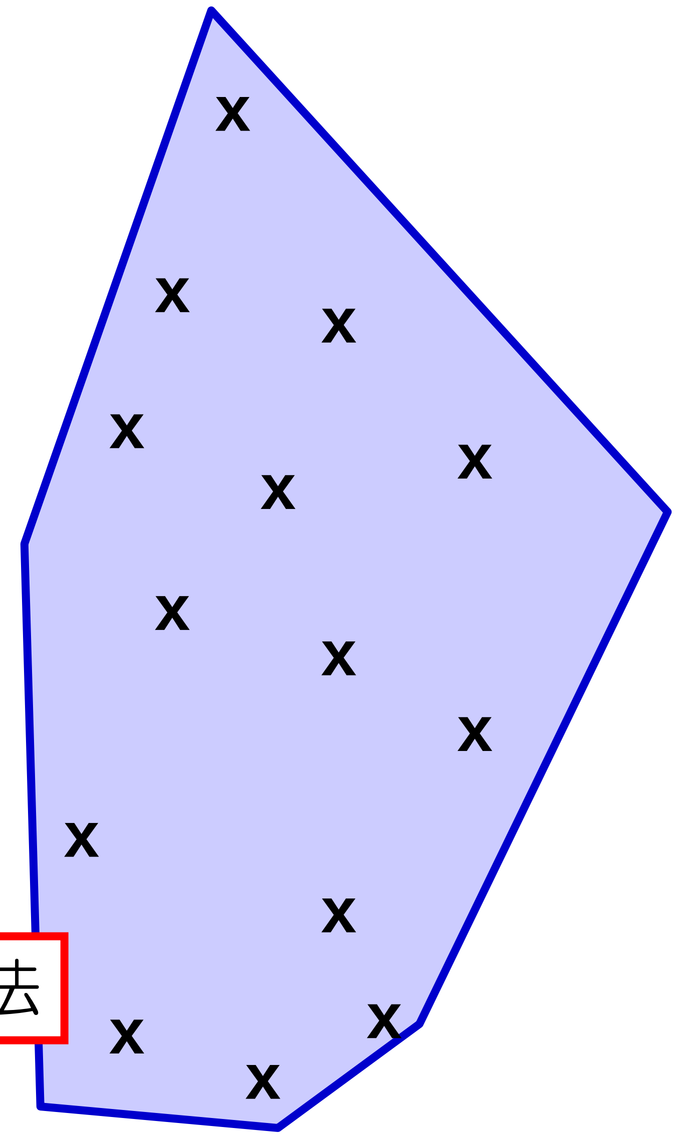
Ball-walkを用いて,

一様ランダムサンプリング可能

どうやって,

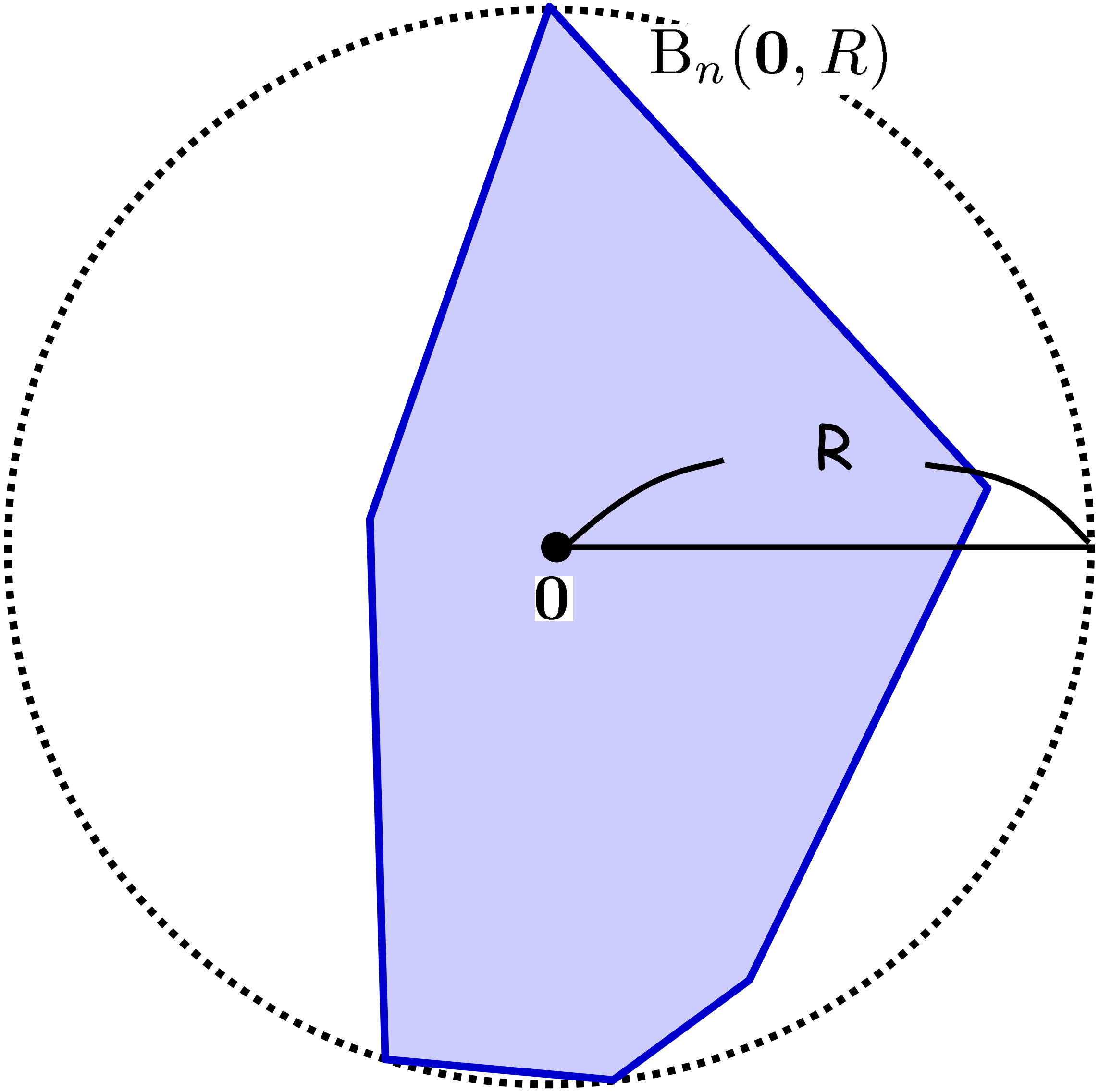
体積を計算するのか？

再帰的アルゴリズム + モンテカルロ法



再帰のアイデア

K: n次元凸体
体積V(K)を計算

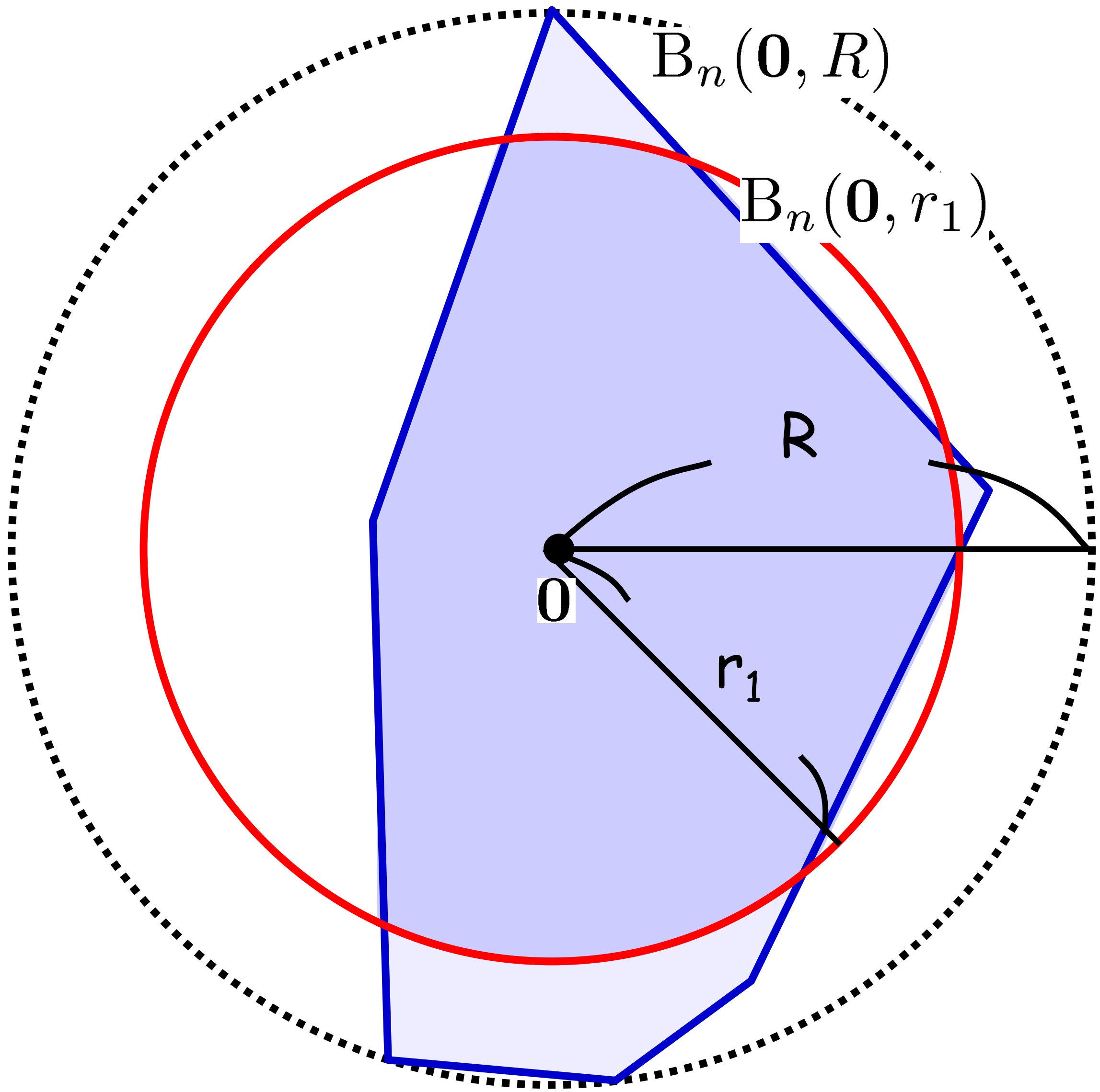


再帰のアイデア

K: n次元凸体
体積V(K)を計算

$$K_1 \stackrel{\text{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_1)$$

$$\mathrm{V}(K) = \frac{\mathrm{V}(K)}{\mathrm{V}(K_1)} \cdot \mathrm{V}(K_1)$$



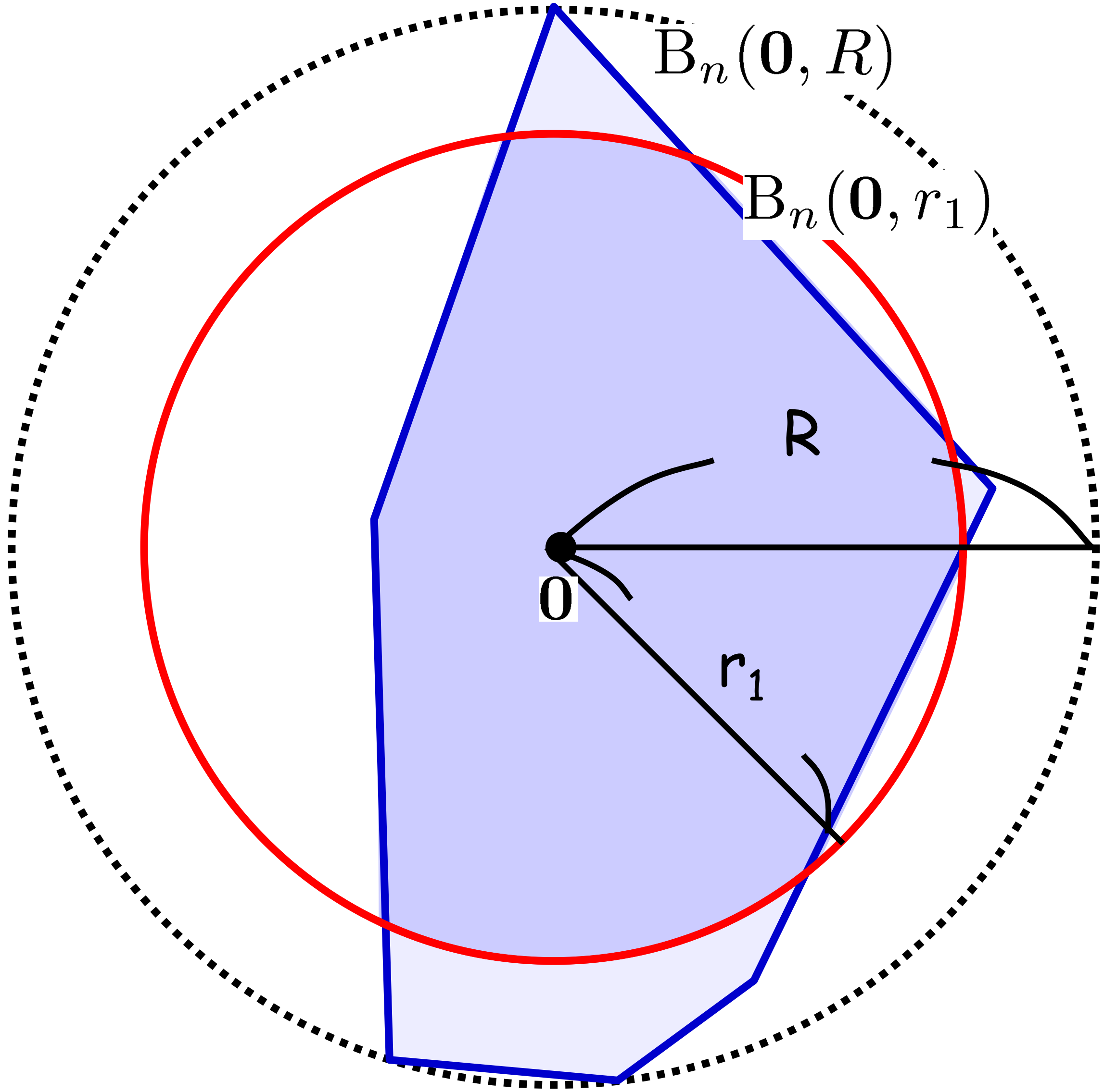
再帰のアイデア

K: n次元凸体
体積V(K)を計算

$$K_1 \stackrel{\text{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_1)$$

