ラティス暗号入門

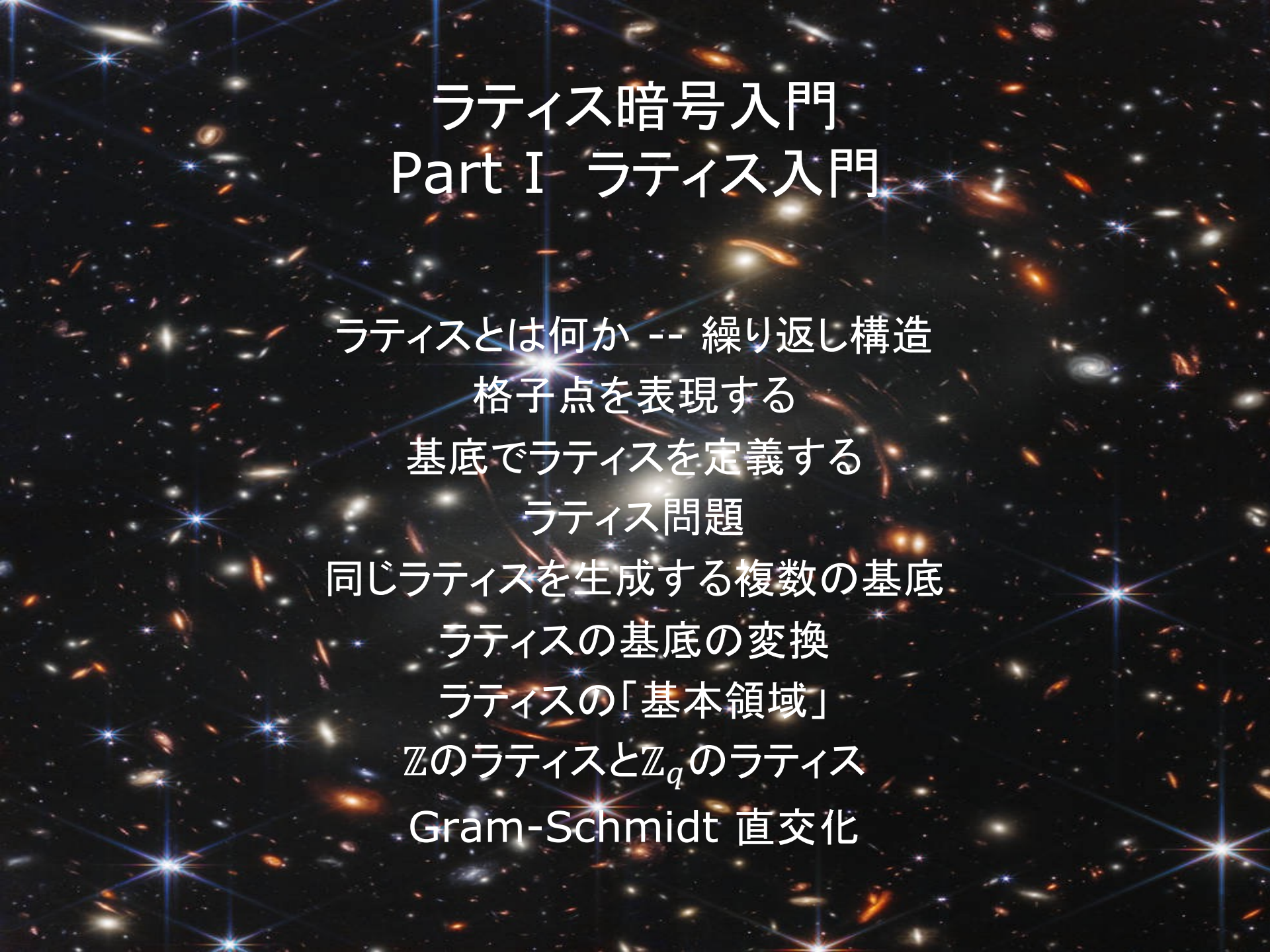
ラティス暗号入門
Agenda

Part I　ラティス入門

Part II　ラティス暗号 LWE

Part III　ラティスとラティス暗号

© NASA, ESA/CSA/Judy Sch



ラティス暗号入門
Part I　ラティス入門

ラティスとは何か -- 繰り返し構造

格子点を表現する

基底でラティスを定義する

ラティス問題

同じラティスを生成する複数の基底

ラティスの基底の変換

ラティスの「基本領域」

$\mathbb{Z}$のラティスと$\mathbb{Z}_q$のラティス

Gram-Schmidt 直交化

ラティス暗号入門
Part II　ラティス暗号 LWE

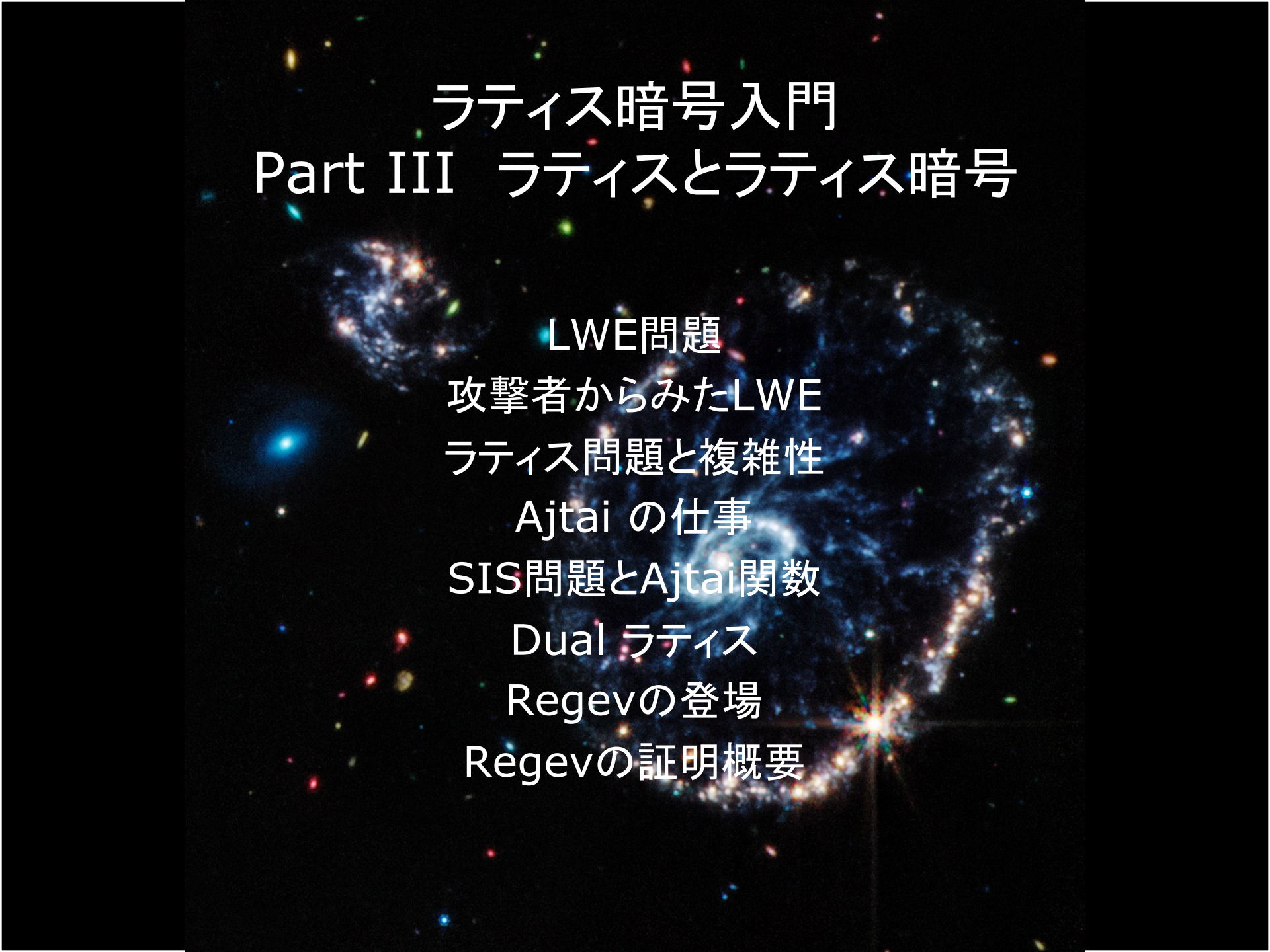
単純な例で学ぶ LWE (1)

単純な例で学ぶ LWE (2)

