# Coarse 格子を正単体とする入れ子格子符号

## Nested Lattice Codes With Regular Simplex Coarse Lattices

九州大学大学院システム情報科学府情報学専攻
◎韓 楽歴

東京工業大学工学院情報通信系
實松 豊

2021年12月10日（金）
10時00分ー10時20分

九州大学

# 1. はじめに（研究背景）

- ガウス型通信路における誤り訂正符号
  - 2進符号
    - 低SNRでは，bipolar(+1,-1)で送信
    - 高SNRでは，多値変調の信号点にマッピング
  - 実数値に直接符号化
    - non-binary 符号
    - スパース重ね合わせ符号（←圧縮センシングに基づく復号法）
    - 格子符号（本研究）
- (重要な先行研究) ErezとZamir(2004)は，入れ子格子符号によりガウス型通信路の容量を達成できることを証明した.
  - ディザ信号（人工雑音）を送受信機で共有する，独特な符号化法 （素朴な疑問：ディザは必要か？ ）
- そこで，ディザ信号の効用を，具体的に実験して調査することにした.

# これまでの格子符号の研究(1/2)



AWGN通信路容量の達成可能性の証明

- de Buda (1975) 球型領域を持つ格子符号とラティス復号により
  レート $1/2 \log SNR$ を達成できることを証明.
- de Buda (1989) 球の表面領域を持つ格子符号と最近傍復号により
  通信路容量$1/2 \log(1 + SNR)$ を達成できると主張
  (誤りが発見されたが，後に Linderらにより修復された)
- Loeliger (1992) ラティス復号では，$1/2 \log SNR$ までしか達成できない
  と推測
- Erez & Zamir (2004) Dither信号を入れた入れ子格子符号とラティス復
  号により通信路容量$1/2 \log(1 + SNR)$を達成できることを証明

(注意) 最近傍復号：最も近い符号語に復号（NP困難）
ラティス復号：格子点(符号語でない格子点も含む)に復号

# これまでの格子符号の研究(2/2)



符号の具体的な構成法（入れ子格子符号)

- Conway & Sloane (1983) :
  整数倍の格子のボロノイ領域を持つ入れ子格子を提案
- Forney (1989) Conway&Sloaneの入れ子格子を拡張し，入れ子格子と二元の誤り訂正符号を拡張する Construction A, B, C, Dを提案.

- LDPC符号との組み合わせ
  - Sommer (2008) LDLC（Low Density Lattice Codes)符号
  - di Pietro (2012) 整数符号LDA (Low Density Construction-A)
  - B. da Silva & D. Silva (2019): LDPC符号をConstruction D’で拡張した符号を提案
  - Zhou & Kurkoski (2021) B. da Silva & D. Silvaの方法を Shaping Gain を得られるように改良

本論文では，Erez-Zamir(2004)の符号化・復号化法を研究する

# 研究背景



Erez-Zamirの論文では,

- 入れ子格子符号を使う.
  - fine格子とcoarse格子の2つを使った構成法
- 符号化の例としてランダムmod-p符号と Construction A を組み合わせた方法が示された
- 復号法は，ユークリッドラティス復号（詳細は後述）が仮定されていた.
  しかし，その実装方法は記載がなく不明だった.

- 本研究では，ランダム mod-p符号と Construction A を使い，計算機への実装を試みた
- Coarse格子を，n次元超立方体から，正単体に変化させた
  - （ねらい）格子点間の距離が改善するのではないか？

（結果）

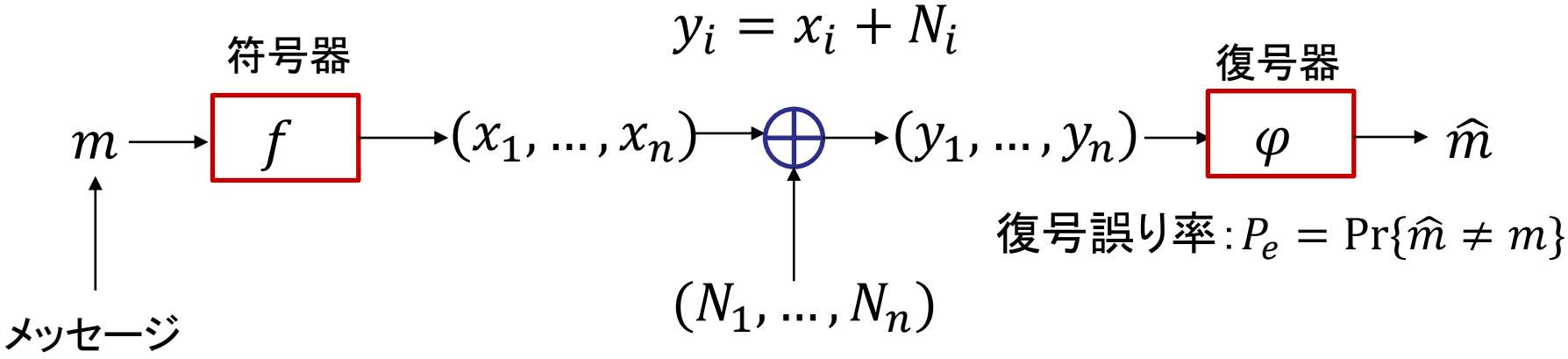
- 符号化法の選択を誤った．現実的な時間で復号できない.
  - 低次元に限れば，格子点間距離は正単体により改善した.
  - 今回のSNR設定でディザ信号の有無は誤り率に影響しない