$$\mathrm{V}(K) = \frac{\mathrm{V}(K)}{\mathrm{V}(K_1)} \cdot \mathrm{V}(K_1)$$

モンテカルロ法で計算



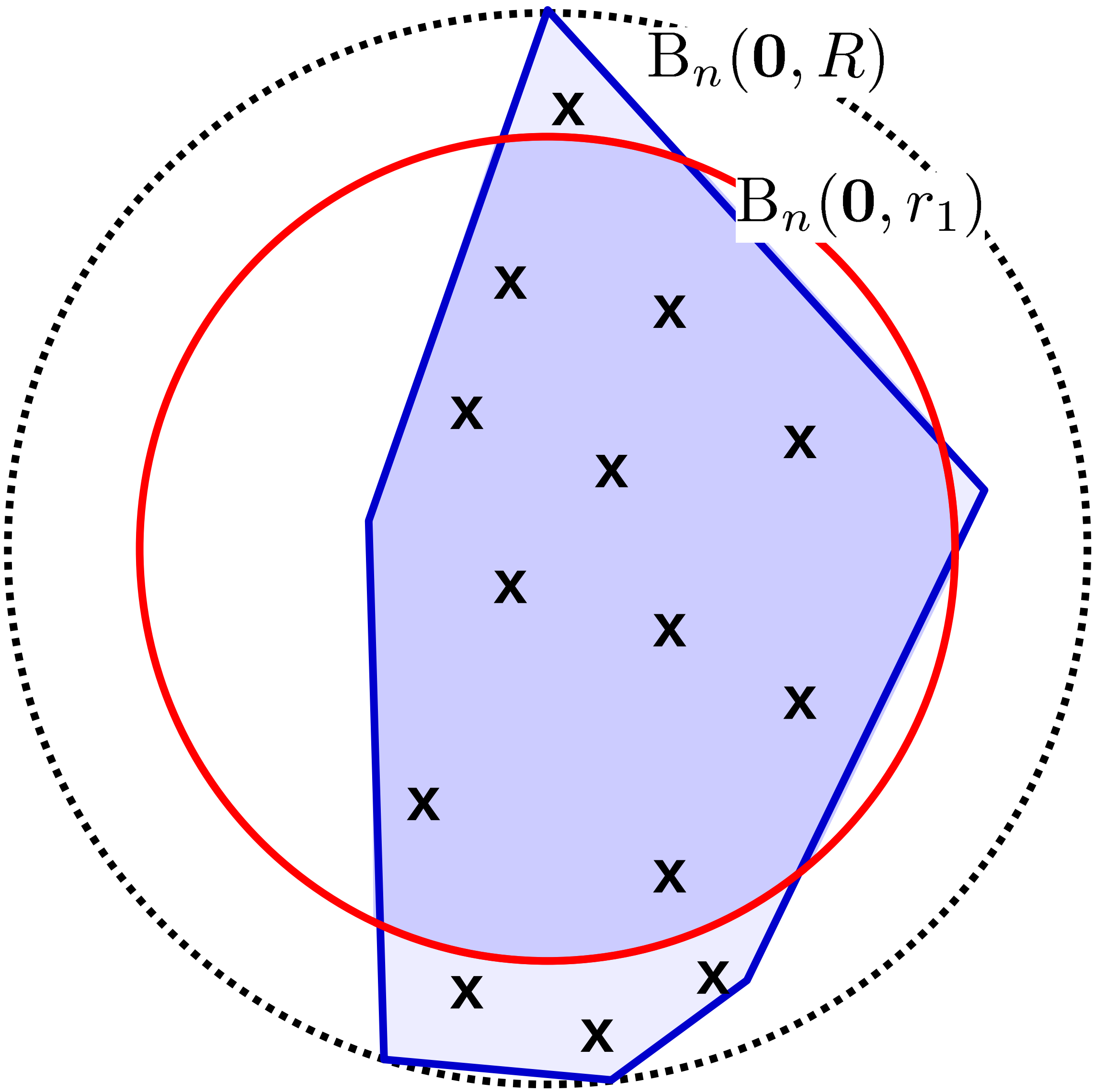
再帰のアイデア

K: n次元凸体
体積V(K)を計算

$$K_1 \stackrel{\text{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_1)$$

$$\mathrm{V}(K) = \frac{\mathrm{V}(K)}{\mathrm{V}(K_1)} \cdot \mathrm{V}(K_1)$$

モンテカルロ法で計算



再帰のアイデア

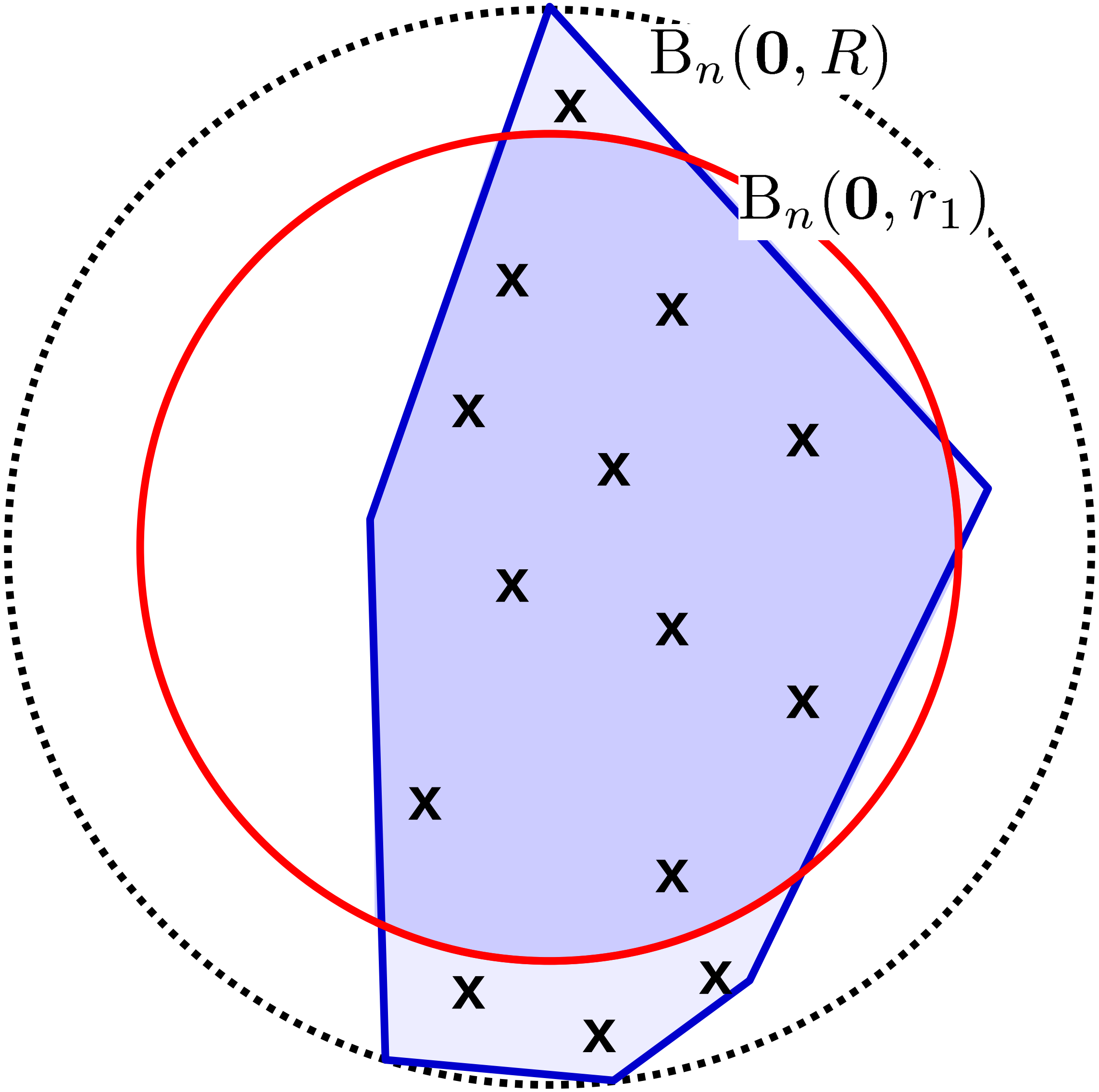
K: n次元凸体
体積V(K)を計算

$$K_1 \stackrel{\mathrm{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_1)$$