単純なサンプルからLWEへ -- 準備編

LWEの基本的なプロトコル

エラー項の役割



ラティス暗号入門
Part III　ラティスとラティス暗号

LWE問題

攻撃者からみたLWE

ラティス問題と複雑性

Ajtai の仕事

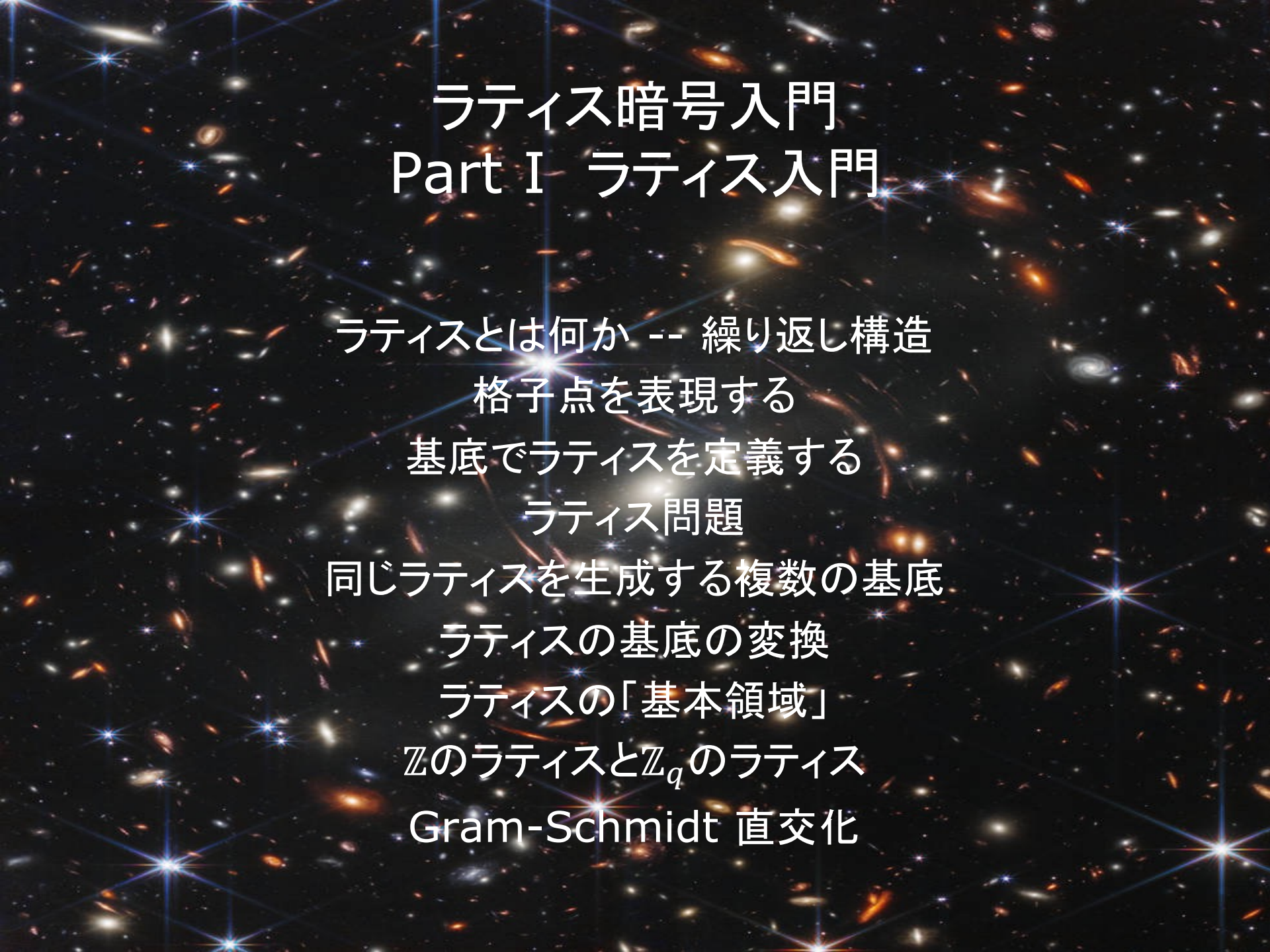
SIS問題とAjtai関数

Dual ラティス

Regevの登場

Regevの証明概要


Part I
ラティス入門



ラティス暗号入門
Part I　ラティス入門

ラティスとは何か -- 繰り返し構造

格子点を表現する

基底でラティスを定義する

ラティス問題

同じラティスを生成する複数の基底

ラティスの基底の変換

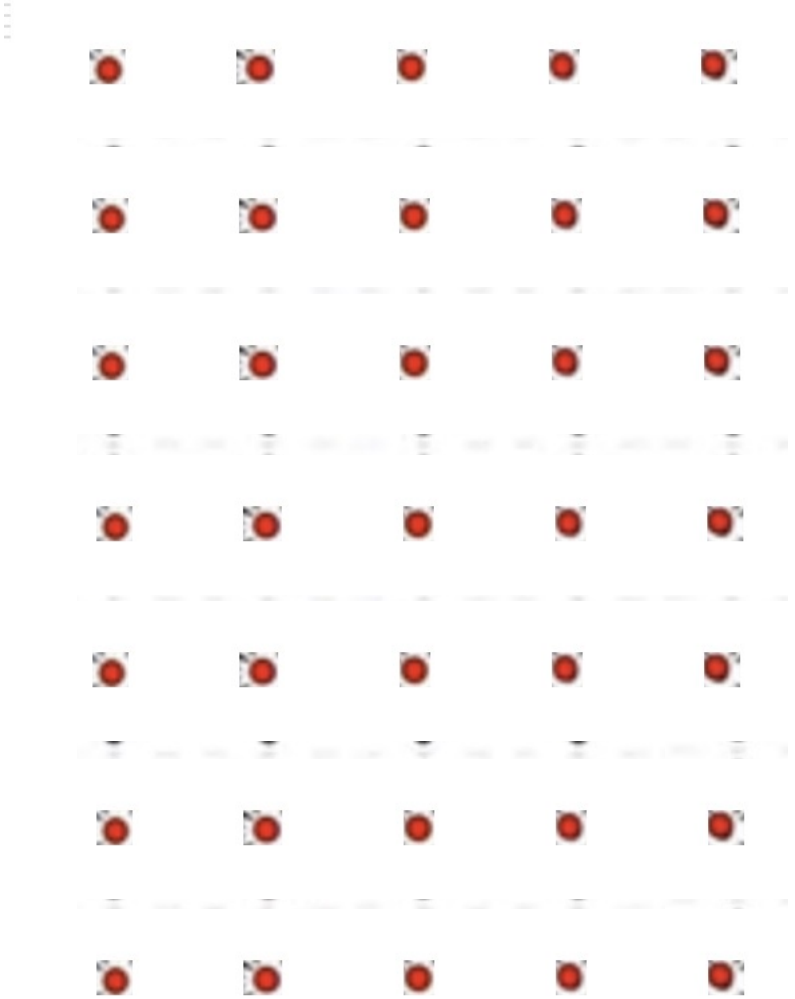
ラティスの「基本領域」

$\mathbb{Z}$のラティスと$\mathbb{Z}_q$のラティス

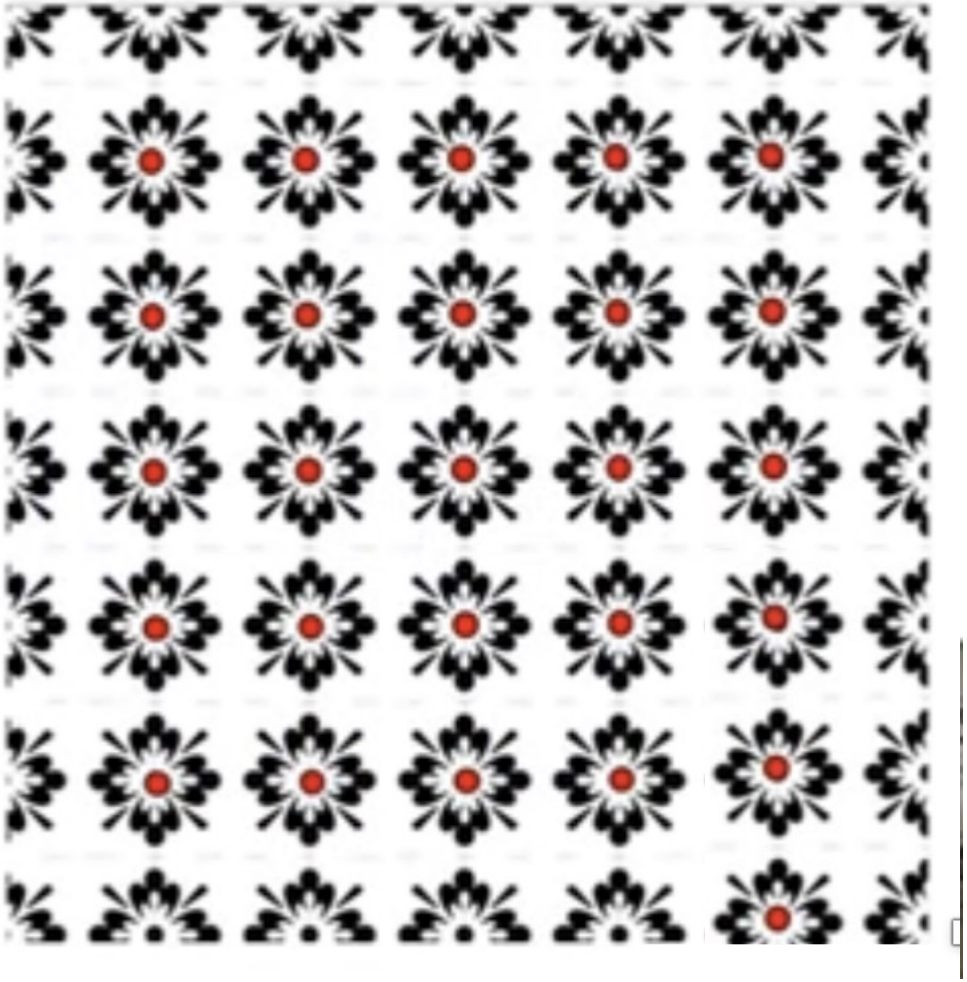
Gram-Schmidt 直交化

ラティスとは何か -- 繰り返し構造

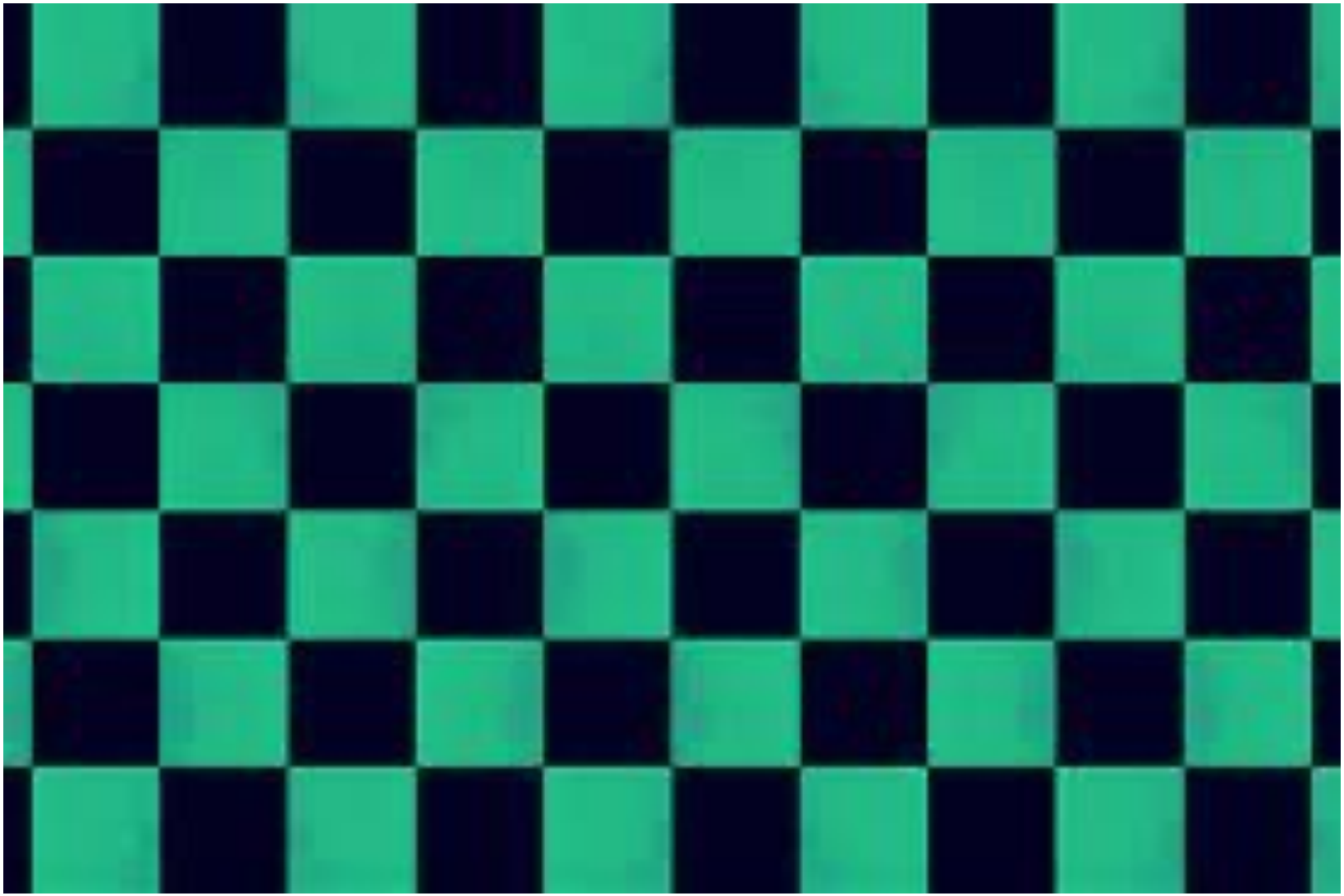
ラティス



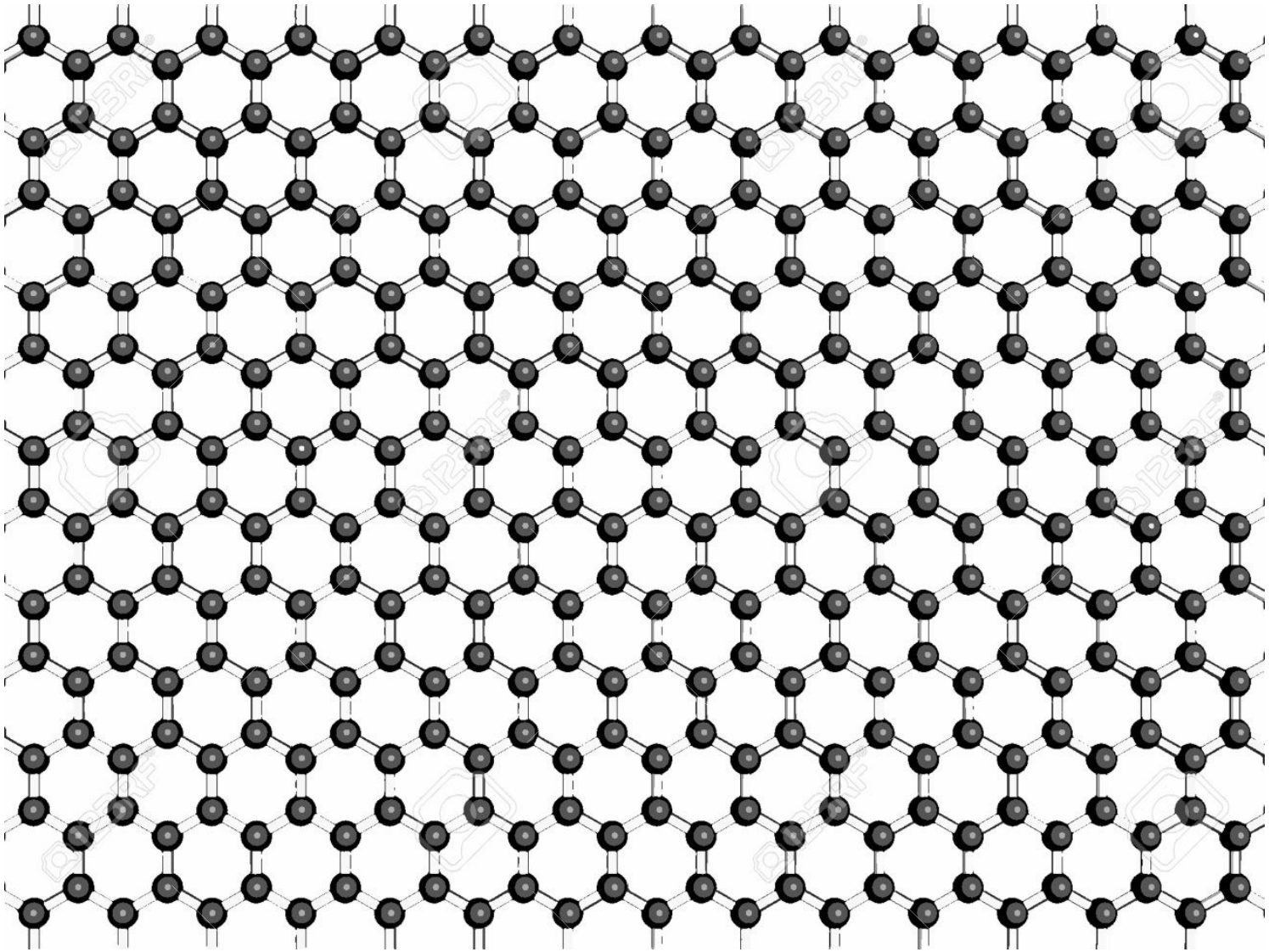
ラティスの繰り返し構造（二次元）