# 入れ子格子符号(1/4)



1. ガウス型通信路

$N_i \sim \mathcal{N}(0, P_N)\ i.i.d.$




復号誤り率 : $P_e = \Pr\{\hat{m} \neq m\}$

電力制約: 符号語$\boldsymbol{x}(m) = x_1(m)x_2(m)\cdots x_n(m), m = 1,2,\dots,M$は次式を満たす. ($n$は符号長)

$$\sum_{i=1}^{n} x_i^2(m) \leq nP_X$$

伝送レート: $R = \frac{1}{n}\log M$

通信路容量 $C = \max_{p(x)\ s.t.\ E[X^2] \leq P_X} I(X;Y) = \frac{1}{2}\log(1 + SNR)$

SNR: Signal-to-Noise Ratio(信号対雑音比), $SNR = P_X/P_N$

# 入れ子格子符号(2/4)

## 格子（Lattice）

定義：$n$次元格子$\Lambda$はフルランク行列$G \in R^{n \times n}$を用いて次式で与えられる

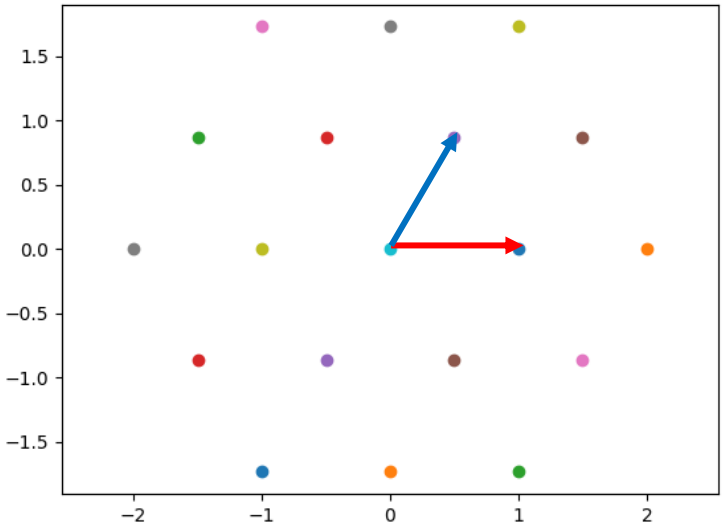
$$\Lambda = \{\boldsymbol{\lambda} = G\boldsymbol{j} : \boldsymbol{j} \in \mathbb{Z}^n\}$$

$$\boldsymbol{\lambda} = j_1\boldsymbol{g}_1 + j_2\boldsymbol{g}_2 + \cdots j_n\boldsymbol{g}_n$$

- Gは格子$\Lambda$の生成行列．($\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots \boldsymbol{g}_n$は線形独立)

2次元の例:

$$G = \begin{bmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix} = [\boldsymbol{g}_1, \boldsymbol{g}_2]$$

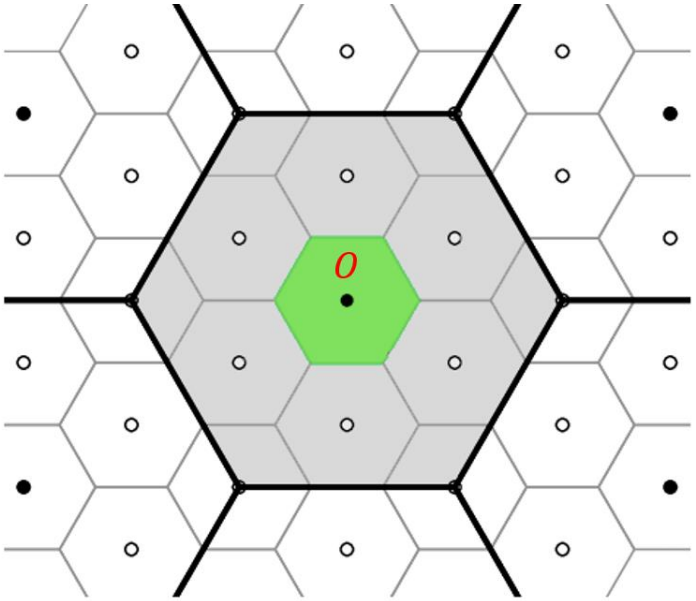


# 入れ子格子符号(3/4)

## 入れ子格子

- $\Lambda_c \subseteq \Lambda_f$を満たす格子ペアを入れ子格子と呼ぶ.
- 格子 $\Lambda_c$ は coarse 格子, 格子$\Lambda_f$は fine 格子と呼ばれる
- coarse格子のボロノイ領域に含まれる格子点を符号語とする.

- 格子そのものは，無限に続いている.
- 原点に近い格子点だけを符号語として選択しなくてはいけない.
- 入れ子格子: coarse格子によって符号語を選択する. ⇒現実的な時間で復号できる
- ボロノイ領域の境界線（面）は，一方は開区間，一方は閉区間.