$$\mathrm{V}(K) = \frac{\mathrm{V}(K)}{\mathrm{V}(K_1)} \cdot \mathrm{V}(K_1)$$

モンテカルロ法で計算

K_1: n次元凸体
体積$V(K_1)$を計算



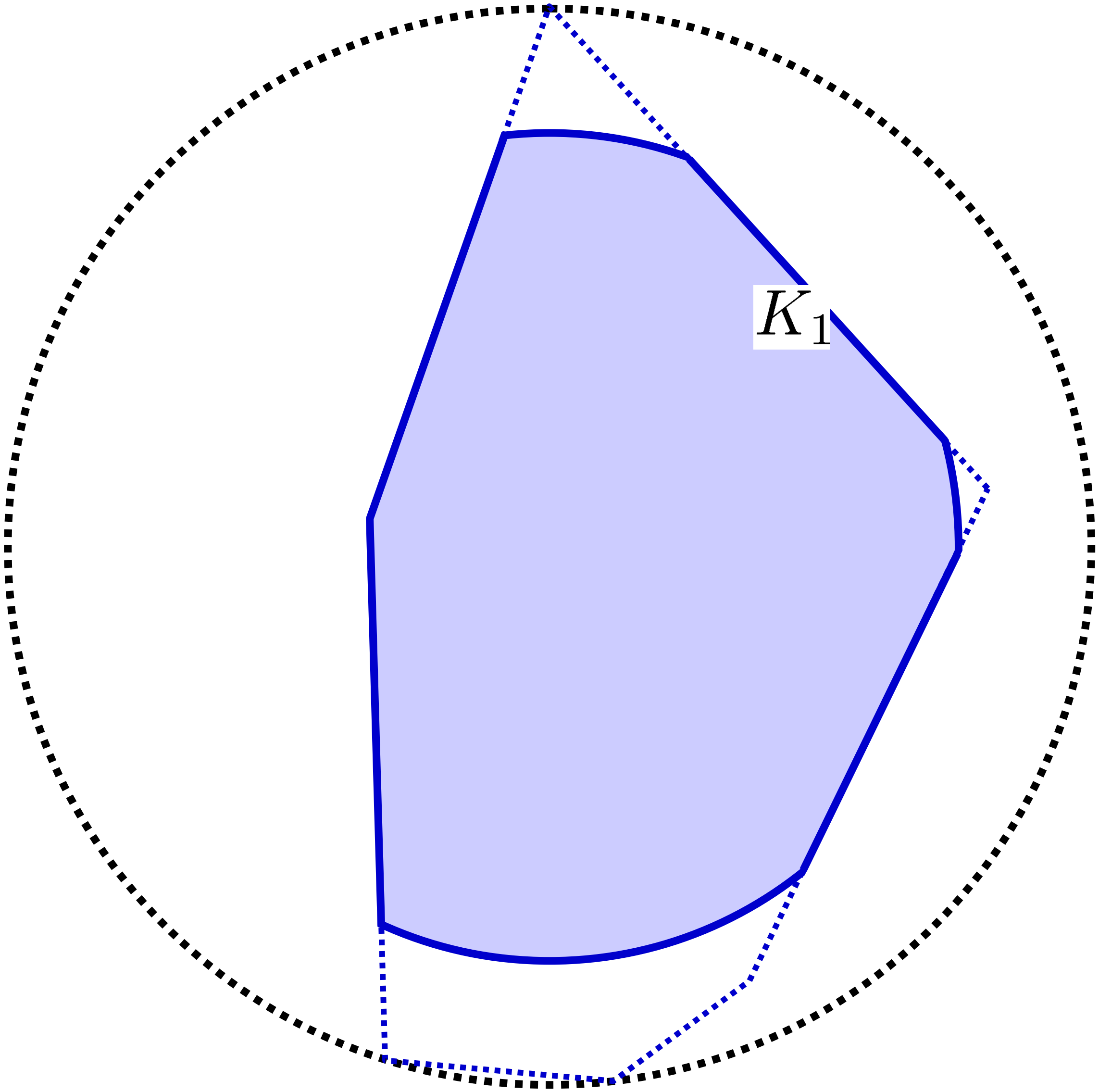
再帰のアイデア

K: n次元凸体
体積V(K)を計算

$$K_1 \overset{\text{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_1)$$

$$\mathrm{V}(K) = \frac{\mathrm{V}(K)}{\mathrm{V}(K_1)} \cdot \mathrm{V}(K_1)$$

モンテカルロ法で計算



K_1: n次元凸体
体積V(K_1)を計算

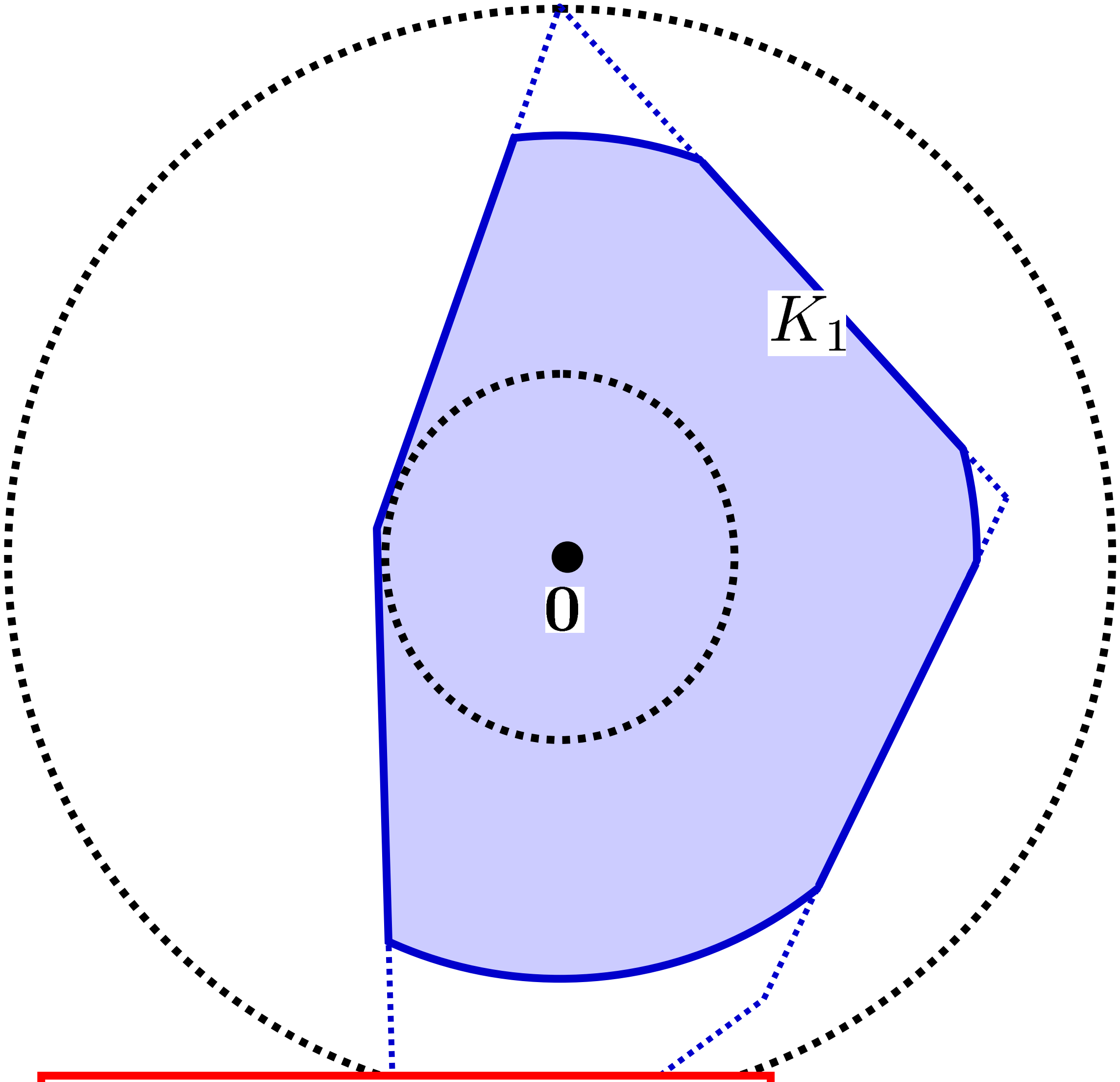
再帰のアイデア

K: n次元凸体
体積V(K)を計算

$$K_1 \stackrel{\text{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_1)$$

$$\mathrm{V}(K) = \frac{\mathrm{V}(K)}{\mathrm{V}(K_1)} \cdot \mathrm{V}(K_1)$$

モンテカルロ法で計算

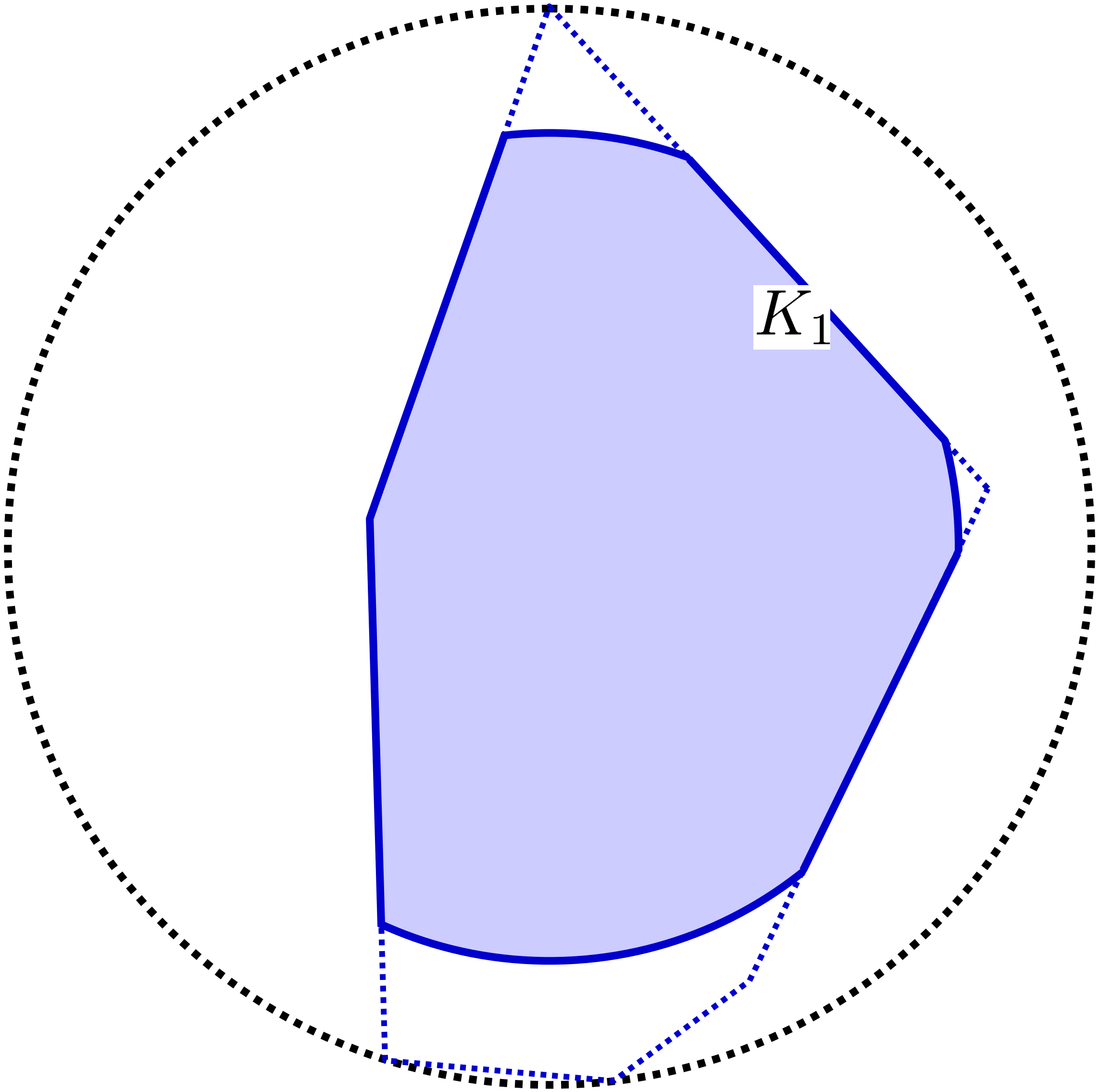


K_1: n次元凸体
体積V(K_1)を計算

K_1はふっくらとしている

再帰のアイデア

K_1: n次元凸体
体積$V(K_1)$を計算

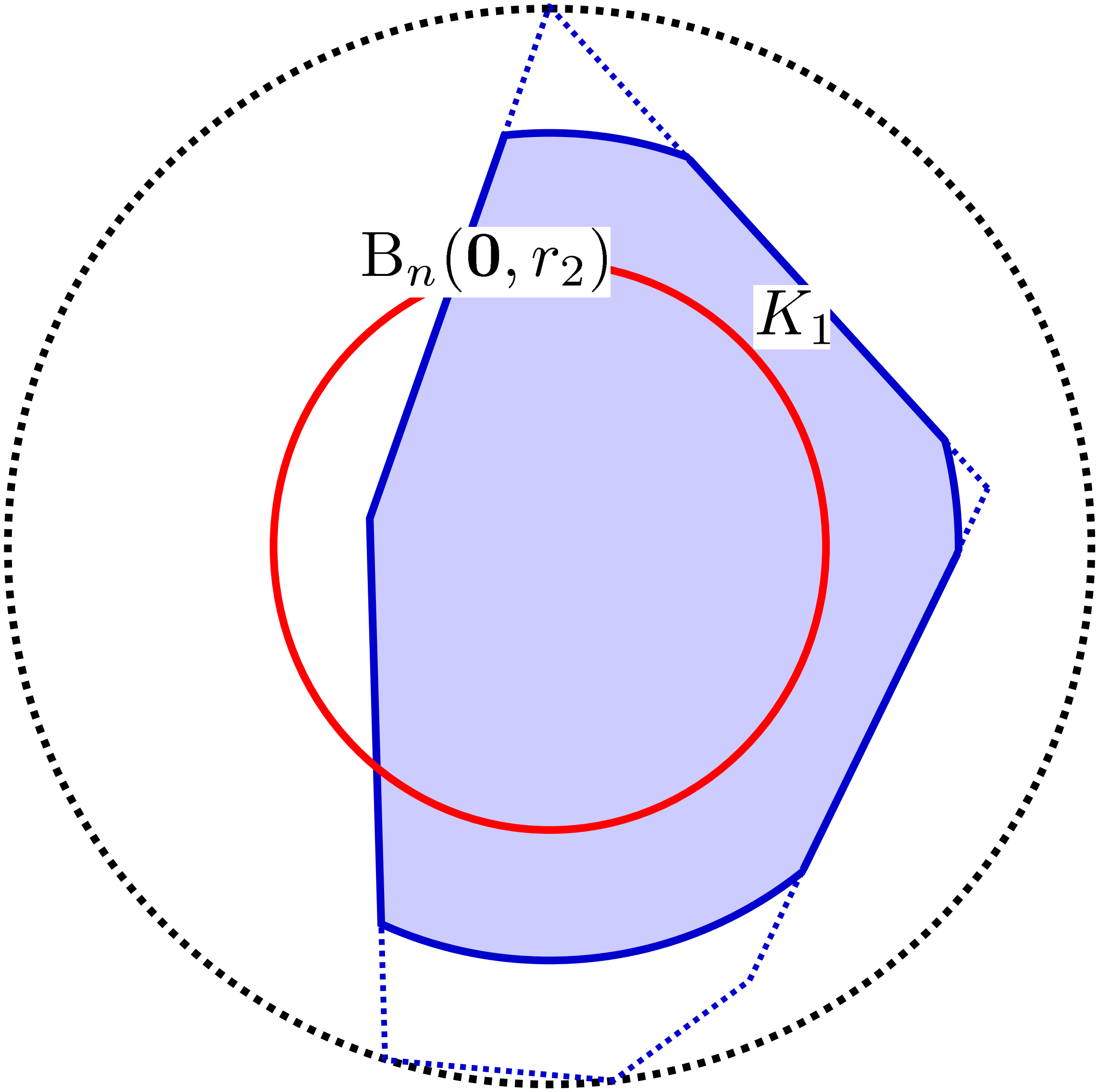


再帰のアイデア

K₁: n次元凸体
体積V(K₁)を計算