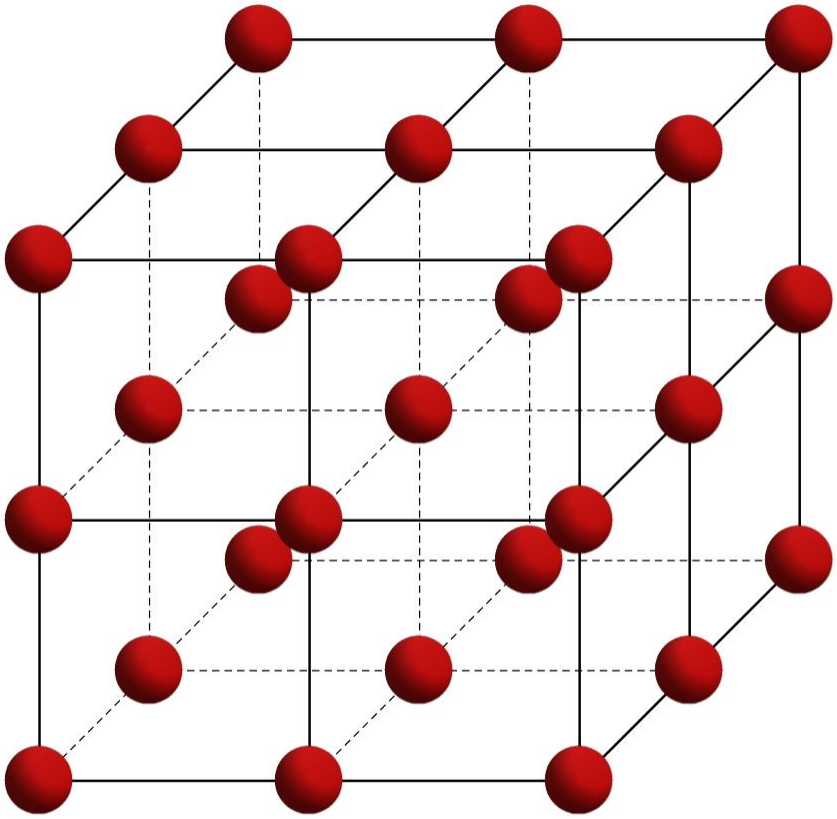
市松模様



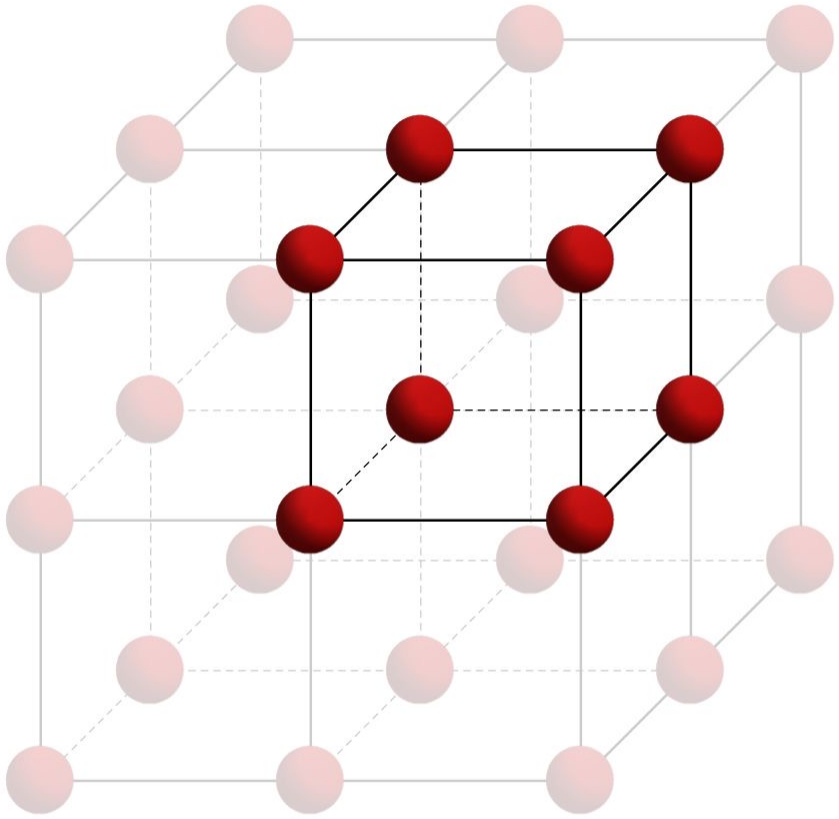
Graphene



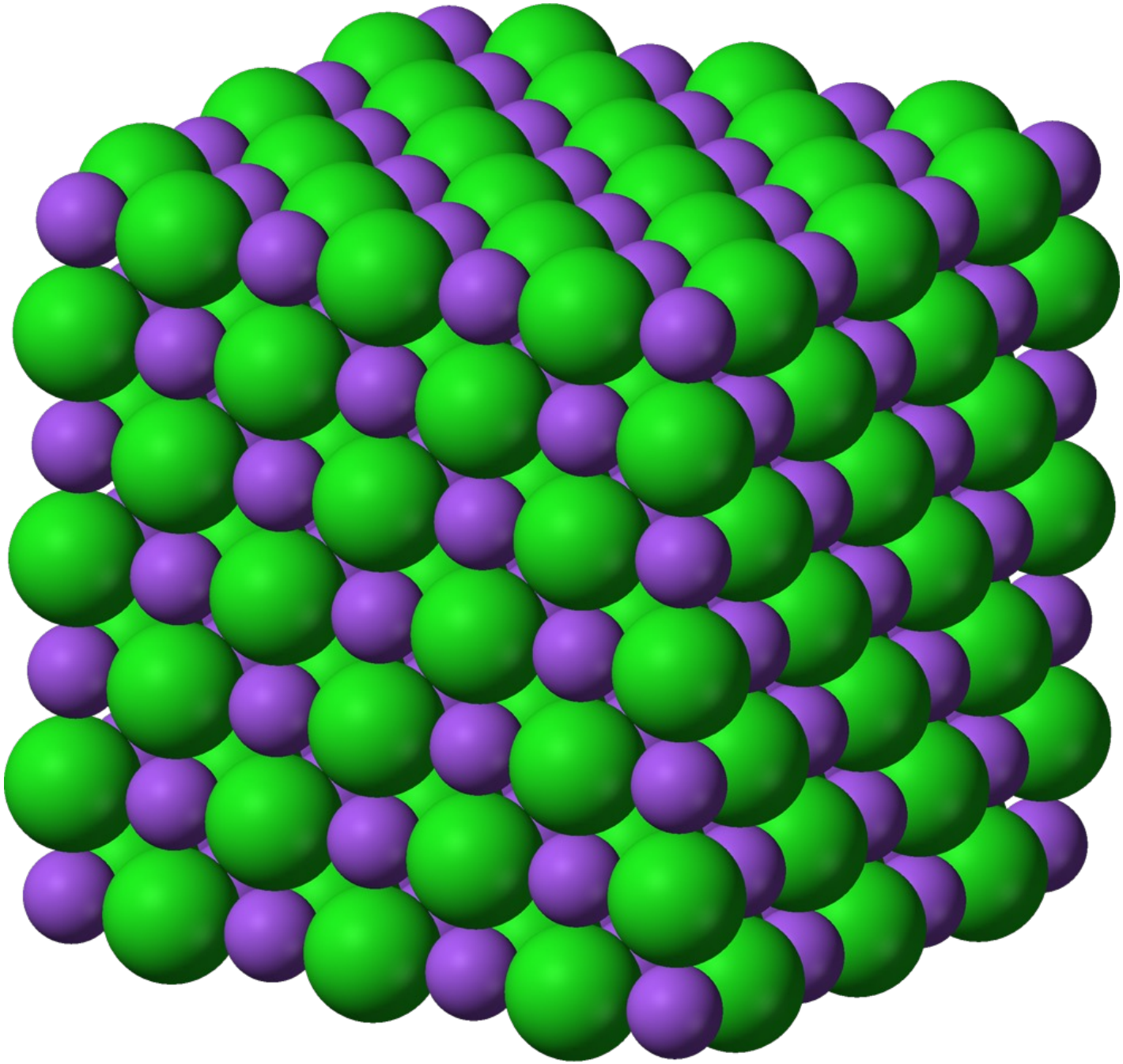
ラティスの繰り返し構造（三次元）



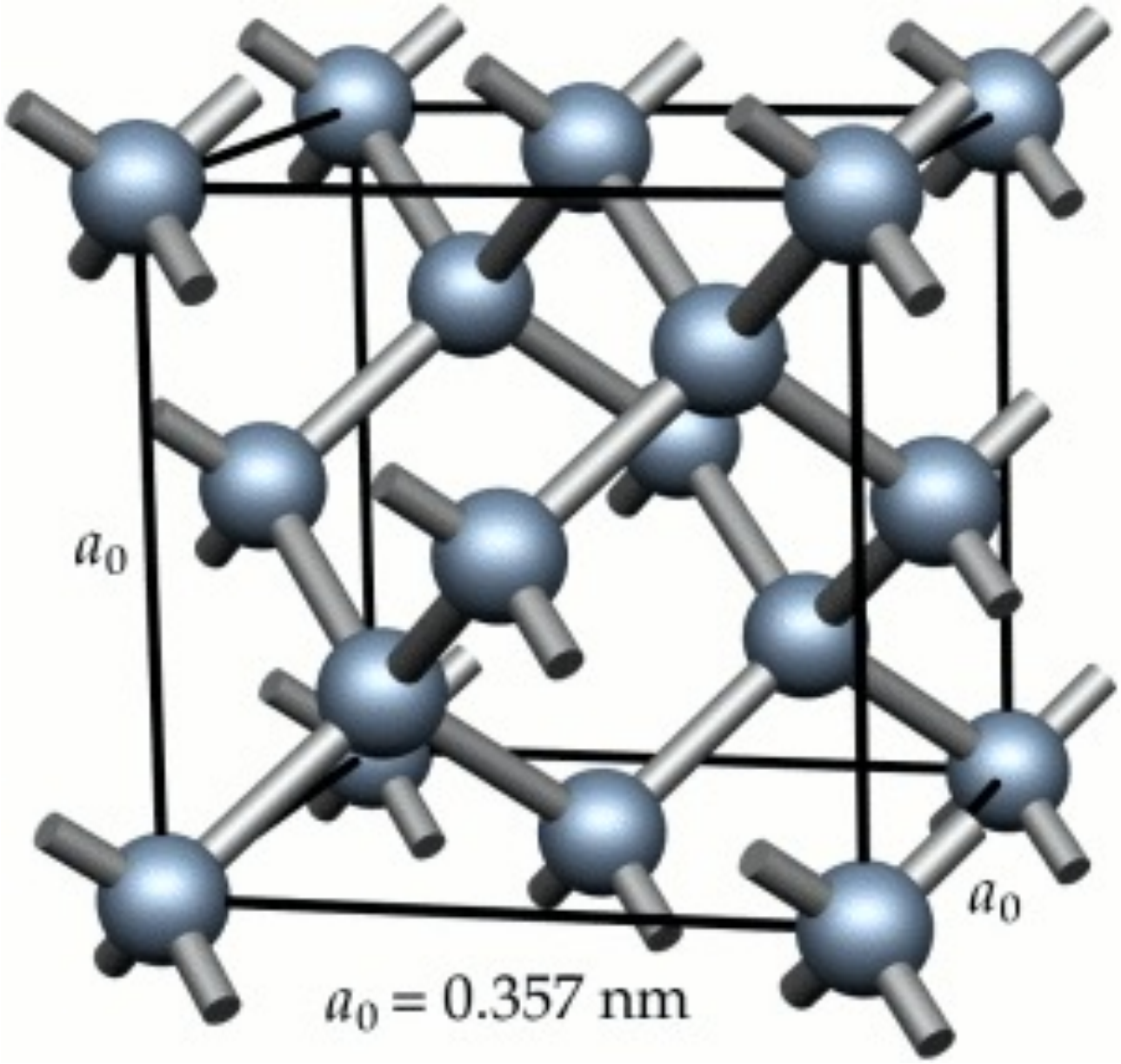
lattice structure

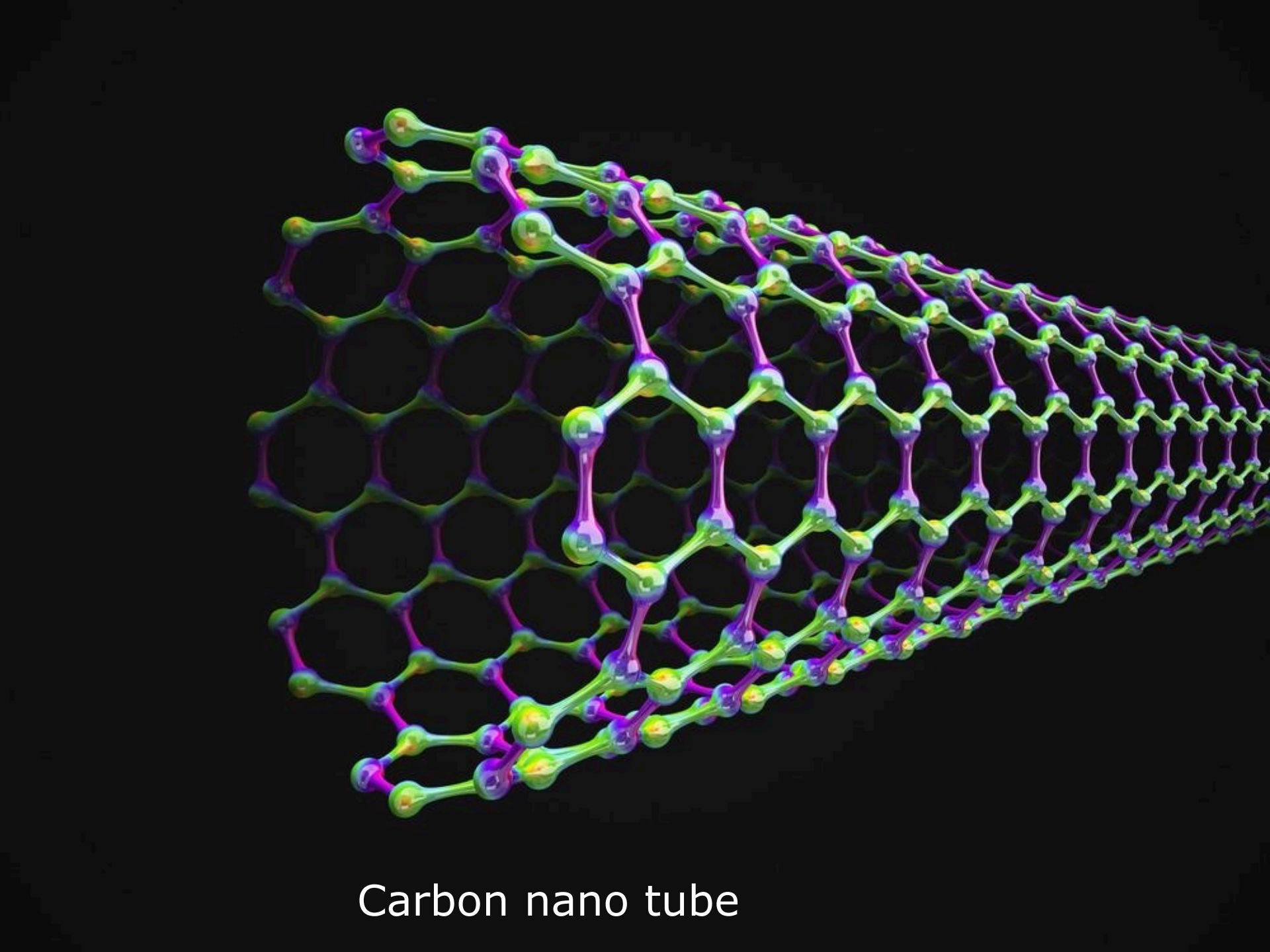


unit cell

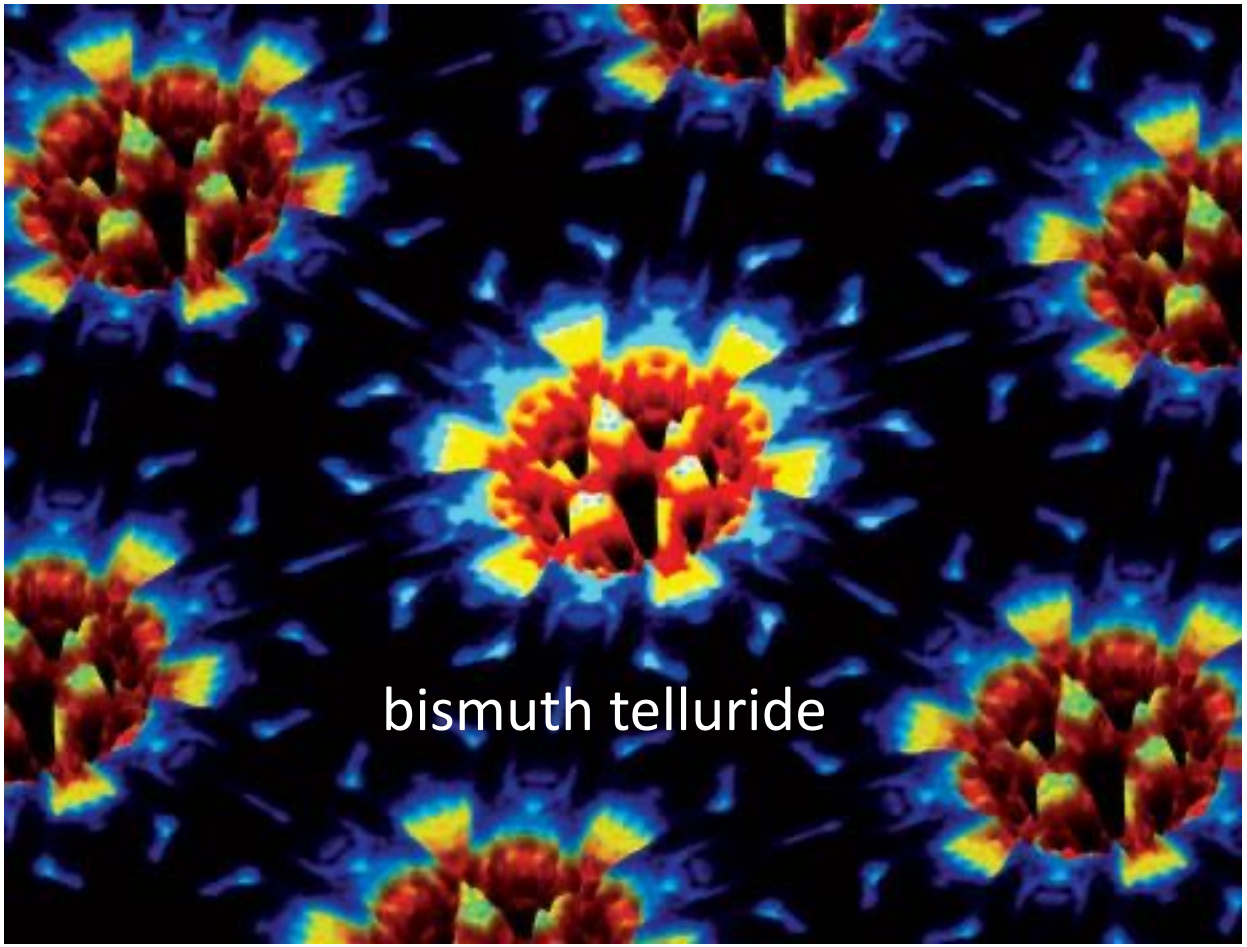


Diamond

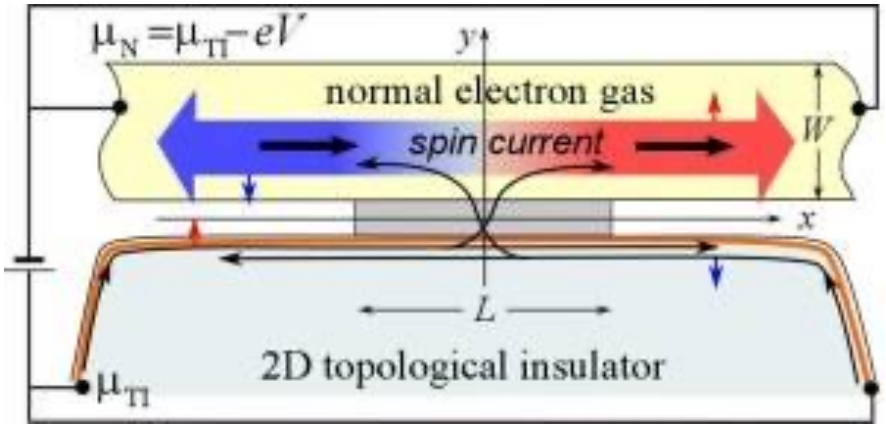




Carbon nano tube



Topological Insulators



nature

THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE

Time crystals

First observations of exotic new state of matter PAGES 164, 185, 217 & 221

BEHAVIOUR

COLLECTIVE AMNESIA