# 入れ子格子符号(4/4)



## ユークリッドラティス復号

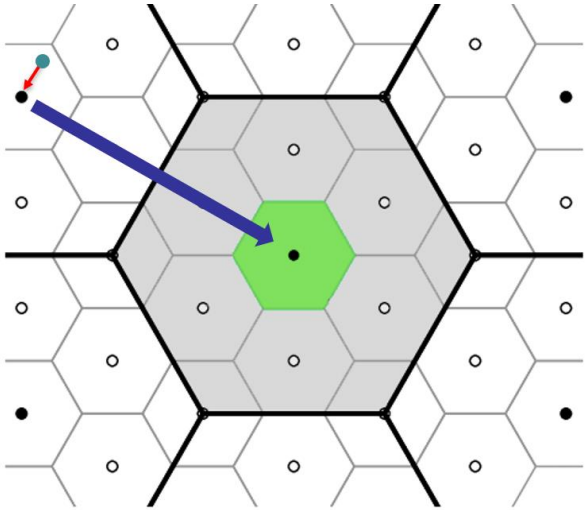
入れ子格子の対称性を利用したユークリッドラティス復号が仮定されている

$$q(\boldsymbol{Y}) = Q_{\mathcal{V}_f}(\boldsymbol{Y}) \bmod \Lambda_c$$

ここで，$Q_{\mathcal{V}_f}(\cdot)$は受信信号から一番近いfine格子点に復号すること
$\bmod \Lambda_c$は$\Lambda_c$に関するモジュロ演算を表す，$\boldsymbol{Y}$は受信信号．

ユークリッドラティス復号を仮定すると，復号誤り確率は送信された符号語にかかわらず

$$P_e = \Pr(\boldsymbol{N} \notin \mathcal{V}_f)$$



ここで，N はガウス雑音である．

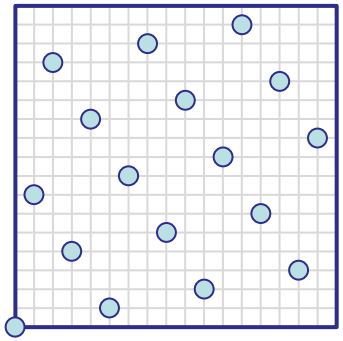
真の最尤復号では，符号語によって復号誤り率が異なる

## ランダムmod-p格子

- fine格子は，ランダムに生成したベクトル$\boldsymbol{g}$に対し，$\boldsymbol{g}$の整数倍(mod $p$)となる点の集合

例. $n = 2,\ p = 17$

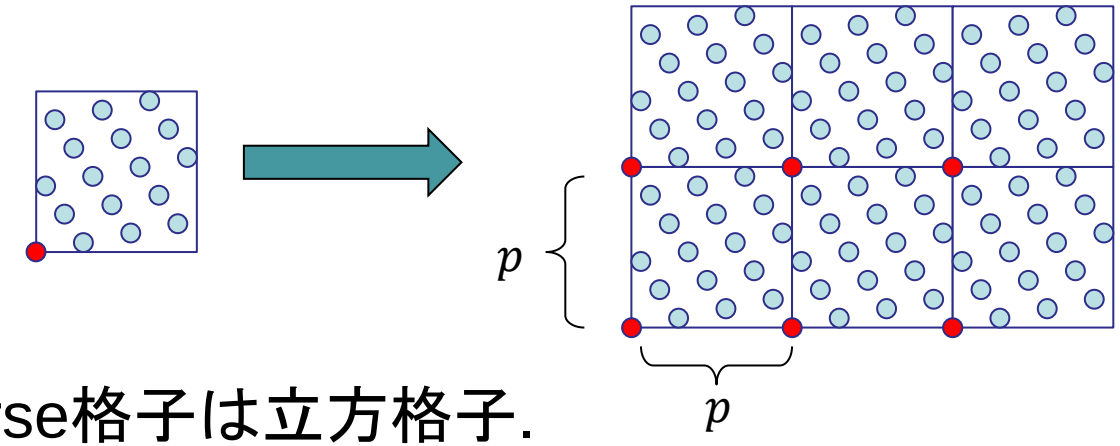
生成ベクトル $\boldsymbol{g} = [3, 4]^T$



- Construction A : GF($p$)上の線形符号$C$を使って，

$$\Lambda = C + p\mathbb{Z}^n$$

とする.




このときCoarse格子は立方格子.

# 提案法



今回は正単体の構造を利用してランダムmod-p格子を変形する

- ErezとZamirのランダムmod-p格子のcoarse 格子は立方格子だった.
- 正単体は立方格子よりは良い選択ではないか？
- 正単体は$n \geq 4$では一般に最密ではないが，$n$ごとに構成法を考えるのは大変.
- 単体格子は，任意の次元で構成できる.

提案した正単体化したランダムmod-p格子の性能を数値実験で検証

- 本来は，ランダムmod-p 格子の符号アンサンブルに関する平均を議論する必要がある.
- 今回は，数値実験を行い，以下の指標を評価した.

1. 符号間最小距離
2. 復号誤り率

# 正単体

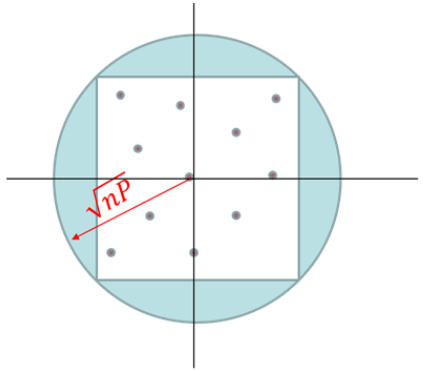


- 正単体は、2 次元の正三角形、3 次元の正四面体、4 次元の正五胞体を $n$次元に一般化した正多胞体である.