$$K_2 \stackrel{\text{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_2)$$

$$\mathrm{V}(K_1) = \frac{\mathrm{V}(K_1)}{\mathrm{V}(K_2)} \cdot \mathrm{V}(K_2)$$



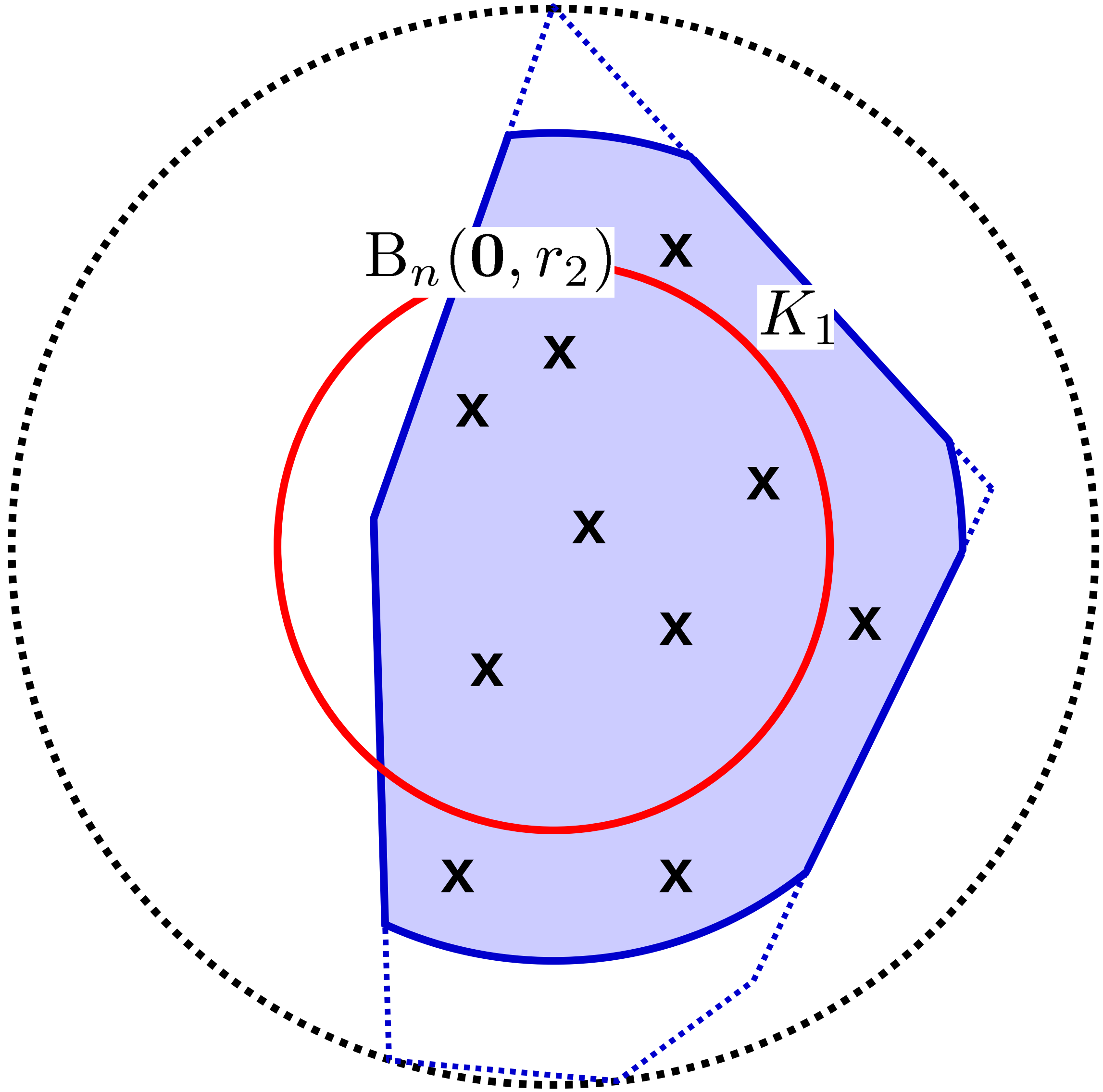
再帰のアイデア

K1: n次元凸体
体積V(K1)を計算

$$K_2 \stackrel{\text{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_2)$$

$$\mathrm{V}(K_1) = \frac{\mathrm{V}(K_1)}{\mathrm{V}(K_2)} \cdot \mathrm{V}(K_2)$$

モンテカルロ法で計算



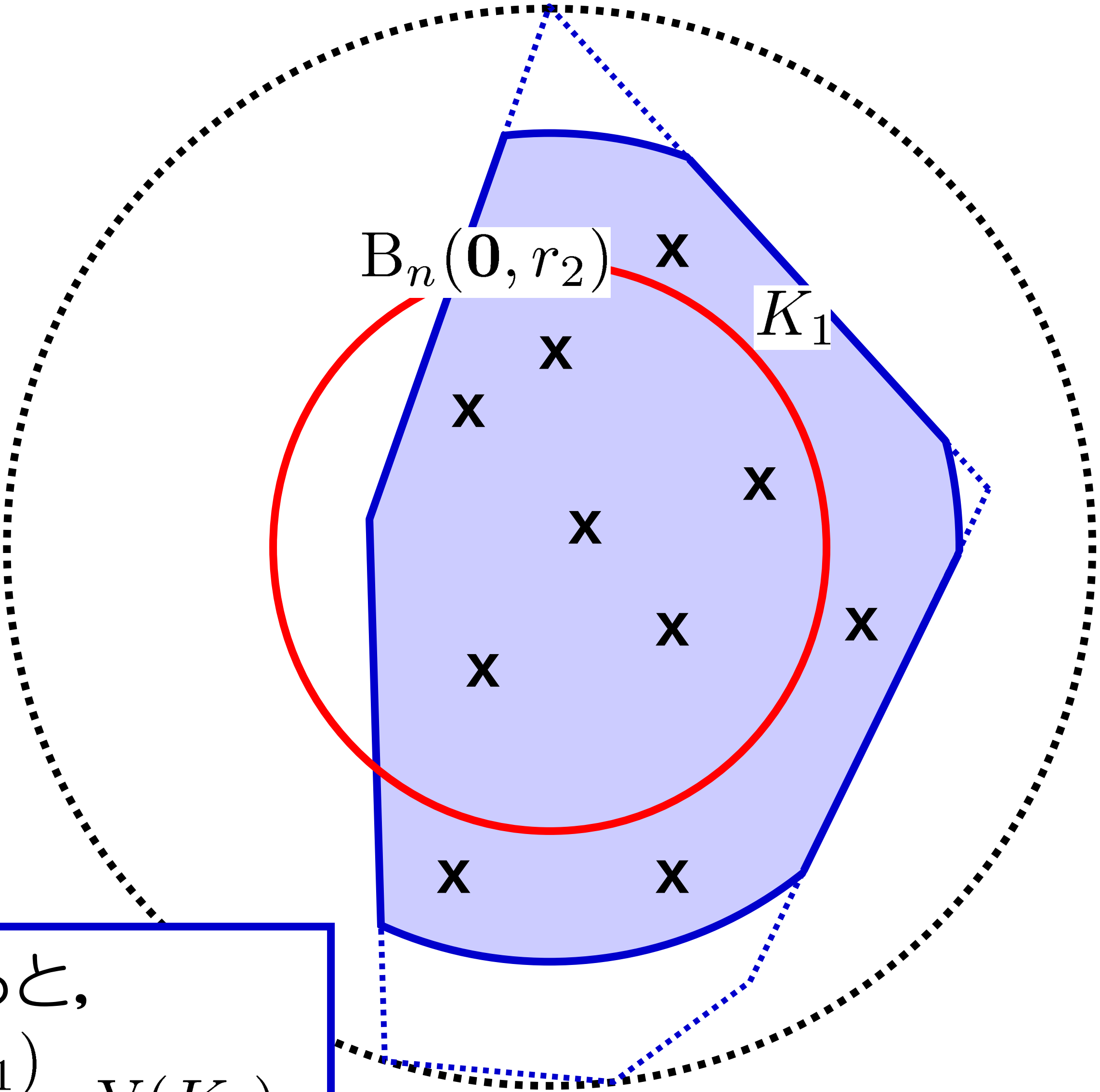
再帰のアイデア

K_1: n次元凸体
体積V(K_1)を計算

$$K_2 \stackrel{\text{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_2)$$

$$\mathrm{V}(K_1) = \frac{\mathrm{V}(K_1)}{\mathrm{V}(K_2)} \cdot \mathrm{V}(K_2)$$

モンテカルロ法で計算



先程の再帰式に代入すると,

$$\mathrm{V}(K) = \frac{\mathrm{V}(K)}{\mathrm{V}(K_1)} \cdot \frac{\mathrm{V}(K_1)}{\mathrm{V}(K_2)} \cdot \mathrm{V}(K_2)$$