How social media and fake news are rewriting history

PAGE 168

APPLIED PHYSICS

COHERENT STRATEGY

Ways to commercialize quantum computers

PAGE 171

ARCHAEOLOGY

TRACING THE SILK ROAD

Iconic trade route arose from nomadic herding network

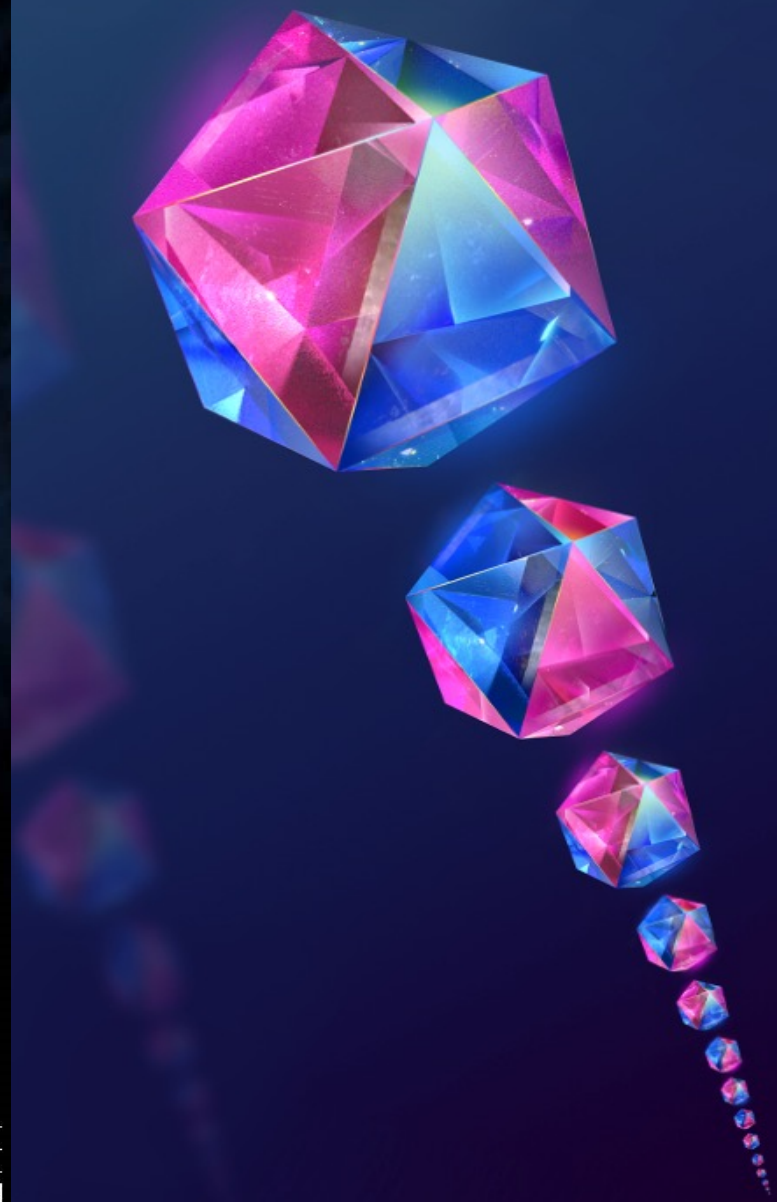
PAGES 188 & 193

NATURE.COM/NATURE

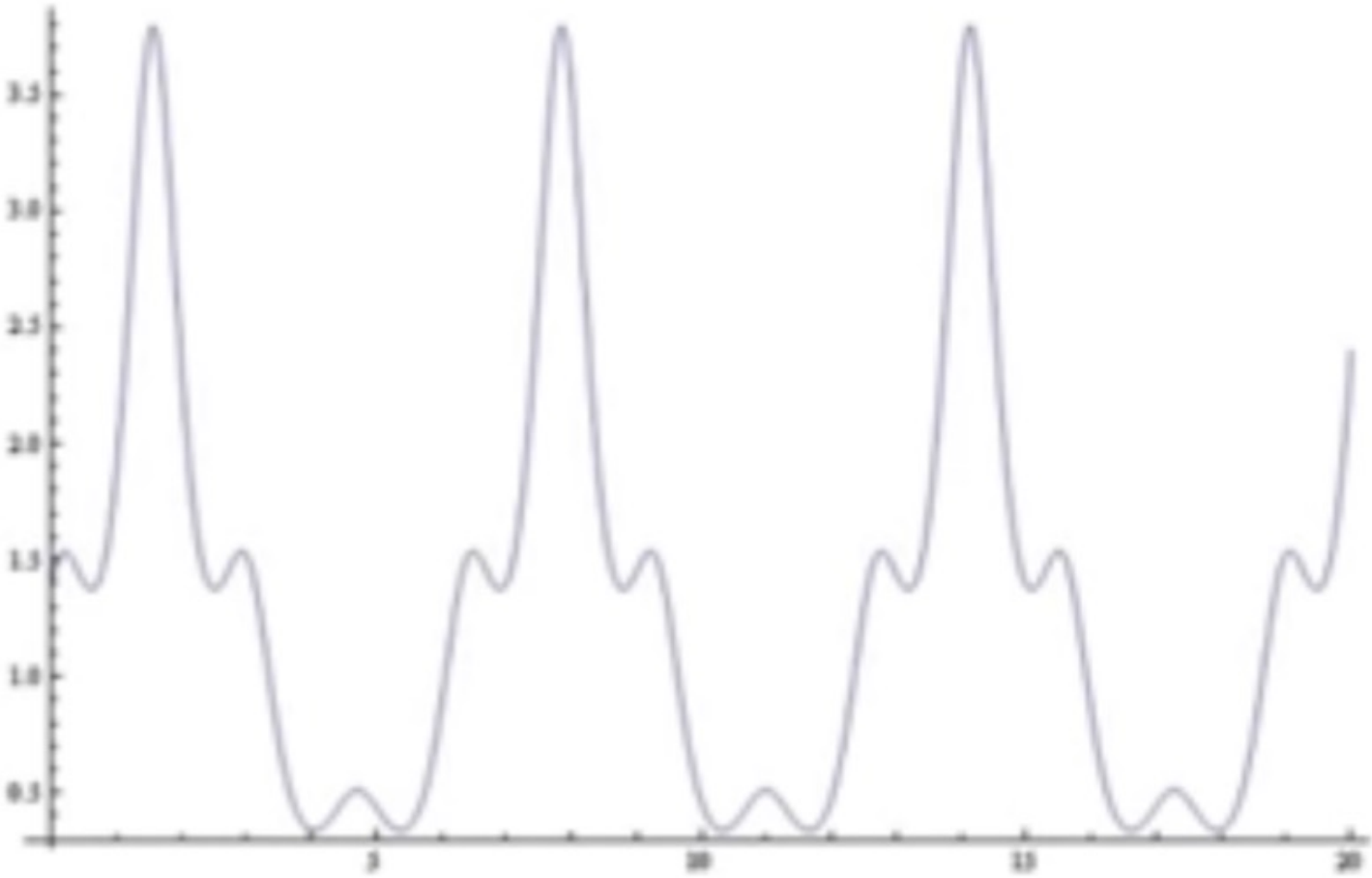
9 March 2017 £10

Vol. 543, No. 7644

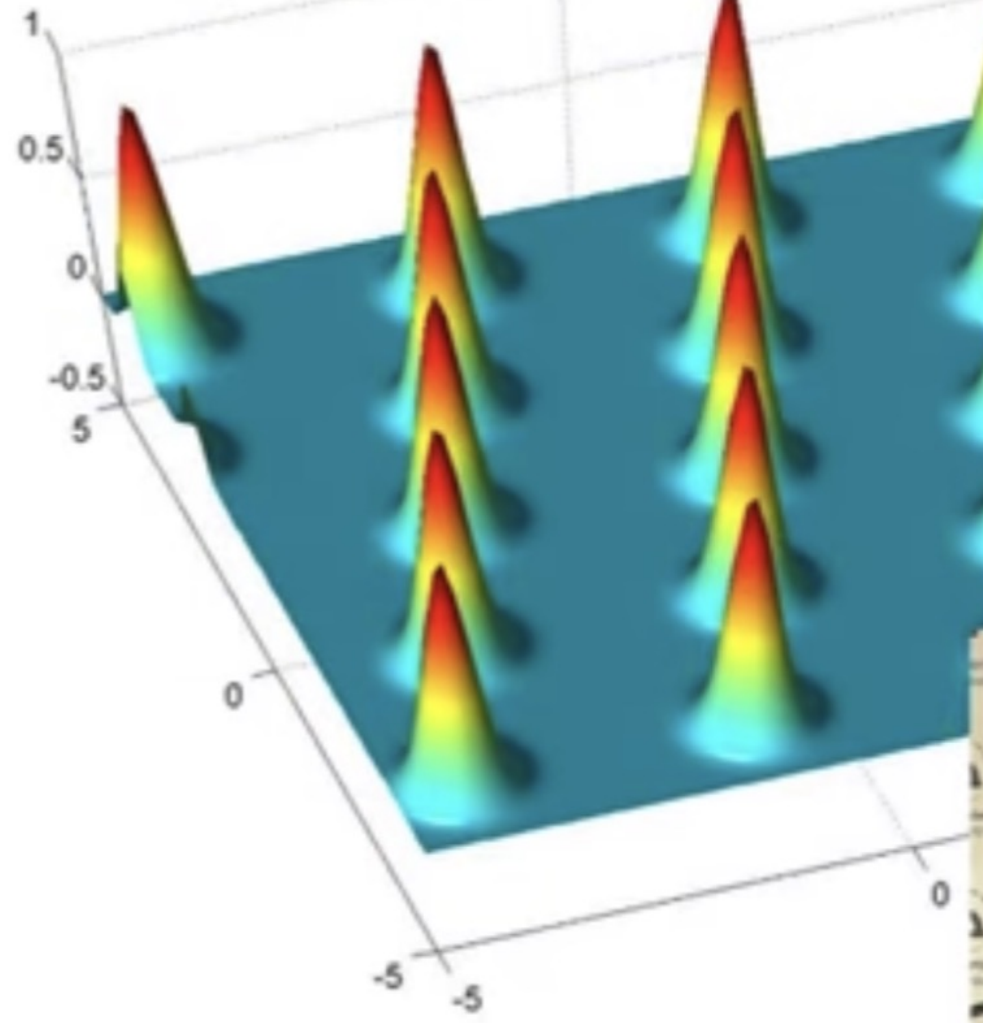
9 770028 083095 7 D



ラティスの繰り返し構造（一次元）
周期関数



$$f_{2/r}^{0,0} = f_{2/r}$$



格子点を表現する

ラティスとは何か

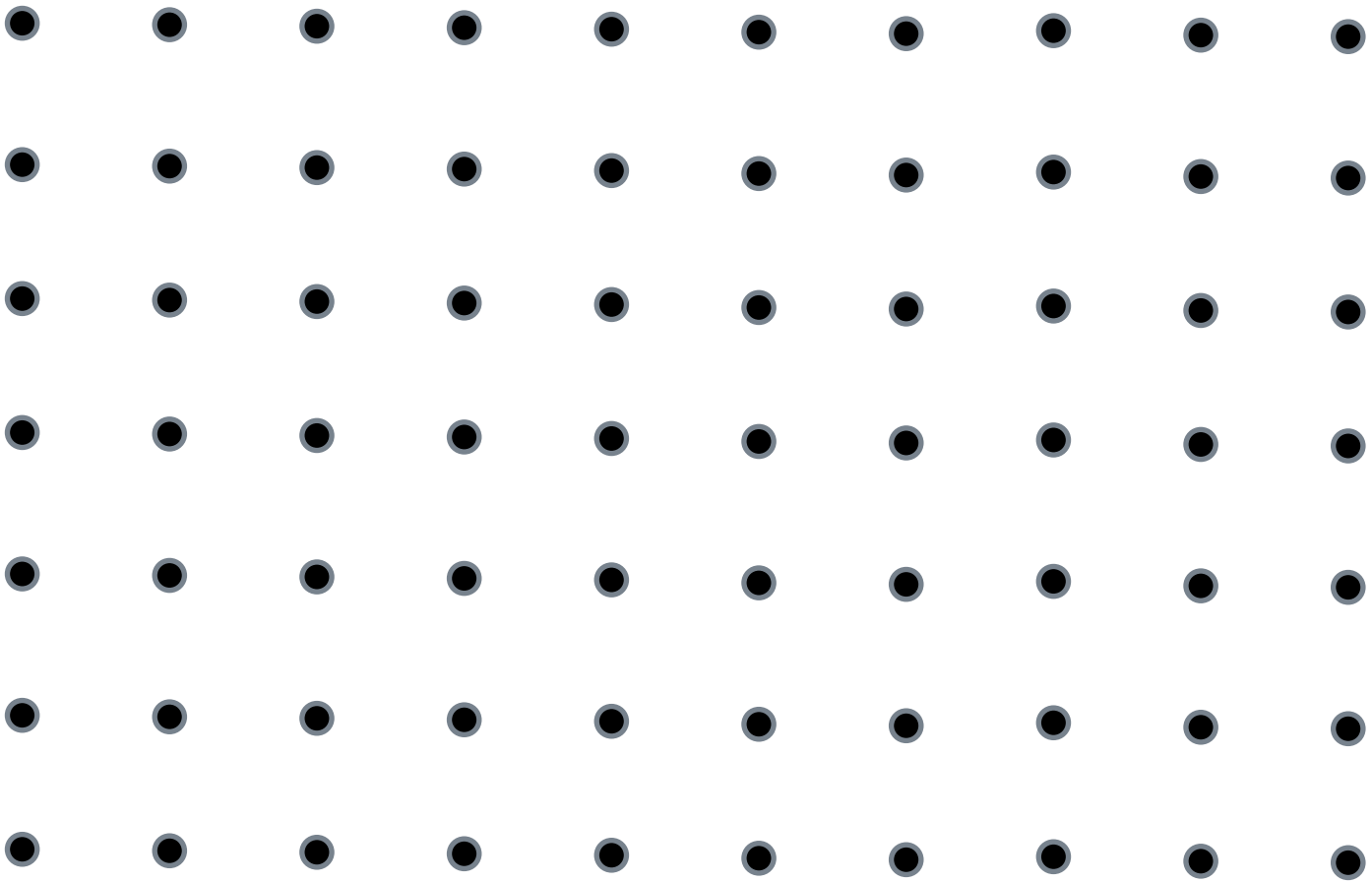
ラティスとは、「格子」のことである。

ラティスとは、空間上に規則的に格子状に配置された点の集合である。