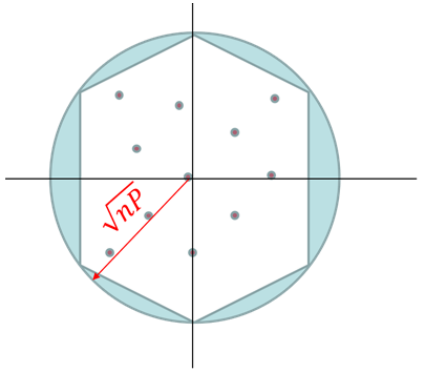
**電力制約**

電力制約に対し, 単体格子の方が効率が良い

体積を同じにした時, 正単体の辺長が立方体より長くなる.

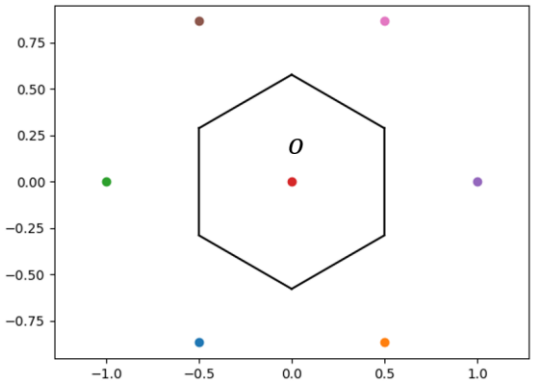


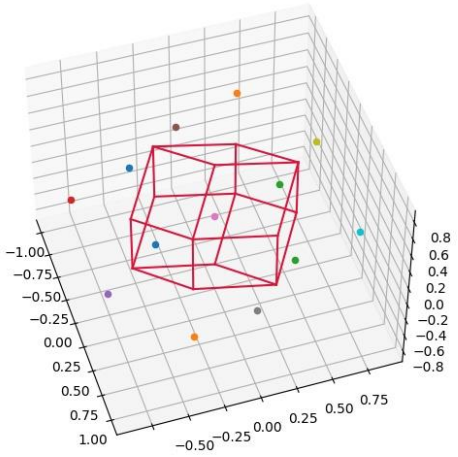

coarse 格子: 立方格子




coarse 格子: 単体格子

**2, 3次元の正単体格子とそのボロノイ領域**



# 提案法

## 符号語の作り方

論文[1]によるランダムmod-p格子:
(以降，立方格子と呼ぶ)

1. $g_i \sim \mathrm{Unif}(0, \ldots, p-1)\ i.i.d., i = 1, \ldots, n$のベクトル$\boldsymbol{g}$を生成する($p$は素数).

2. コードブックを定義する:

$$\mathcal{C}' = \{\boldsymbol{x} \in \mathbb{Z}_p^n : \boldsymbol{x} = (\boldsymbol{g} \cdot q) \bmod p, \\ q = 0, \ldots, p-1\}$$

3. $\mathcal{C}'$を$\mathbb{R}^n$に拡張する:

fine 格子: $\Lambda_f' = \mathcal{C}' + p\mathbb{Z}^n$

coarse 格子: $\Lambda_c' = p\mathbb{Z}^n$

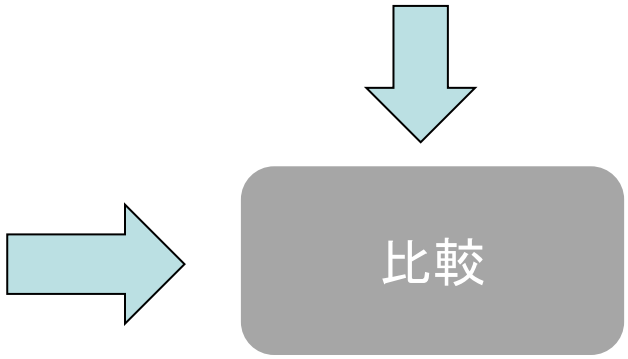
提案した正単体mod-p格子
(以降，単体格子と呼ぶ)

ランダムmod-p格子を単体化する

fine格子 $\Lambda_f = T \cdot \Lambda_f'$

coarse格子 $\Lambda_c = T \cdot \Lambda_c'$

Tは正単体格子の生成行列



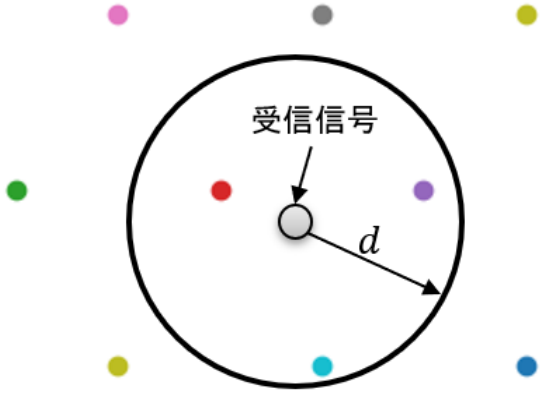
# 復号法について



- **ErezとZamirは，ユークリッドラティス復号を仮定していた.**
- **しかし，その実装方法は記載がなく不明だった.**
- **今回は，スフィア復号を使った.**

## スフィア復号

本研究では雑音が小さいときに比較的
高速に厳密解を計算できるスフィア復号を利用する.