再帰式

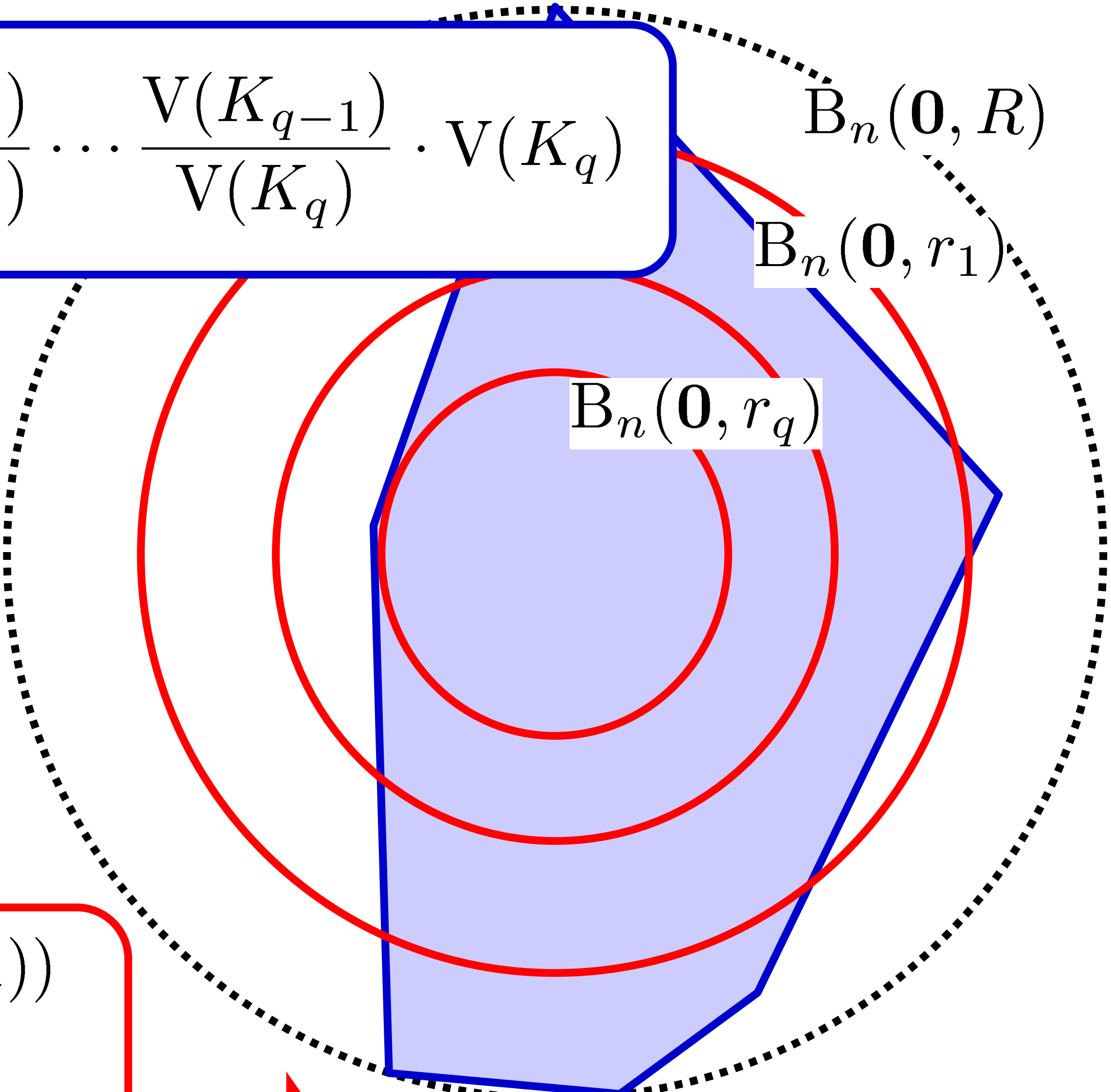
$$K_i \overset{\text{def.}}{=} K \cap \mathrm{B}_n(\mathbf{0}, r_i)$$

$$\mathrm{V}(K) = \frac{\mathrm{V}(K)}{\mathrm{V}(K_1)} \cdot \frac{\mathrm{V}(K_1)}{\mathrm{V}(K_2)} \cdots \frac{\mathrm{V}(K_{q-1})}{\mathrm{V}(K_q)} \cdot \mathrm{V}(K_q)$$

(ただし，$r_q = 1$)

$$\begin{aligned} K_q &= \mathrm{B}_n(\mathbf{0}, r_q) \\ &= \mathrm{B}_n(\mathbf{0}, 1) \end{aligned}$$

$$\begin{aligned} \mathrm{V}(K_q) &= \mathrm{V}(\mathrm{B}_n(\mathbf{0}, 1)) \\ &= \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2} + 1)} \end{aligned}$$



$\mathrm{V}(K)$ が求まる！

再帰の回数

再帰の回数 => 少なくしたい

比 => 大きくしたい

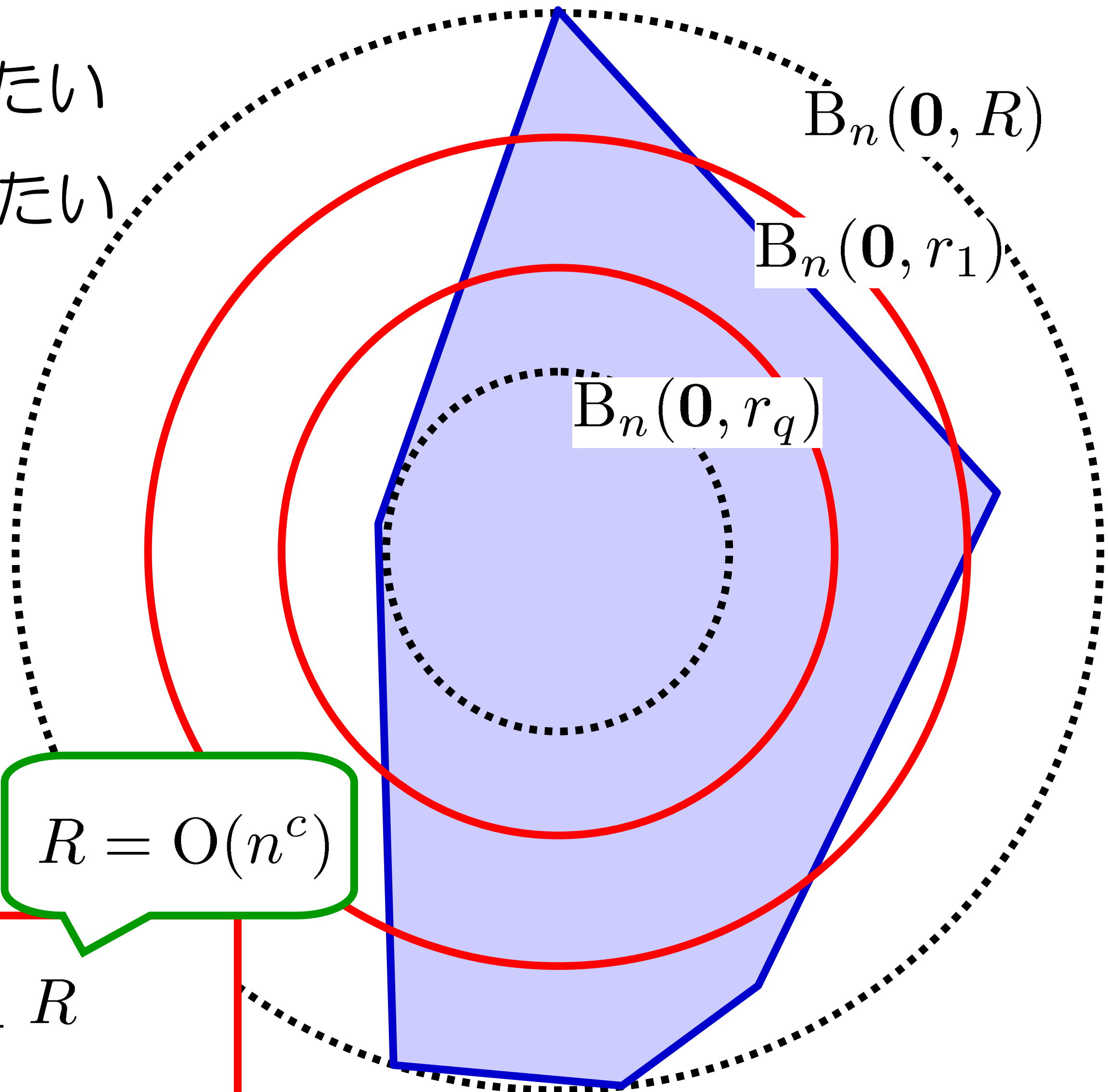
$$\frac{r_i}{r_{i-1}} = \frac{1}{\sqrt[n]{2}}$$

比

$$\frac{V(K_i)}{V(K_{i-1})} \geq \frac{1}{2}$$

$$R = \mathrm{O}(n^c)$$

$$\begin{aligned}(\text{再帰の回数}) &= \log_{2^{\frac{1}{n}}} R \\ &= c \cdot n \log_2 n\end{aligned}$$

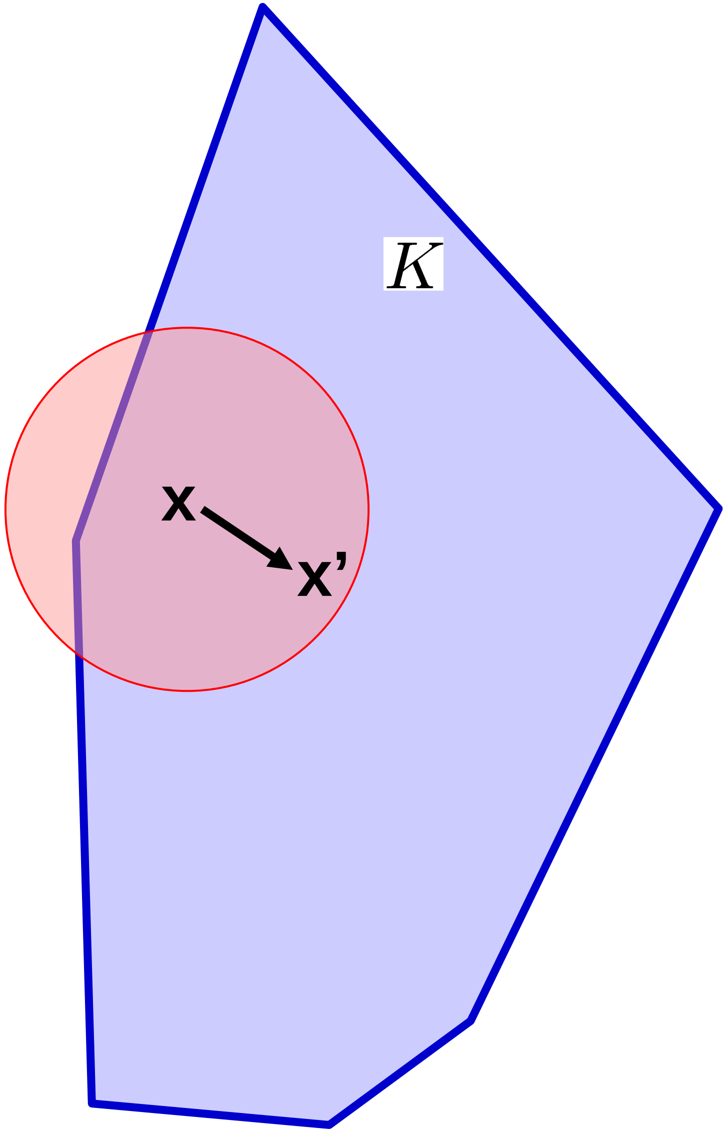


理論的な近似精度保証に向けて

Ball-walk

=> 定常分布は一様.

十分な回数の推移は何回？



[Kannan, Lovasz, Simonovitz 1993]

理論的な近似精度保証に向けて

Ball-walk

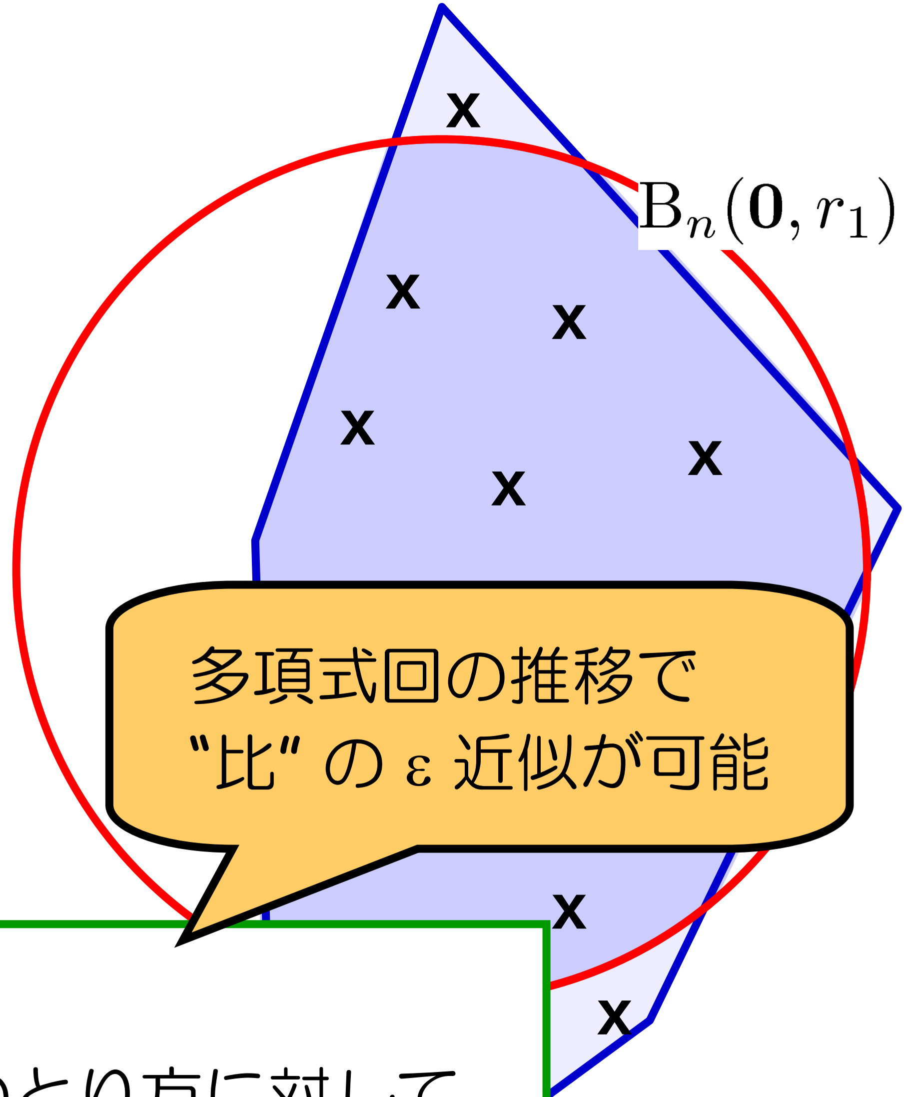
=> 定常分布は一様

十分な回数の推移は何回？

モンテカルロ法

- 真の比 ρ
- 近似比 ρ' (分布の誤差に起因)

|ρ' - ρ| < ε



mixing time:

誤差が最大となる部分集合のとり方に対して,

誤差 ε が保証される推移の回数.