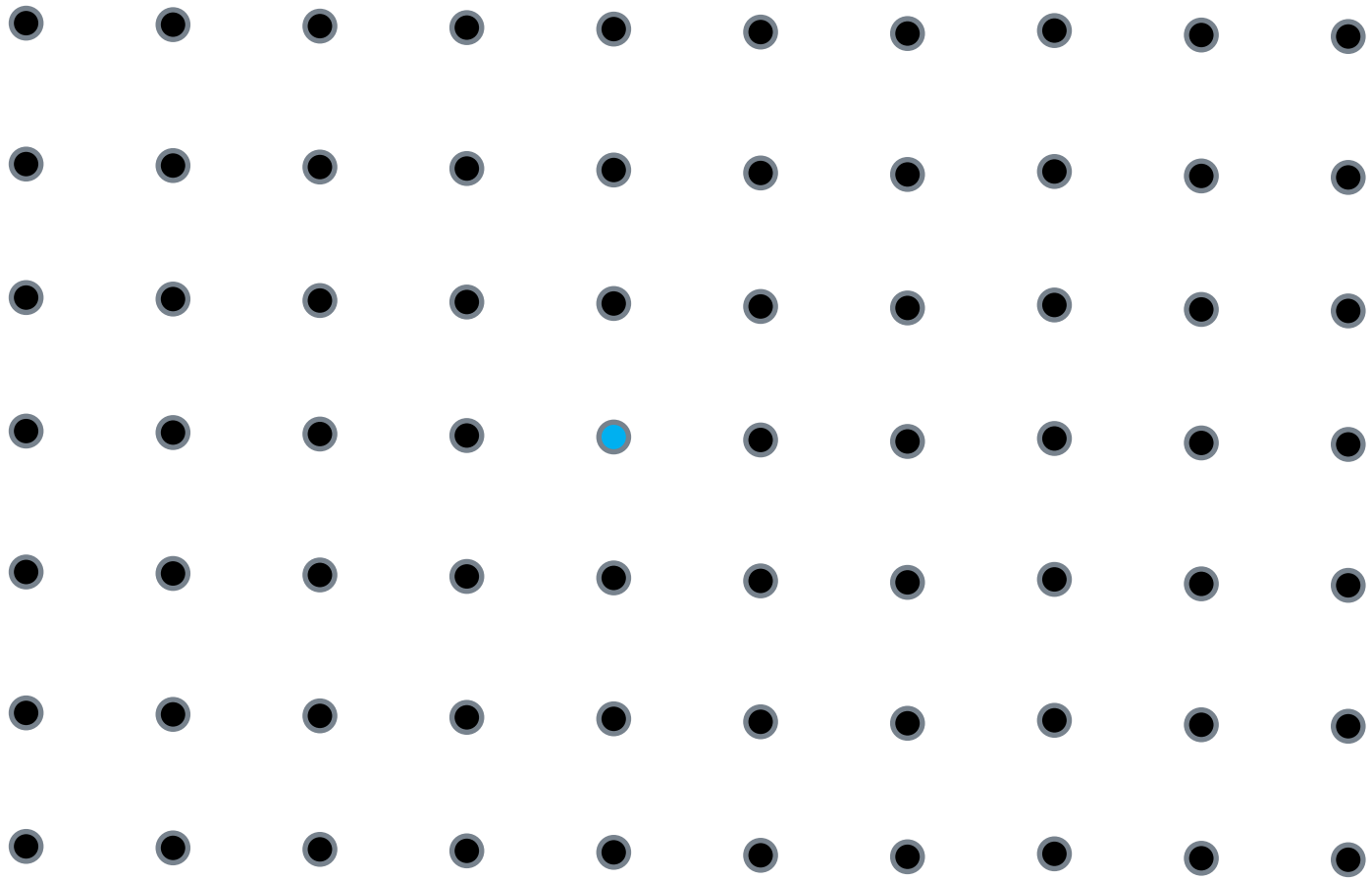
ここでは、こうした格子点の集合をどのように特徴づけるかを考える。

次の図の点の集合は、ラティスである。

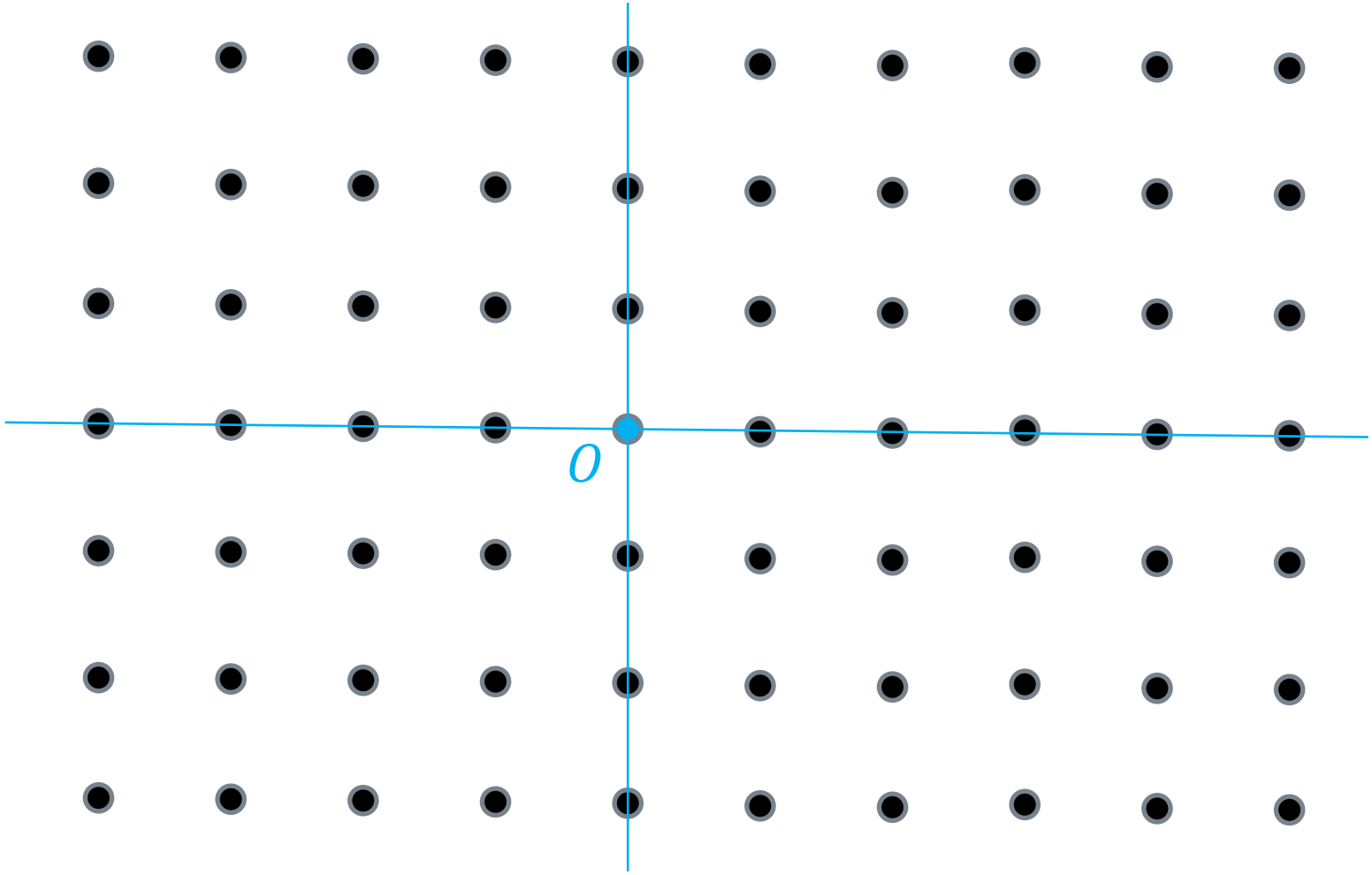
ラティスの例



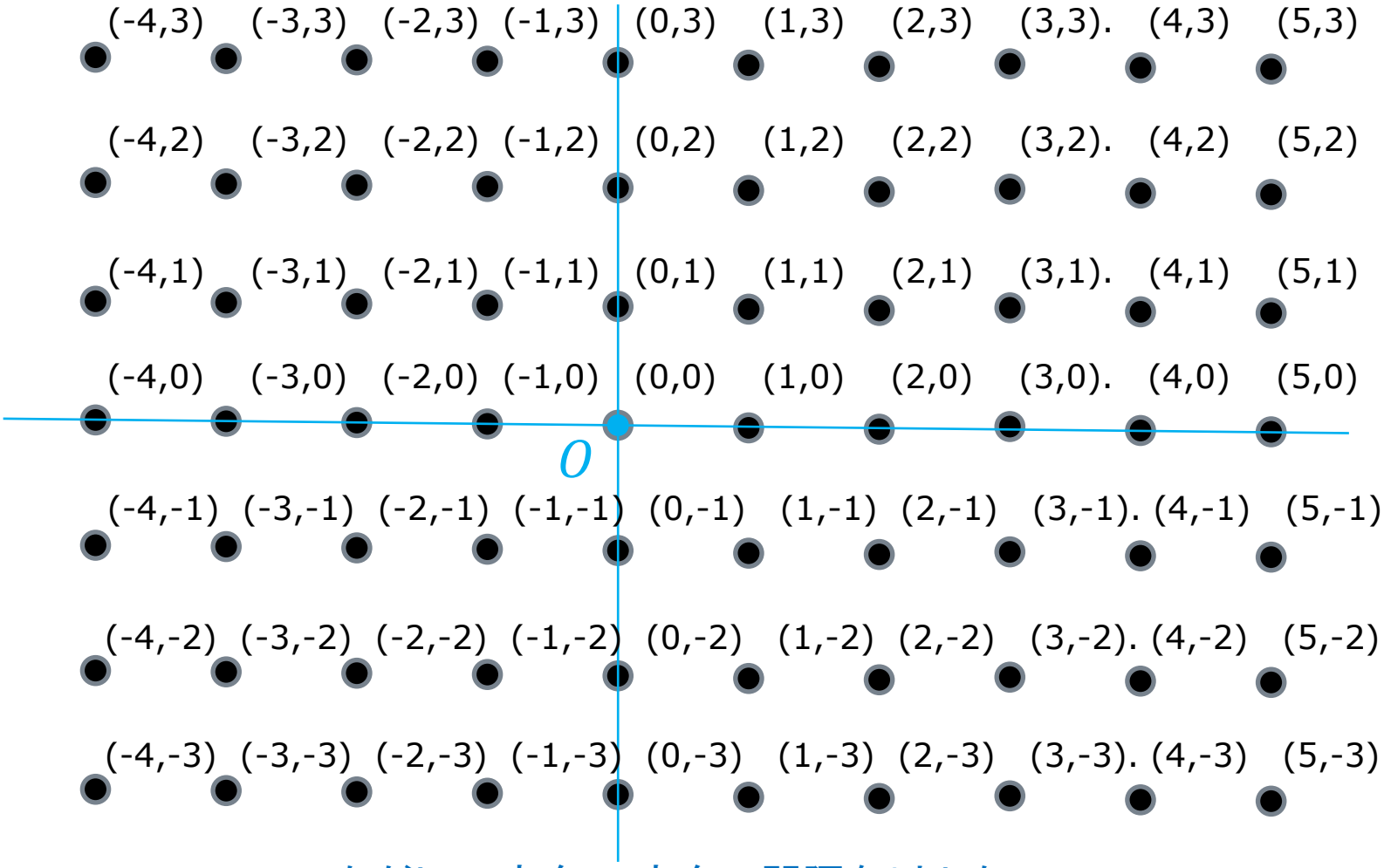
この点の集合の「規則正しさ」を考える



この点の集合の「規則正しさ」を考える
それぞれの点に座標を入れて見る



この点の集合の「規則正しさ」を考える
それぞれの点に座標を入れて見る



ただし、x方向、y方向の間隔を1とした

この点の集合の「規則正しさ」を考える
それぞれの点の座標は二つの整数の組である

(-4,3) (-3,3) (-2,3) (-1,3) (0,3) (1,3) (2,3) (3,3). (4,3) (5,3)

(-4,2) (-3,2) (-2,2) (-1,2) (0,2) (1,2) (2,2) (3,2). (4,2) (5,2)

(-4,1) (-3,1) (-2,1) (-1,1) (0,1) (1,1) (2,1) (3,1). (4,1) (5,1)

(-4,0) (-3,0) (-2,0) (-1,0) (0,0) (1,0) (2,0) (3,0). (4,0) (5,0)