処理の流れ

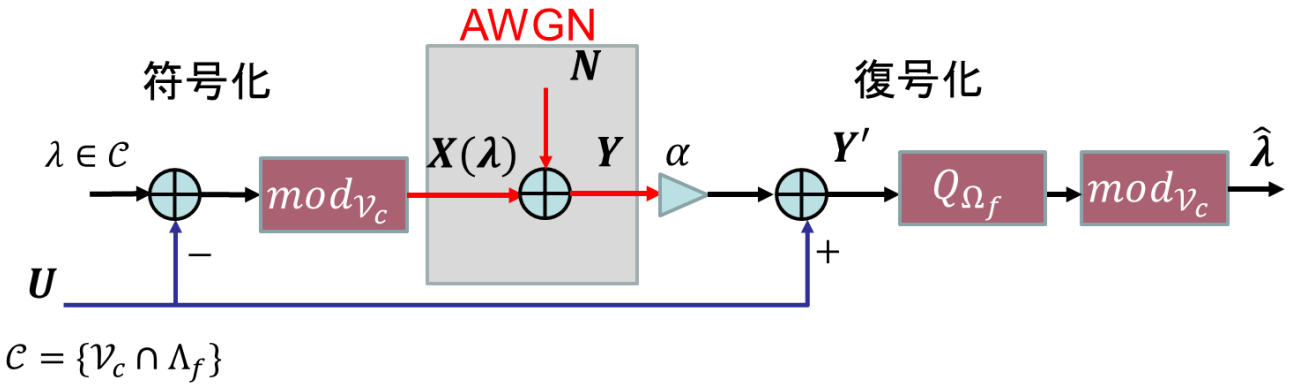
1. 適当な半径d > 0 を設定する
2. 受信信号から半径d 以内の一番近い整数ベクトル$\boldsymbol{j}$ を出力する

$$\min_{\{\boldsymbol{j}\in\mathcal{Z}^m:\|\boldsymbol{y}-G\boldsymbol{j}\|^2\le d^2\}} \|\boldsymbol{y}-G\boldsymbol{j}\|^2$$

今回は，$d=\|\boldsymbol{y}-G\hat{\boldsymbol{j}}\|$とした．$\hat{\boldsymbol{j}}=\mathrm{round}(G^{-1}\boldsymbol{y})$はBabai推定

# 実験1

## dither信号を加えた通信路




- $U$は共通のdither信号．Erez-Zamirが示した符号化法に登場する．
- Erez-Zamir は，この符号化法を使って，通信路容量を達成できることを証明した．