3章のまとめ

効率的な再帰的モンテカルロ法を設計するには？

1. 帰着先の割合が大きい
2. 再帰の回数が小さい
3. 帰着先のランダムサンプリングが容易

⇓

離散構造の場合は？

MCMC法における近似精度保証 - Table of Contents

4. 自己帰着可能性と離散構造

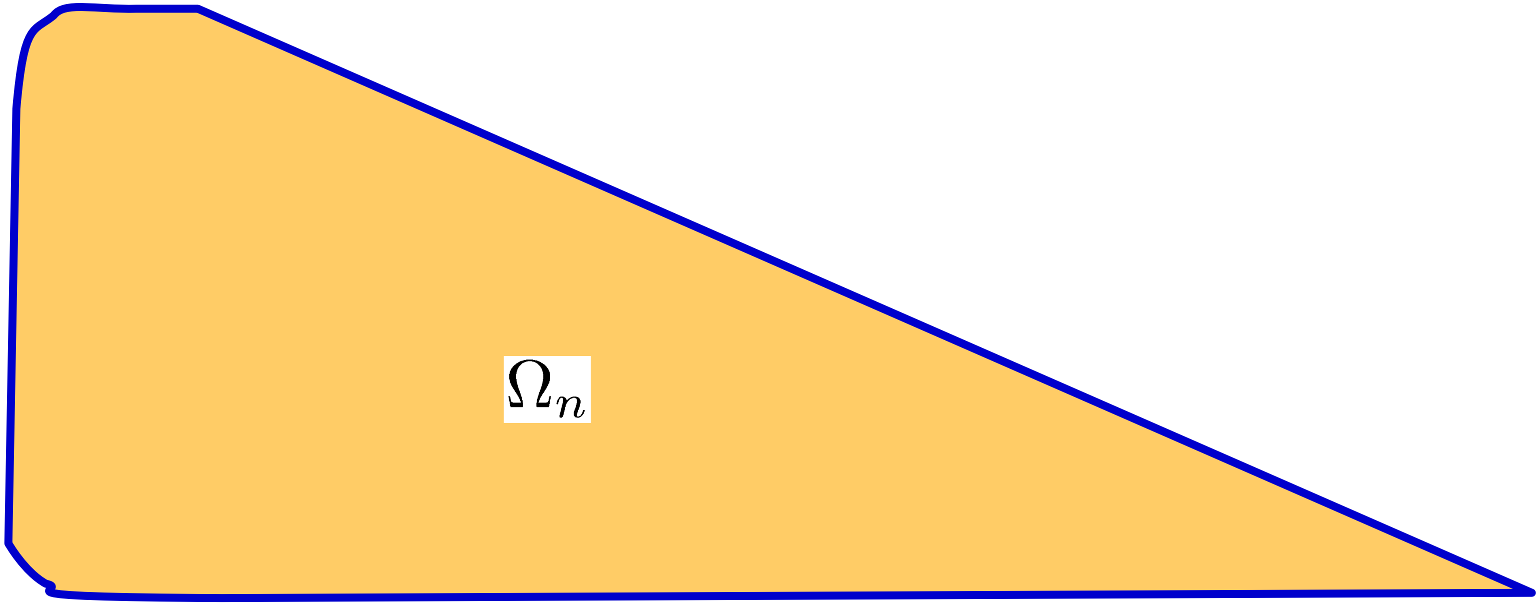
4.1: 0-1ナップサック解の数え上げ

0-1ナップサックの許容解集合

Given

正数 $a_1, a_2, \ldots, a_n$, b.

$$\Omega_n \stackrel{\text{def.}}{=} \left\{ \boldsymbol{x} \in \{0,1\}^n \;\middle|\; \sum_{i=1}^{n} a_i x_i \leq b \right\}$$



0-1ナップサックの許容解集合

Given

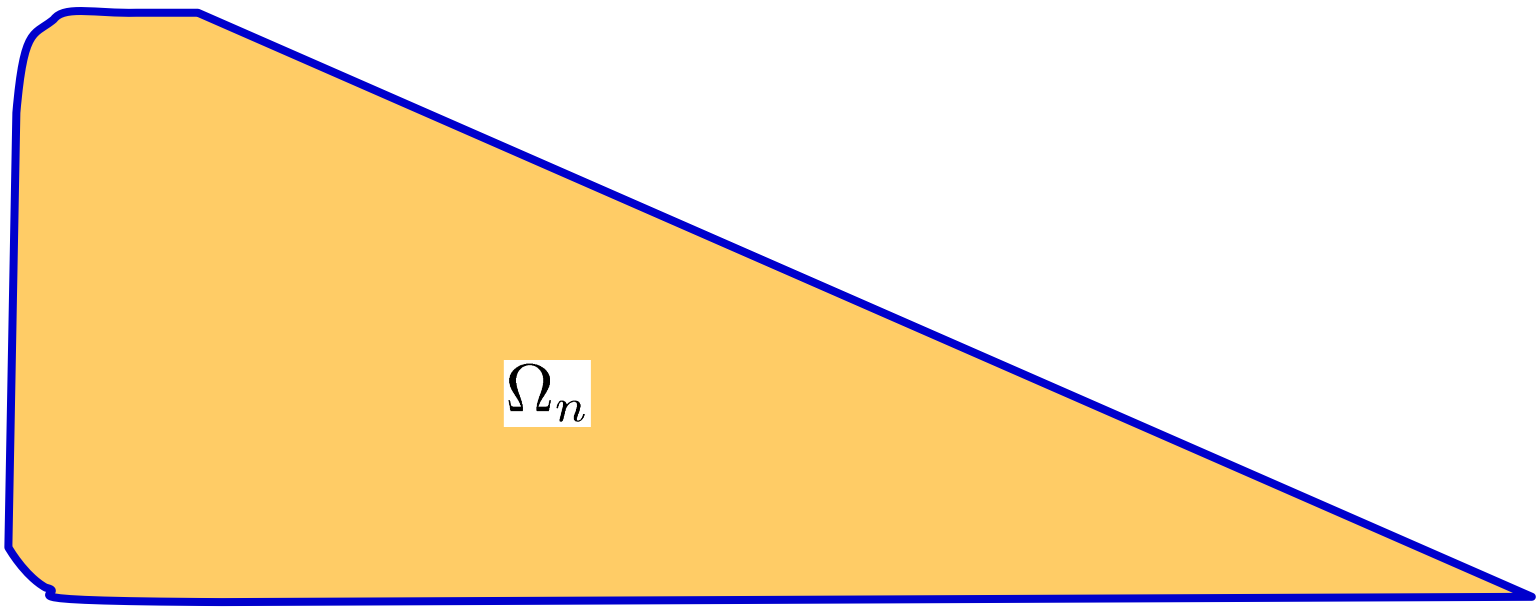
正数 $a_1, a_2, \dots, a_n$, b.

$$\Omega_n \stackrel{\text{def.}}{=} \left\{ \boldsymbol{x} \in \{0,1\}^n \;\middle|\; \sum_{i=1}^{n} a_i x_i \leq b \right\}$$

目的

$|\Omega_n|$の計算

#P完全



[Morris Sinclair 2004]

0-1ナップサックの許容解集合上のマルコフ連鎖

Given

正数 $a_1, a_2, \ldots, a_n$, b.

$$\Omega_n \stackrel{\text{def.}}{=} \left\{ \boldsymbol{x} \in \{0,1\}^n \;\middle|\; \sum_{i=1}^{n} a_i x_i \leq b \right\}$$

マルコフ連鎖

X ∈ Ω_n : 現在の状態

X' : 次の時刻の状態

Step 1.

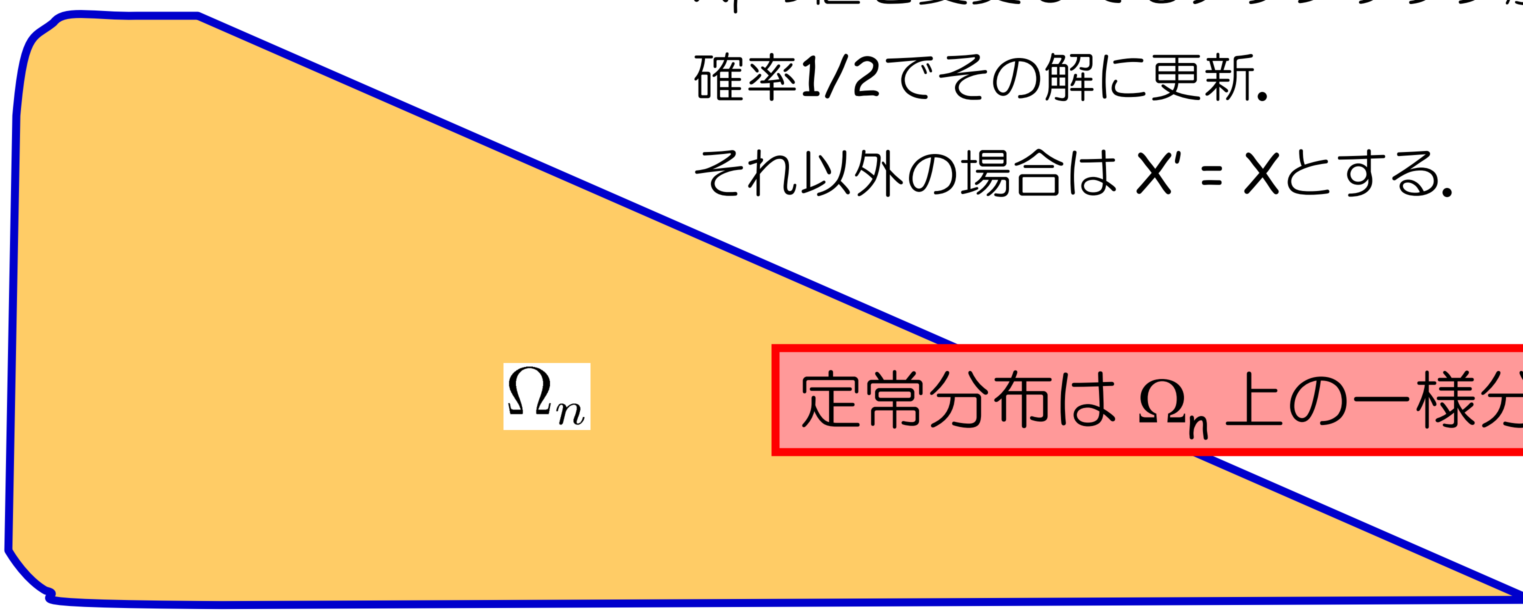
添え字 i ∈{1,...,n}を一様ランダムに選ぶ.

Step 2.