(-4,-1) (-3,-1) (-2,-1) (-1,-1) (0,-1) (1,-1) (2,-1) (3,-1). (4,-1) (5,-1)

(-4,-2) (-3,-2) (-2,-2) (-1,-2) (0,-2) (1,-2) (2,-2) (3,-2). (4,-2) (5,-2)

(-4,-3) (-3,-3) (-2,-3) (-1,-3) (0,-3) (1,-3) (2,-3) (3,-3). (4,-3) (5,-3)

ただし、x方向、y方向の間隔を1とした

$\mathbb{Z}^2$ は、ラティスである

平面上の格子点の座標は、二つの整数の組である。

整数を $\mathbb{Z}$で表す。

$$\mathbb{Z} = \{ \ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4 \ \ldots \}$$

整数の二つの組を、$\mathbb{Z} \times \mathbb{Z} = \mathbb{Z}^2$ と表す。

$\mathbb{Z}^2$ は、ラティスである。

一般に、n次元の空間で、$\mathbb{Z}^n$ は、ラティスである。

ラティスとその「基底」

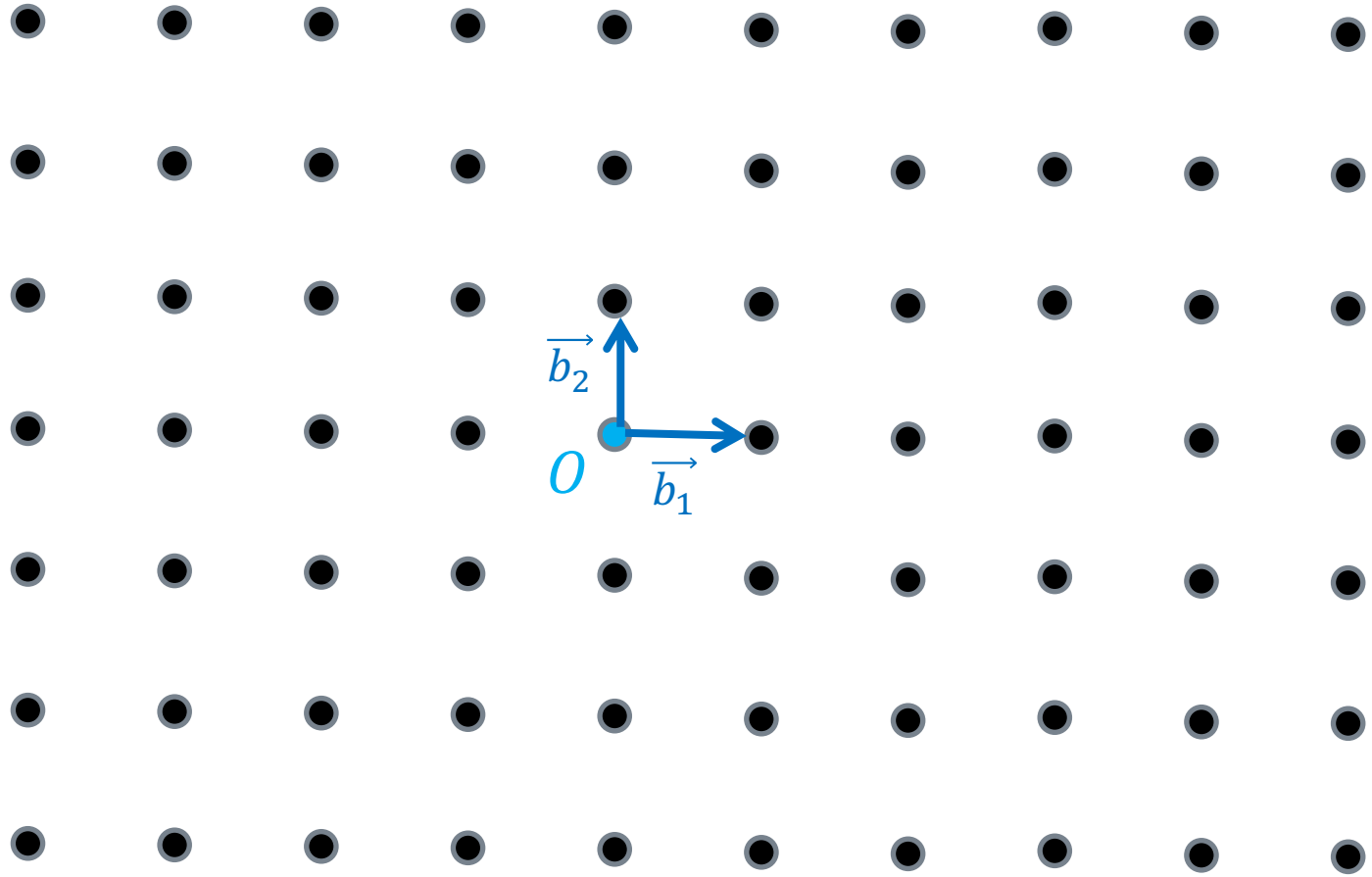
先に、二つの整数の組としてラティスを見てきたが、ここでは、他の捉え方もしてみよう。