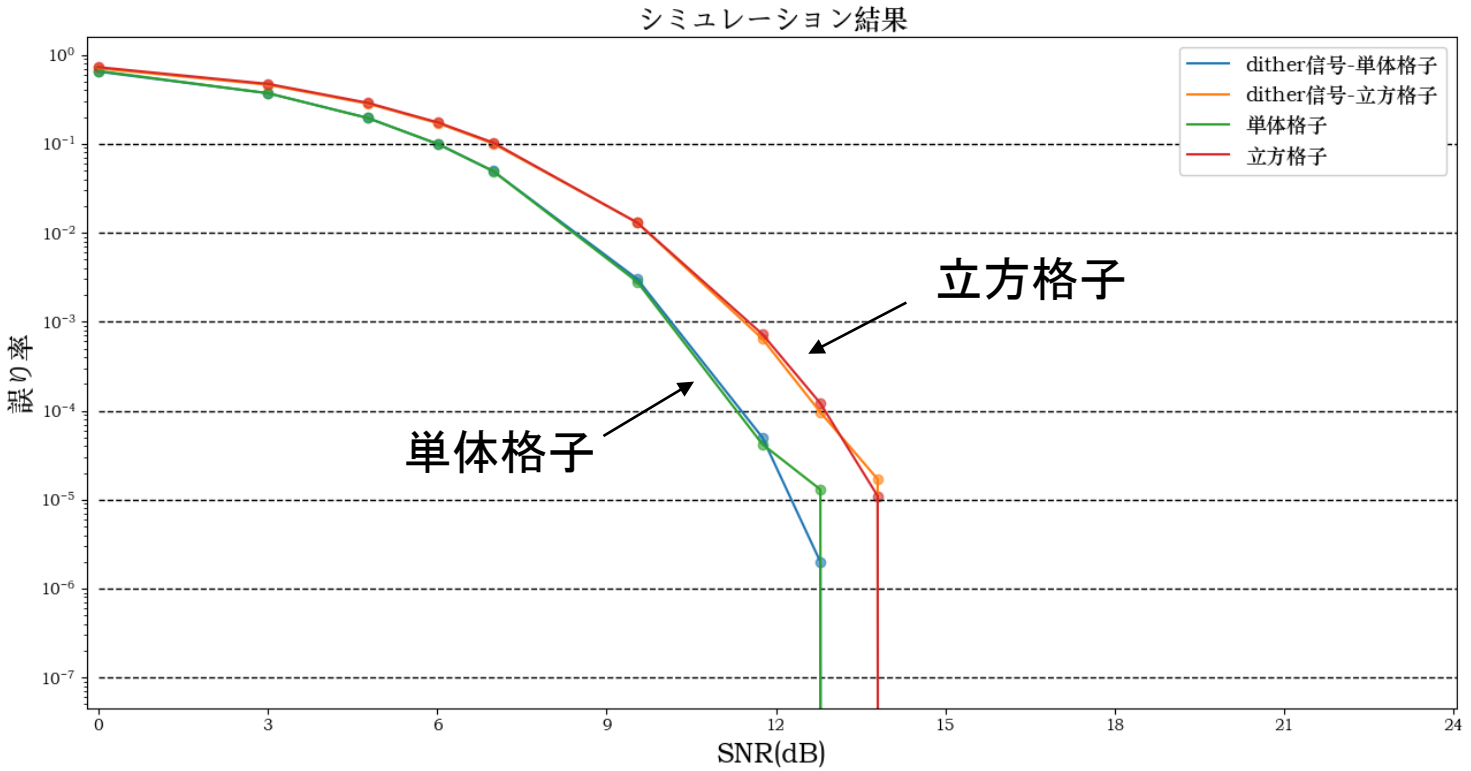
# 実験1の結果



## ディザ信号の有無による復号誤り率の比較

設定: $n = 6, p = 37$

生成ベクトル$g$はこの設定で最小距離最大の$g$




- 大きな違いは見られない.
- 以後の実験ではディザ信号は使用しない.

# 実験2

## 符号間最小距離の比較

- 符号語の平均電力が等しくなるよう正規化し，符号語の最小距離を計測
- 符号間最小距離は生成ベクトル$g$ に依存するため，今回は各設定で10,000 回符号語を作って実験を行った.
- その中で，最小距離の最大値とそのときの生成ベクトル$g$を記録.

| | | 立方格子 | | 単体格子 | |
|---|---|---|---|---|---|
| 次元数 | 符号語の数 | 最大のベクトル$g$ | 最小距離 | 最大のベクトル$g$ | 最小距離 |
| $n=4$ | $p=37$ | [1,8,26,31] | 14.90 | [1,29,31,33] | 17.32 |
| $n=4$ | $p=257$ | [1,22,178,187] | 66.97 | [1,30,57,192] | 74.60 |
| $n=6$ | $p=37$ | [1,2,12,21,22,31] | 20.17 | [1,4,11,13,17,27] | 23.55 |
| $n=6$ | $p=257$ | [1,44,55,76,91,93] | 102.08 | [1,9,12,36,97,194] | 117.89 |

他の$(n,p)$の値でも実験を行った

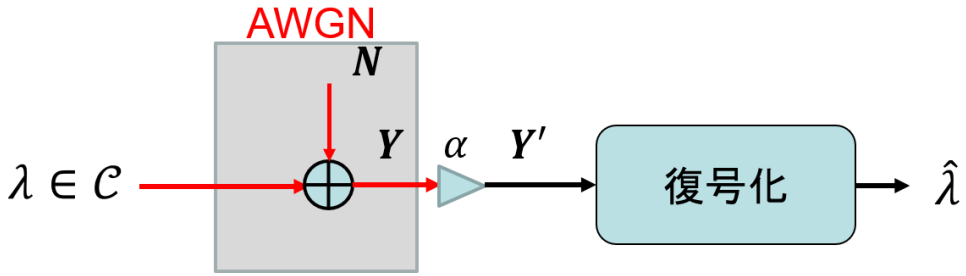
最小距離の最大値と平均値は，単体格子のほうが大きかった.

# 実験3

## 復号誤り率の測定

ランダムに生成した符号語と最小距離最大の符号語について復号誤り率を
数値シミュレーションにより評価する.

符号語 $\lambda$ を等確率にひとつ選び, 送信する,
受信側は受信信号Y と減衰率 $(0 < \alpha < 1)$ をかけて入れ子格子の復号化をする.
(Erez-Zamirの復号法)




復号:

$$\boldsymbol{Y}' = \alpha \boldsymbol{Y} = \alpha(\boldsymbol{\lambda} + \boldsymbol{N})$$
$$\hat{\boldsymbol{\lambda}} = Q_{sf}(\boldsymbol{Y}') - Q_{sc}(\boldsymbol{Y}')$$