X_i の値を変更してもナップサック解なら,

確率1/2でその解に更新.

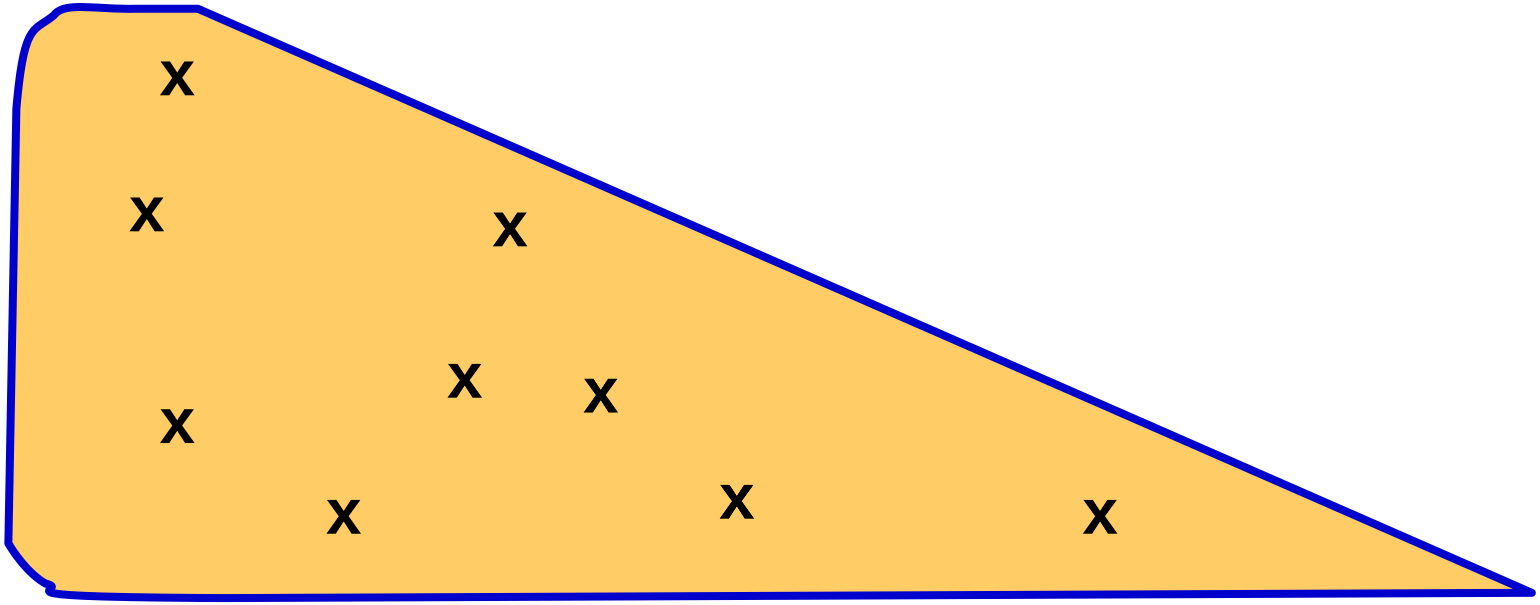
それ以外の場合は X' = Xとする.



定常分布は Ω_n 上の一様分布

一様ランダムサンプリング

$$\Omega_n \stackrel{\text{def.}}{=} \left\{ \boldsymbol{x} \in \{0,1\}^n \;\middle|\; \sum_{i=1}^{n} a_i x_i \le b \right\}$$



再帰のアイデア

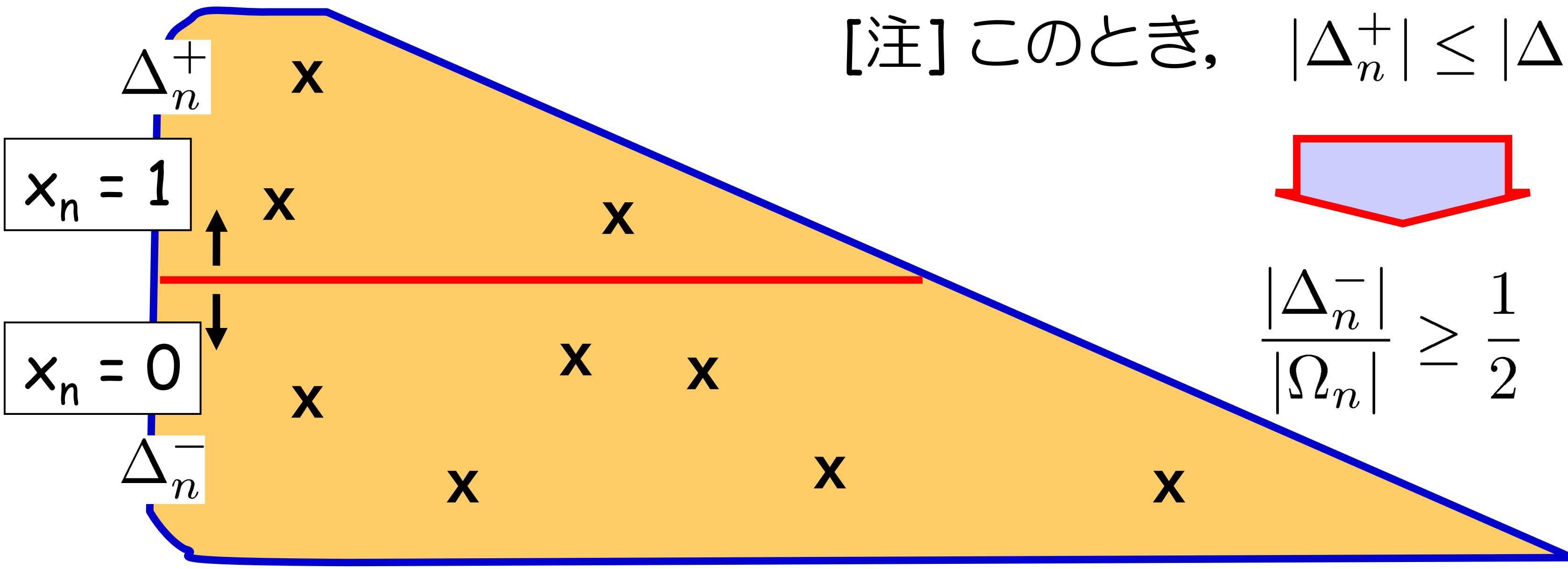
$$\Omega_n \stackrel{\text{def.}}{=} \left\{ \boldsymbol{x} \in \{0,1\}^n \;\middle|\; \sum_{i=1}^{n} a_i x_i \leq b \right\}$$

$x_n = 1$

$x_n = 0$

$$\Delta_n^+ \stackrel{\text{def.}}{=} \{\boldsymbol{x} \in \Omega_n \mid x_n = 1\}$$

$$\Delta_n^- \stackrel{\text{def.}}{=} \{\boldsymbol{x} \in \Omega_n \mid x_n = 0\}$$



[注] このとき, $|\Delta_n^+| \leq |\Delta_n^-|$

$$\frac{|\Delta_n^-|}{|\Omega_n|} \geq \frac{1}{2}$$

再帰のアイデア

$$\Omega_n \stackrel{\text{def.}}{=} \left\{ \boldsymbol{x} \in \{0,1\}^n \;\middle|\; \sum_{i=1}^{n} a_i x_i \leq b \right\}$$

$x_n = 0$

$$\Delta_n^- \stackrel{\text{def.}}{=} \{ \boldsymbol{x} \in \Omega_n \mid x_n = 0 \}$$

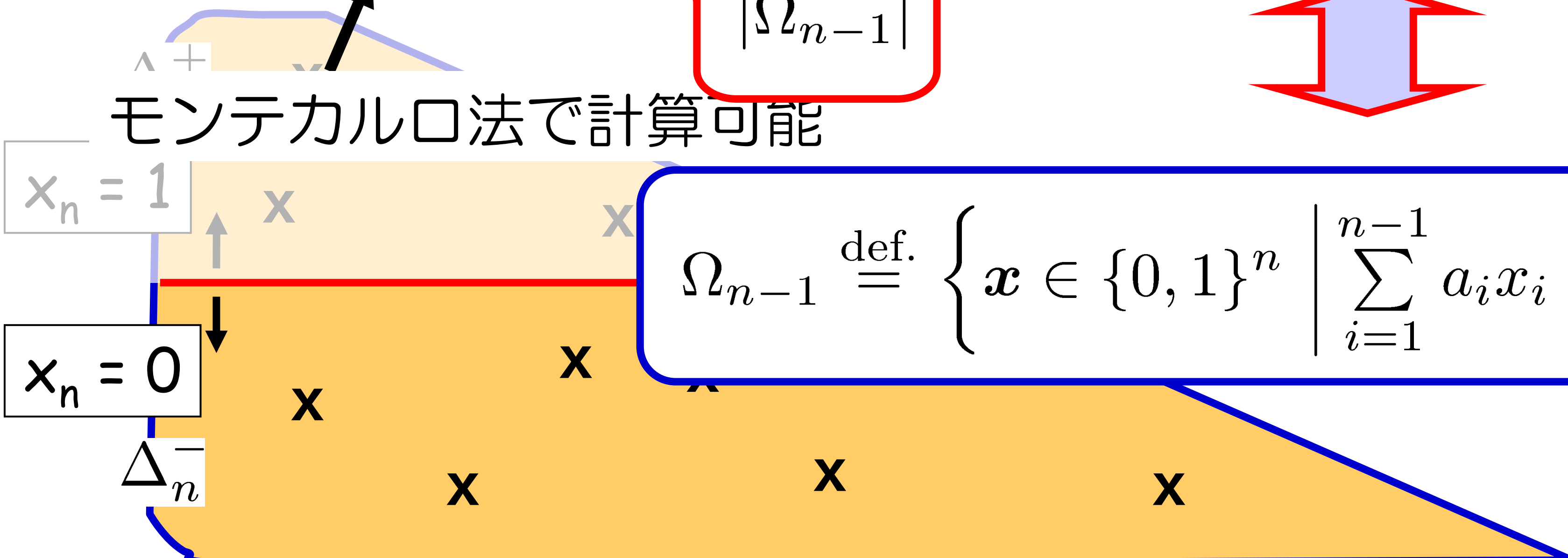
関係式

$$|\Omega_n| = \frac{|\Omega_n|}{|\Delta_n^-|} \cdot |\Delta_n^-|$$

$|\Omega_{n-1}|$

モンテカルロ法で計算可能

$$\Omega_{n-1} \stackrel{\text{def.}}{=} \left\{ \boldsymbol{x} \in \{0,1\}^n \;\middle|\; \sum_{i=1}^{n-1} a_i x_i \leq b \right\}$$



再帰式

$$\boxed{\Omega_n \stackrel{\text{def.}}{=} \left\{ \boldsymbol{x} \in \{0,1\}^n \;\middle|\; \sum_{i=1}^{n} a_i x_i \le b \right\}}$$

$$\begin{aligned}
|\Omega_n| &= \frac{|\Omega_n|}{|\Delta_n^-|} \cdot |\Omega_{n-1}| \\
&= \frac{|\Omega_n|}{|\Delta_n^-|} \cdot \frac{|\Omega_{n-1}|}{|\Delta_{n-1}^-|} \cdot |\Omega_{n-2}| \\
&= \cdots \\
&= \frac{|\Omega_n|}{|\Delta_n^-|} \cdot \frac{|\Omega_{n-1}|}{|\Delta_{n-1}^-|} \cdots \frac{|\Omega_2|}{|\Delta_2^-|} \cdot |\Omega_1|
\end{aligned}$$

自己帰着可能性 (self-reducibility)

1. $\Delta_i \subseteq \Omega_i$
2. $|\Delta_i| = |\Omega_{i-1}|$
3. $\dfrac{|\Delta_i|}{|\Omega_i|}$ が，ある程度大きい.

再帰式

$$|\Omega_n| = \frac{|\Omega_n|}{|\Delta_n|} \cdot \frac{|\Omega_{n-1}|}{|\Delta_{n-1}|} \cdots \frac{|\Omega_2|}{|\Delta_2|} \cdot |\Omega_1|$$

に基づく再帰的モンテカルロ法

Ω_i 上の効率的なサンプリング法が鍵

4. 自己帰着可能性と離散構造

4.2: 2元分割表の数え上げ (問題紹介)

- 輸送多面体中の整数格子点の数え上げ.