それは、二つのベクトル $\overrightarrow{b_1}$, $\overrightarrow{b_2}$ の線型結合のベクトルとして、ラティスの点集合を捉えることである。

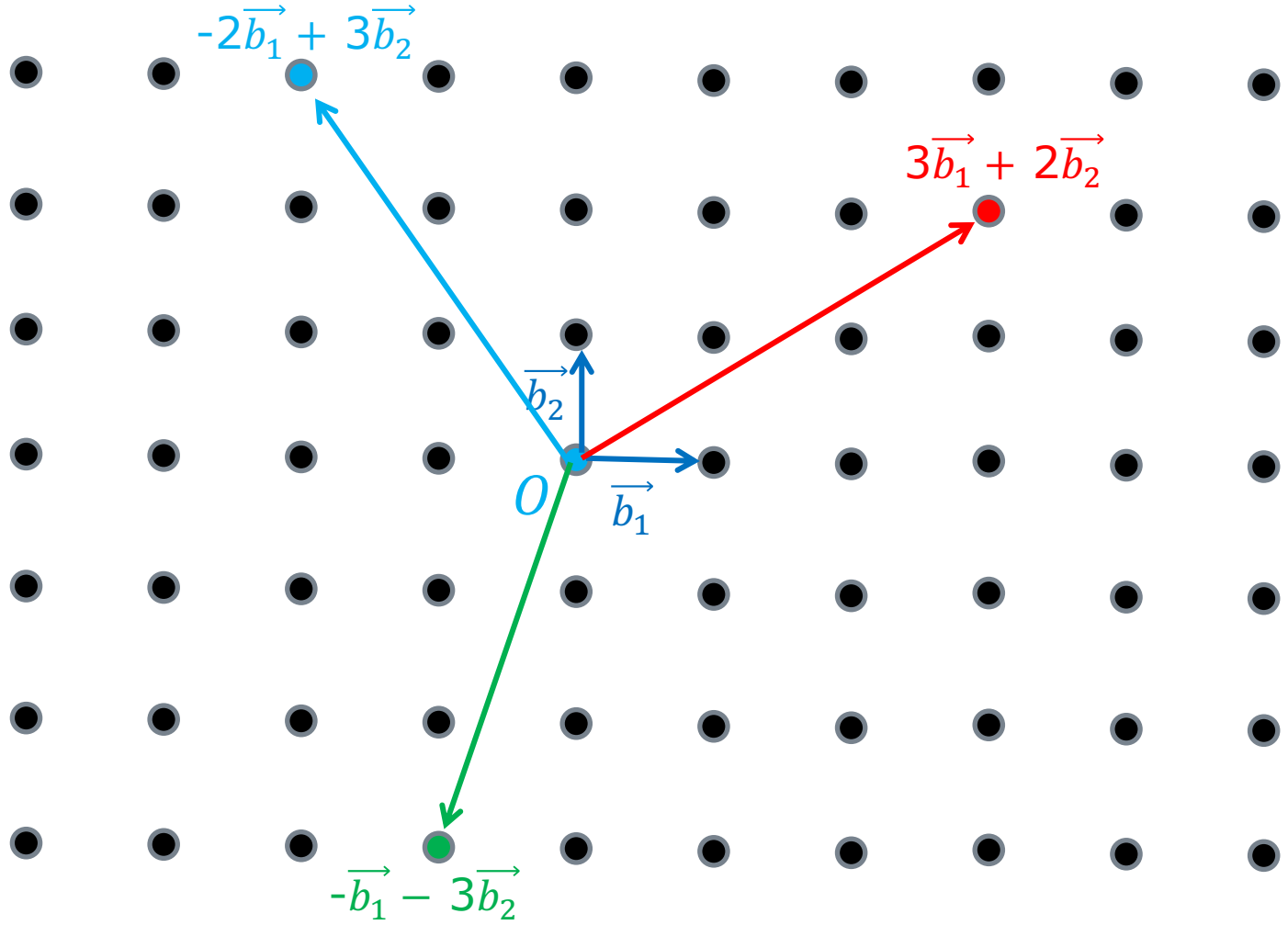
ラティス上の全ての点がそうした表現を持つ時、ベクトル $\overrightarrow{b_1}$, $\overrightarrow{b_2}$をそのラティスの「基底」という。

ベクトル $\overrightarrow{b_1}=(1,0)$, $\overrightarrow{b_2}=(0,1)$ は、先のラティスの基底である。

基底 $\overrightarrow{b_1}$=(1,0), $\overrightarrow{b_2}$=(0,1)