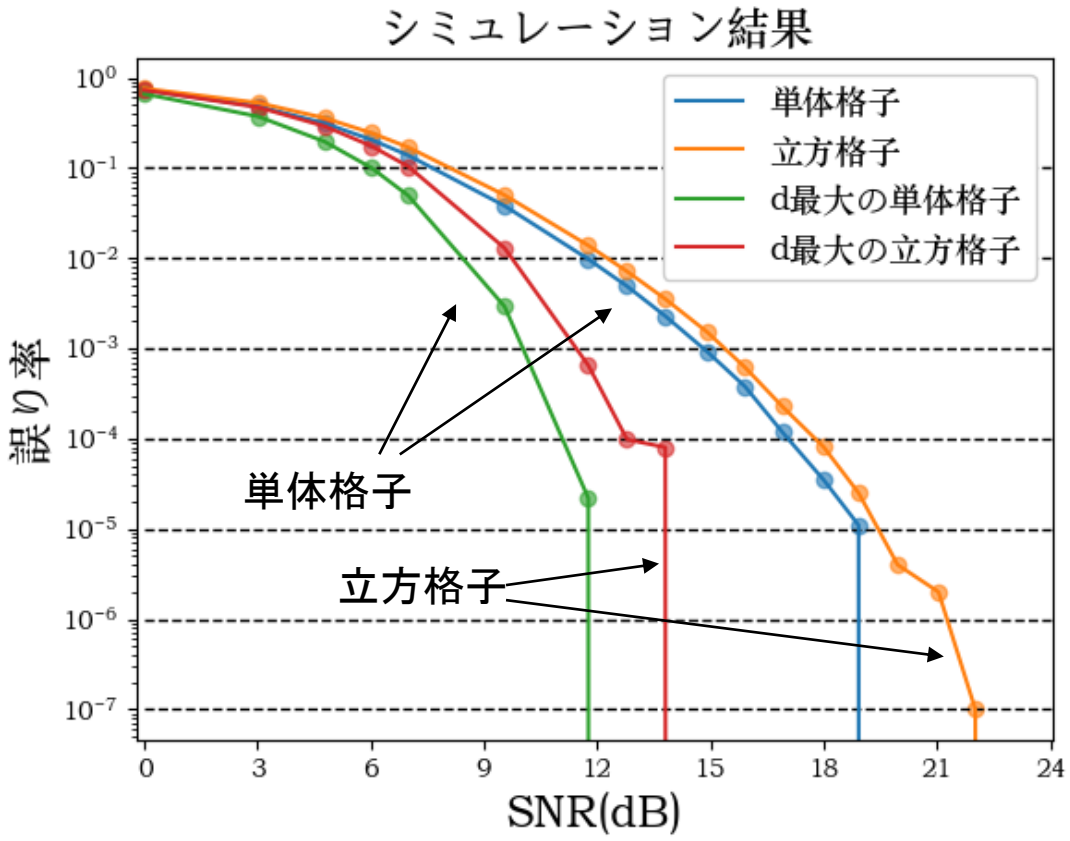
$\boldsymbol{N}$ はガウス雑音のn 次元ベクトル
$Q_{sf}$ はfine 格子に対するスフィア復号
$Q_{sc}$ はcoarse格子に対するスフィア復号
減衰率$\alpha = \mathrm{SNR}/(1 + \mathrm{SNR})$に設定する