2元分割表

2元分割表

- ✓ m 行 n 列の表 (mxn行列)
- ✓ 各セルには非負整数が入る
- ✓ （与えられた）周辺和を満たす

						12
						18
5	4	3	7	5	6	30

cf. Hitchcock 輸送問題の整数解
・供給地 2 / 需要地 6

2元分割表

2元分割表

- ✓ m 行 n 列の表 (mxn行列)
- ✓ 各セルには非負整数が入る
- ✓ （与えられた）周辺和を満たす

						12
						18
5	4	3	7	5	6	30

5	4	3	0	0	0	12
0	0	0	7	5	6	18
5	4	3	7	5	6	30

状態A

4	3	1	3	1	0	12
1	1	2	4	4	6	18
5	4	3	7	5	6	30

状態B

0	0	0	1	5	6	12
5	4	3	6	0	0	18
5	4	3	7	5	6	30

状態C

与えられた周辺和を満たす2元分割表の個数を求める問題

⇒ #P完全 (行数を2に固定しても)

2元分割表

2元分割表

- ✓ m 行 n 列の表 (mxn行列)
- ✓ 各セルには非負整数が入る
- ✓ （与えられた）周辺和を満たす

						12
						18
5	4	3	7	5	6	30

- 2xn分割表のFPRAS [Dyer, Greenhill 2000]
- m x n分割表の再帰構造 [K, Matsui 2002]
- m*x n分割表の多項式時間サンプリング法 [Cryan et al 2002]

末解決

m x n分割表の多項式時間サンプリング法

MCMC法における近似精度保証 - Table of Contents

5: パーマネントの計算

- 2部グラフの完全マッチングの数え上げ

パーマネント

A: 正方行列 (nxn)

$$\mathrm{per}(A) \stackrel{\mathrm{def.}}{=} \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

(S_n: n次の対称群)

cf.

$$\det(A) \stackrel{\mathrm{def.}}{=} \sum_{\sigma \in S_n} \mathrm{sign}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

各項は2部グラフの

完全マッチングに対応

2部グラフの完全マッチング

#P完全 [Valiant 1979]

A: 0-1正方行列 (nxn)

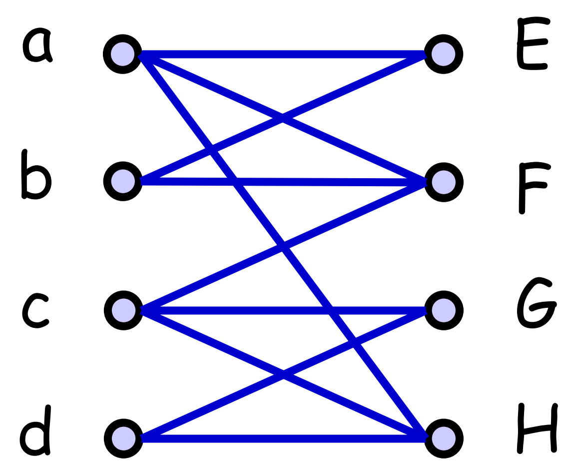
$$\mathrm{per}(A) \stackrel{\mathrm{def.}}{=} \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

(S_n: n次の対称群)

$$\begin{array}{c|c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} E \\ F \\ G \\ H \end{array} & \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

パーマネント

$\Longleftrightarrow$



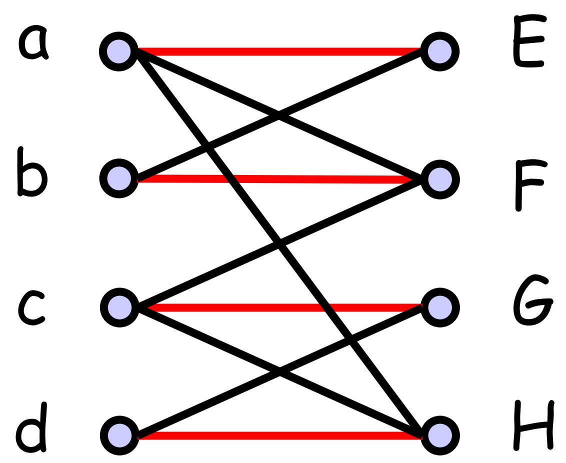
完全マッチングの個数

完全マッチング上のマルコフ連鎖の設計は難しい

Ω := 完全マッチング全体の集合

X $\in$ Ω: 現在の状態

次の時刻の状態？？？



状態空間: n マッチング+ n-1マッチング [Broder 1986]

Ξ := n マッチング + n-1マッチング

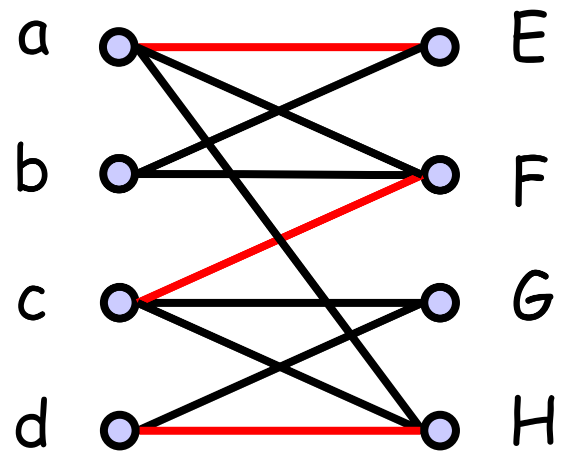
状態空間: Ξ

推移規則:

枝 e を一様ランダムに選ぶ.

- X が n マッチングの場合
- X が n-1 マッチングの場合

定常分布: Ξ 上の 一様分布



状態空間: n マッチング+ n-1マッチング [Broder 1986]

Ξ := n マッチング + n-1マッチング

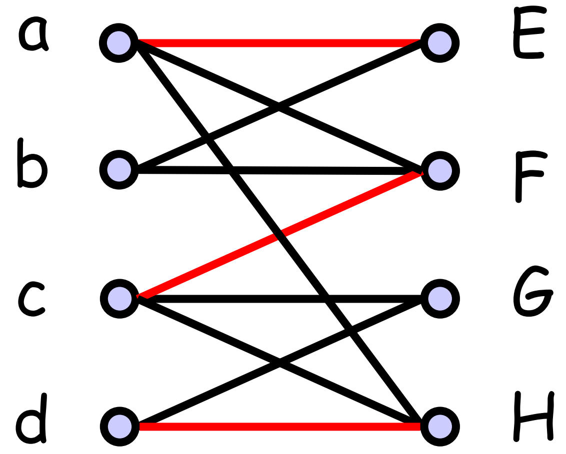
状態空間: Ξ

推移規則:

枝 e を一様ランダムに選ぶ.

- X が n マッチングの場合
- X が n-1 マッチングの場合

定常分布: Ξ 上の 一様分布

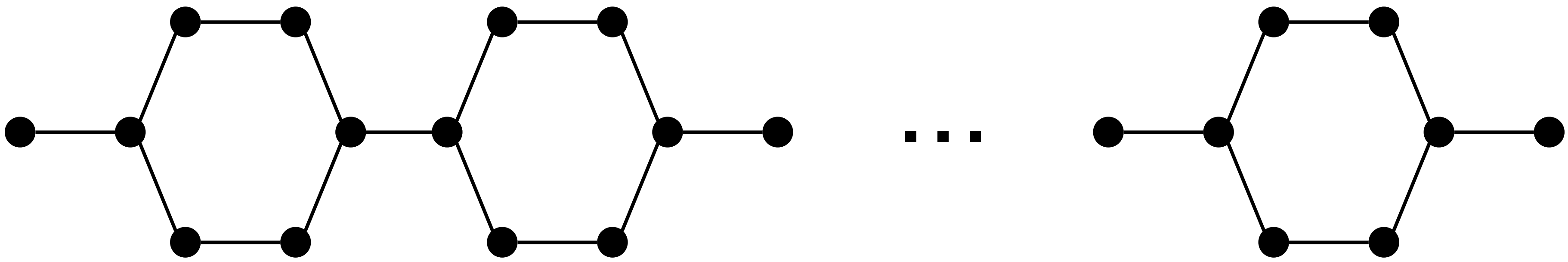


rapidly mixing [Jerrum, Sinclair 1993]

棄却サンプリング？

棄却サンプリングは非効率的

k-hexagon



n-1マッチングの個数 $\cong 2^k \cong 2^{n/6}$

完全マッチングの個数 = 1

棄却サンプリングでは,

完全マッチングを得るのに指数時間かかる

FPRAS [Jerrum, Sinclair, Vigoda 2001]

“アニーリング”を利用したアルゴリズム

FPRAS [Jerrum, Sinclair, Vigoda 2001]

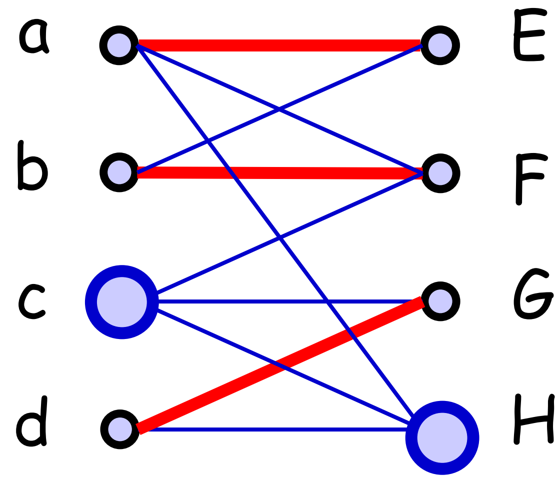
問題点: (一般には) n-1マッチングの比率が大きい.

アイデア 1.

マッチングに "重み" をつけて,

重みに比例する確率分布を考える.

マッチング重み関数 w : "hole" -> R_+

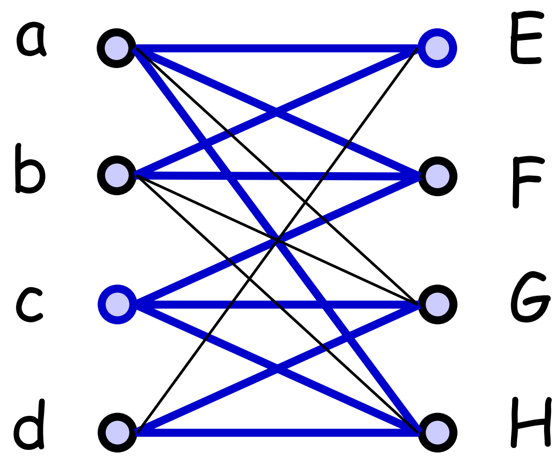


hole: n-1マッチングで,
マッチングされていない頂点対

$$\sum_{X\text{ が完全マッチング}} w(X) = \sum_{X\text{がn-1マッチング, [u,v]がhole}} w(X)$$

重み関数 w の計算が困難. => "アニーリング" を利用

FPRAS [Jerrum, Sinclair, Vigoda 2001]



アイデア 2.

完全2部グラフ + 枝活性度

=> 活性度を "アニーリング" する.

- 状態空間: nマッチング+n-1マッチング
- マッチング重み関数: 現在の枝活性度に関して

枝活性度の積

$$\sum_{X\text{ が完全マッチング}} w(X)\cdot\lambda(X) \cong \sum_{\substack{X\text{がn-1マッチング},\\ [u,v]\text{がhole}}} w(X)\cdot\lambda(X)$$

- 定常分布: (枝活性度の積) x (マッチング重み) に比例

枝活性度を "アニーリング" しながら, 重み関数 w を算定.

<= "平均" の近似計算 by モンテカルロ法

6: Concluding Remarks

まとめ

MCMC法と近似精度保証

key word: 自己帰着可能性 (self-reducibility)

1. 帰着先の割合が大きい
2. 再帰の回数が小さい
3. 帰着先のランダムサンプリングが容易

(理論的な厳密さを追求しなければ)

アイデアはそれほど難しくない.

ぜひ，MCMC法を使ってみましょう！

今後の課題

未解決問題 [多項式時間収束のマルコフ連鎖]

- ✓ 2元分割表 (m x n)
- ✓ マトロイドの基
- ✓ 分配束構造
- ✓ グラフ中の森
- ✓ コーダルグラフサンドイッチ

離散の難しさ (連続との比較)

- 非等方性
- スケーリングの難しさ

関連する話題

- ◆ 統計学: bridge sampling, path sampling
- ◆ 統計物理: レプリカ交換法, 拡張アンサンブル法
- ◆ 最適化: annealing

The end

Thank you for the attention.