# 実験3の結果

## 復号誤り率

設定: $n = 6, p = 37$

各SNR に対し10,000 回のシミュレーション



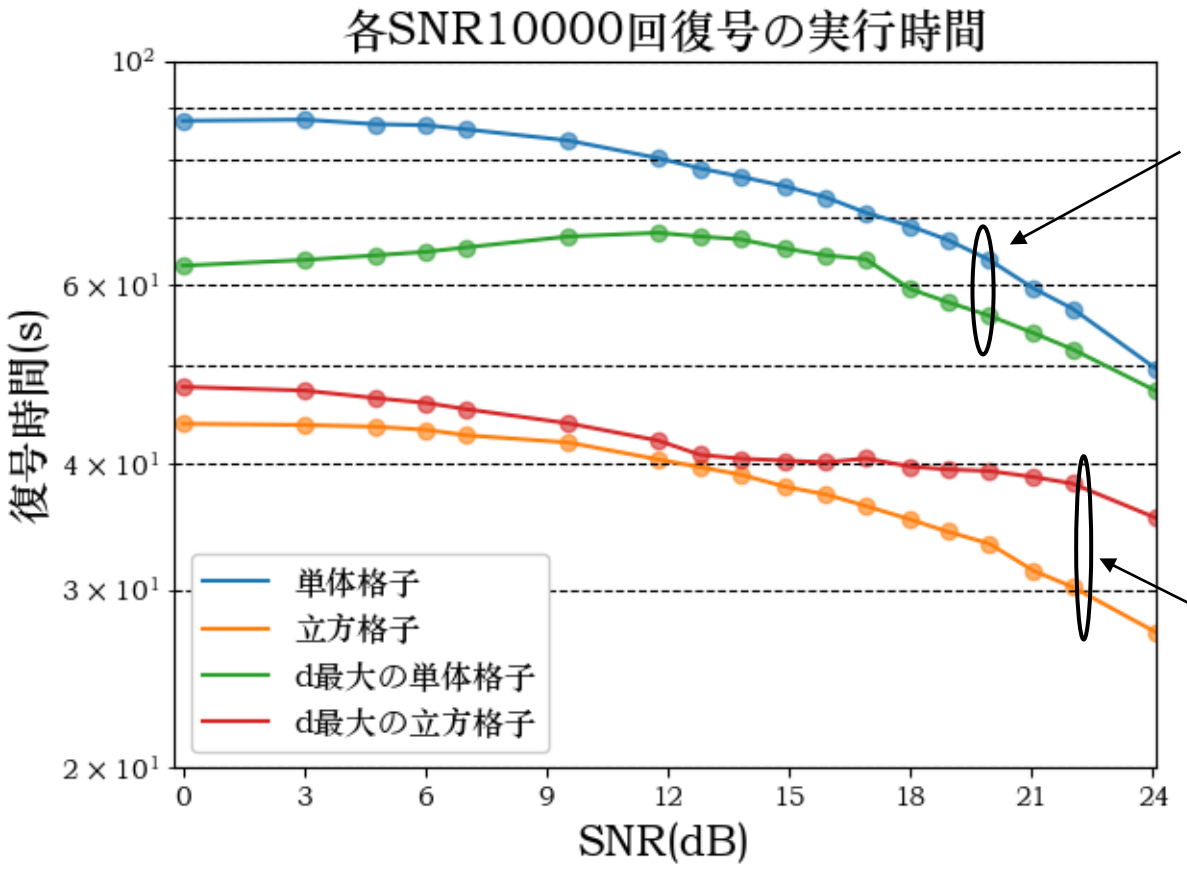

単体格子の誤り率は立方格子に比べ改善している

最小距離が一番大きな符号語を選択した場合は，復号誤り率が平均値よりかなり小さくなっている

# 実験3の結果

## 実行時間

各SNR に対する10000 回の実行時間の総和




単体格子

単体格子は立方格子より1.5倍以上の時間がかかっている

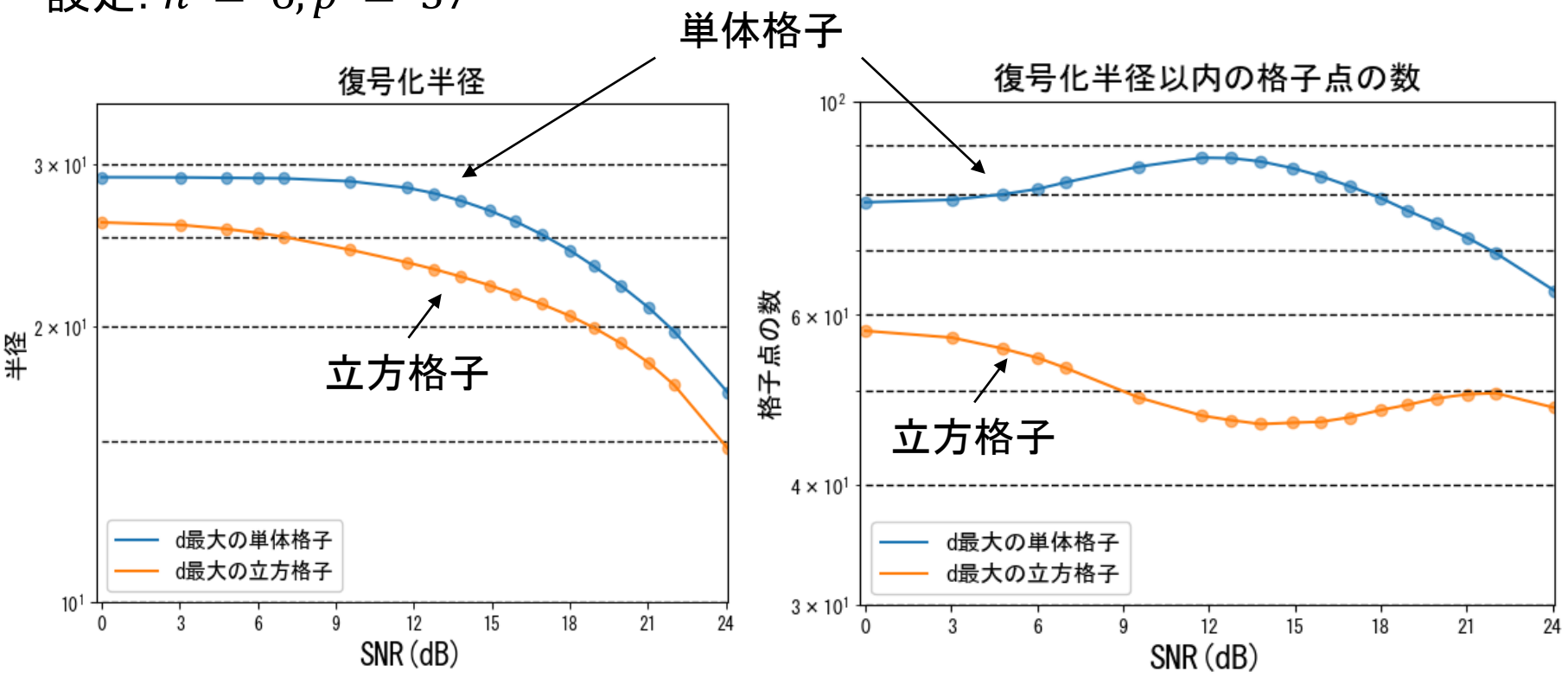
立方格子

# 実験3：計算時間増大の原因探索

## スフィア復号における探索距離 d と 格子点の数

最小距離最大の単体格子と立方格子について
各SNRで10000回シミュレーション，復号化の半径dと半径内の格子点の数の平均値を記録する

設定: $n = 6, p = 37$




単体格子が立方格子より時間がかかる理由は、復号化半径が大きいため

# LLLアルゴリズムで生成行列を変換する

# まとめ & 今後の課題



- 入れ子格子符号の符号化・復号化実験を行った
  - 符号化: ランダム mod-p と Construction A
  - 復号化 : スフィア復号

- ディザ信号の必要性を調査した
  - 結果 : 今回のSNR設定でディザがあっても無くても復号誤り率は同じ
- 正単体Coarse格子によって，超立方体Coarse格子を改善できるか検討した.
  - 正単体格子 : 復号誤り率は改善したが，復号にかかる計算時間は悪化した.
  - 実験は，$n = 4,6$までしか行えなかった. →符号の構成法が悪い.
- 今後は，効率的に復号化できる入れ子格子符号の構成法を検討する.

# 実験について

## SNR設定

符号語の平均エネルギー
$\bar{P}_X = (1/p)\sum_{\boldsymbol{\lambda}\in\mathcal{C}}\|\boldsymbol{\lambda}\|^2$

雑音のエネルギー
$\bar{P}_N = \|\boldsymbol{N}\|^2$

$$SNR = \bar{P}_X/\bar{P}_N$$

SNR設定値
1; 2; 3; 4; 5; 9; 15; 19; 24; 31;39; 49; 63; 78; 99; 127; 159; 255