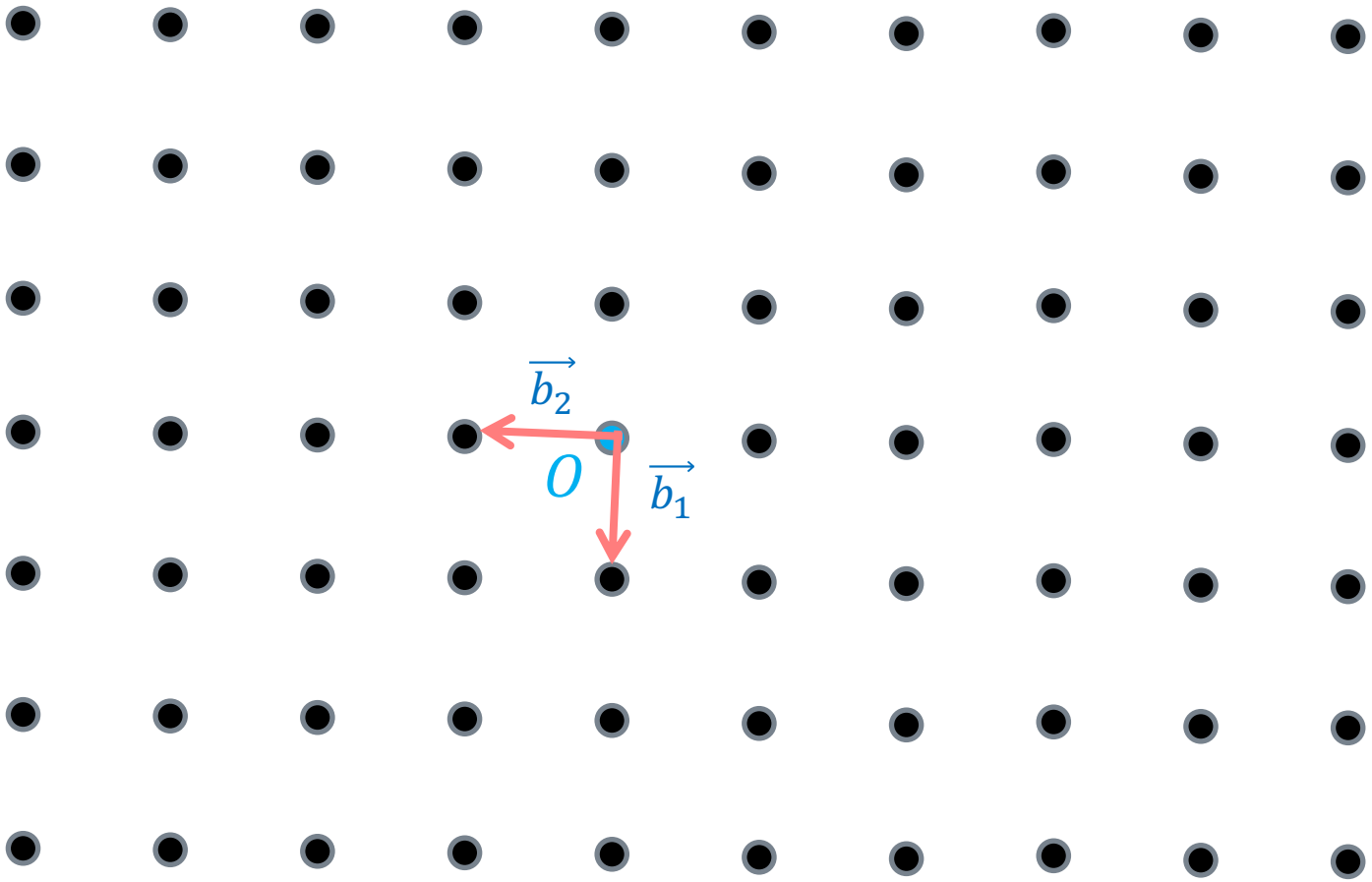


基底 $\overrightarrow{b_1}$=(1,0), $\overrightarrow{b_2}$=(0,1) の線型結合

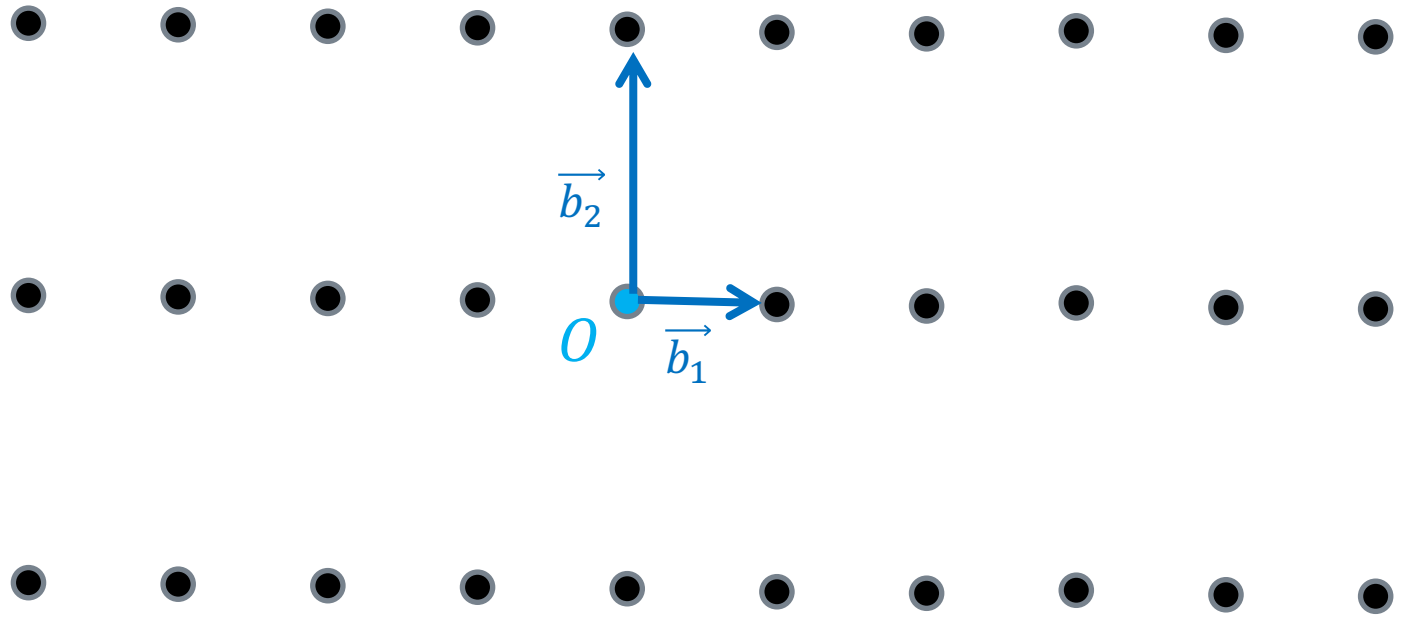


基底 は一つではない

$\overrightarrow{b_1}$=(0,-1), $\overrightarrow{b_2}$=(-1,0) も基底である

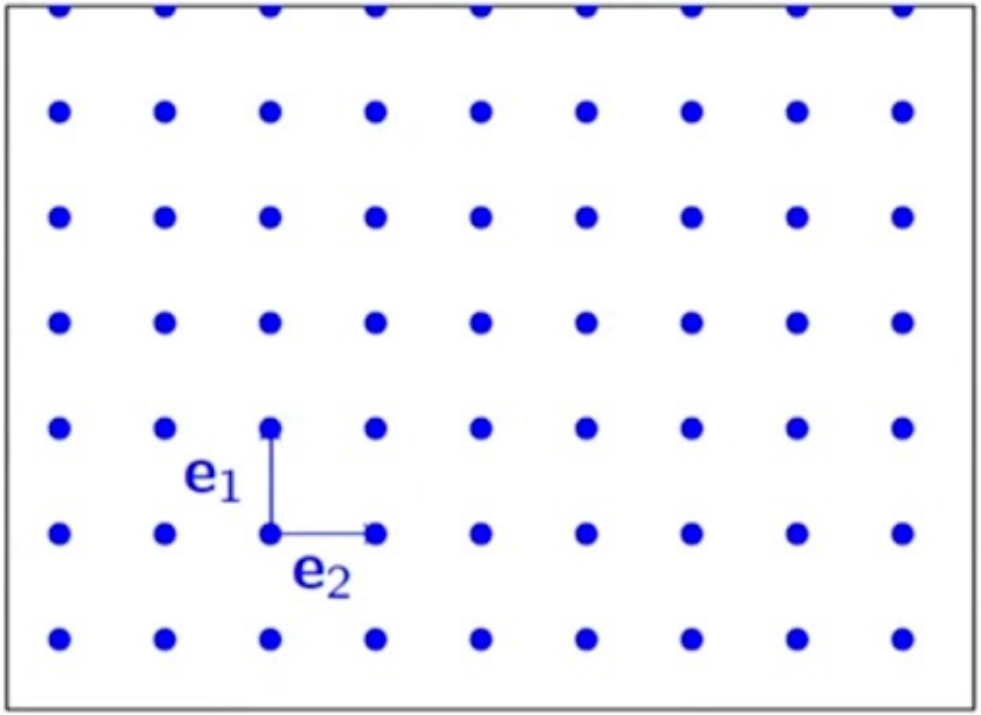


基底 $\overrightarrow{b_1}$=(1,0), $\overrightarrow{b_2}$=(0,2) から作られるラティス



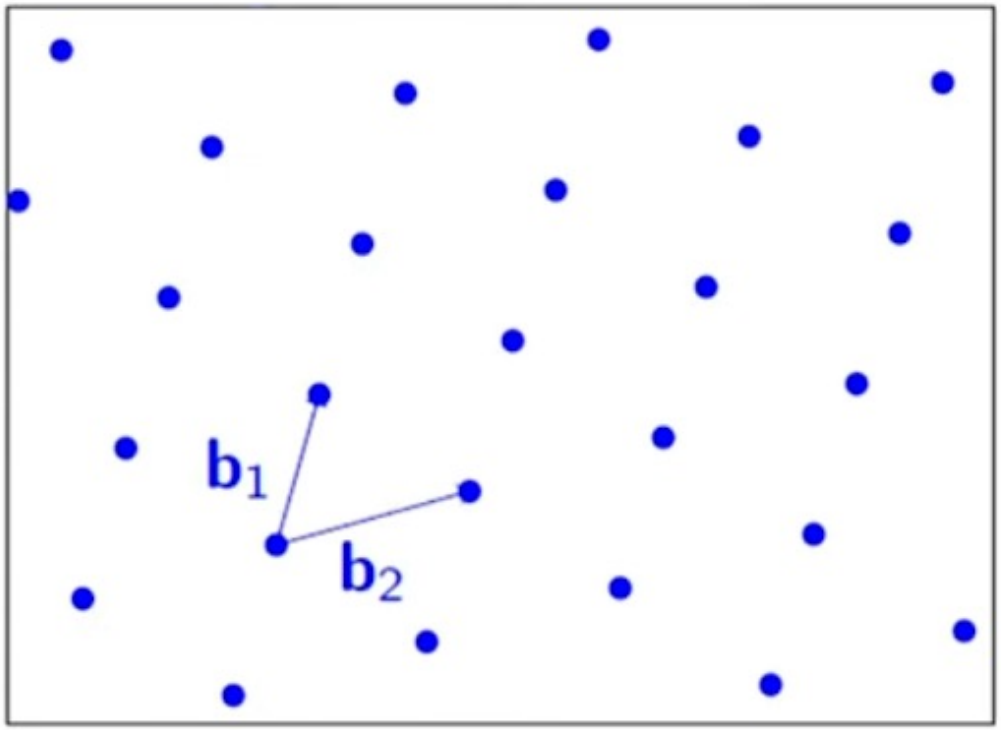
基底でラティスを定義する

前回見た単純なラティス



基底 e_1, e_2 で定義されるラティスLを$L = L(e_1, e_2)$ と表わそう。
正規直交基底 e_1, e_2 で定義されるラティスは、$\mathbb{Z}^2$ に等しい。
一般のn次元では、正規直交基底で定義されるラティスは、$\mathbb{Z}^n$になる 。

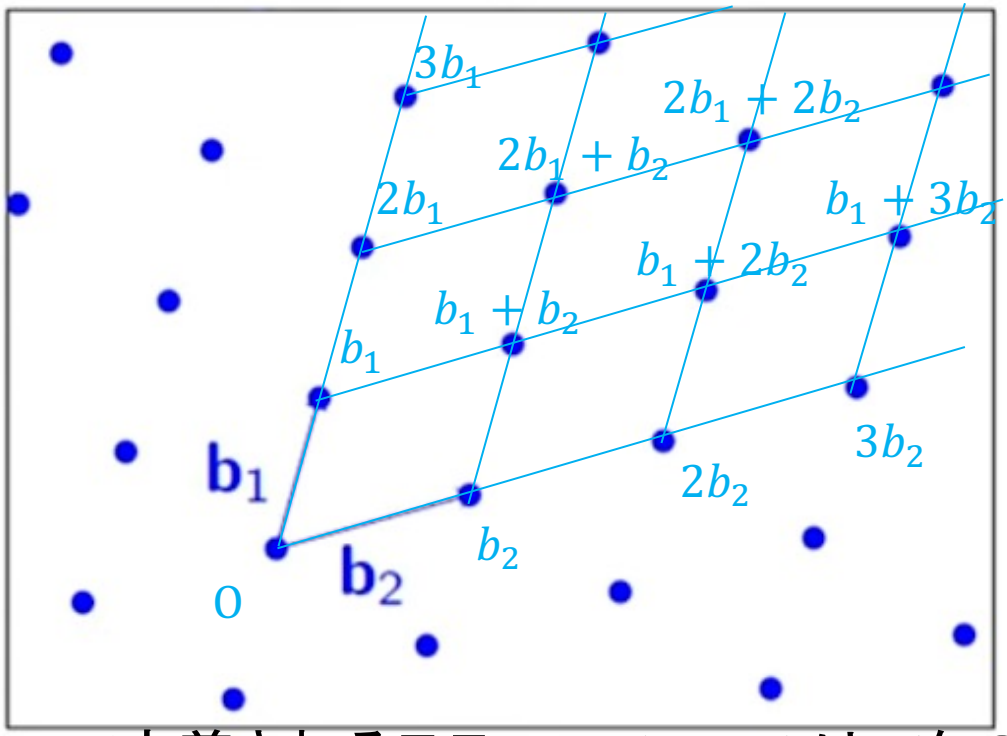
基底は、直交する単位ベクトルとは限らない



基底は、直交する単位ベクトルとは限らない。

上の図形は、基底 b_1, b_2で定義されるラティスL(b_1, b_2) である。

ラティスは、基底の整数倍の和で表わされる



この基底$b_1.b_2$で定義されるラティス$L(b_1.b_2)$は、次のような格子点から構成される。

$$o, b_2, 2b_2, 3b_2, \ldots, b_1, b_1 + b_2, b_1 + 2b_2, b_1 + 3b_2, \ldots$$

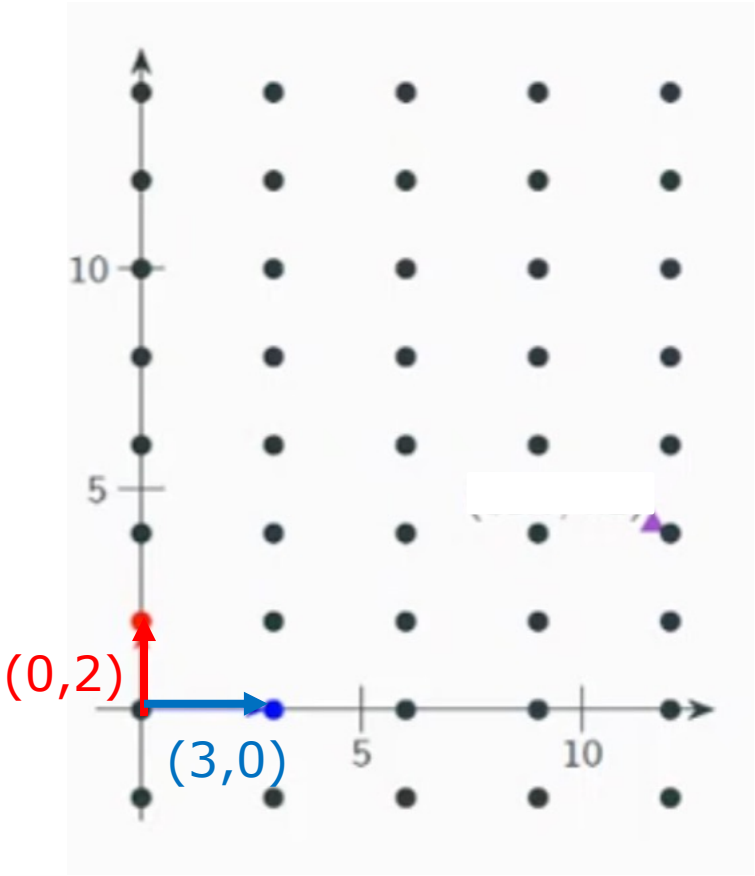
ラティスは、基底の整数倍の和で表わされる

ラティスと基底 – 一般形

n次元のラティスの基底を$b_1, b_2, \ldots, b_n$とする。
ラティスの各点は、n個の基底$b_1, b_2, \ldots, b_n$の整数倍の和で表される。

$$L(b_1, b_2, \ldots, b_n) = \sum_{i=1}^{n} b_i \mathbb{Z}$$

同一のラティスでも、複数の基底がある

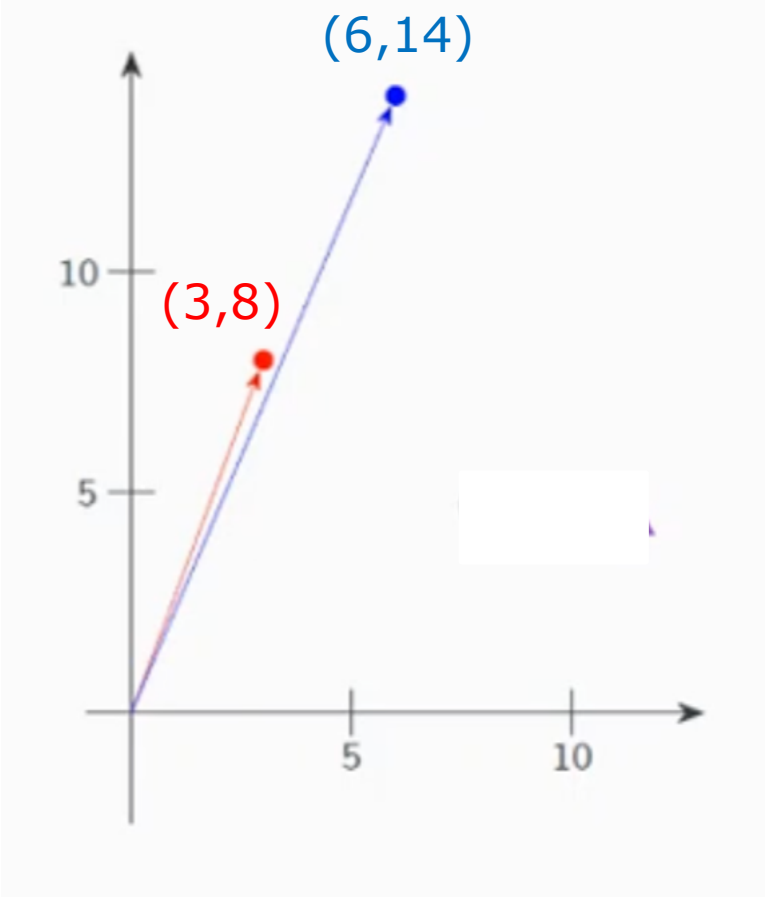


左のラティスには、
(3, 0) と(0,2) という基底があるのはすぐわかる。

ただ、同一のラティスでも、基底は複数存在する。

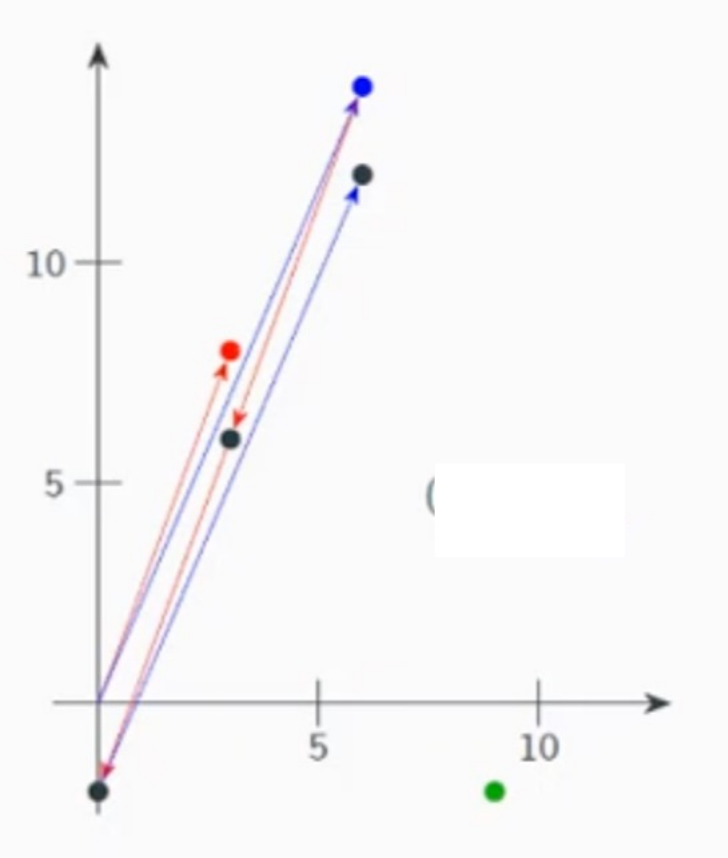
次のものは、このラティスの基底である。

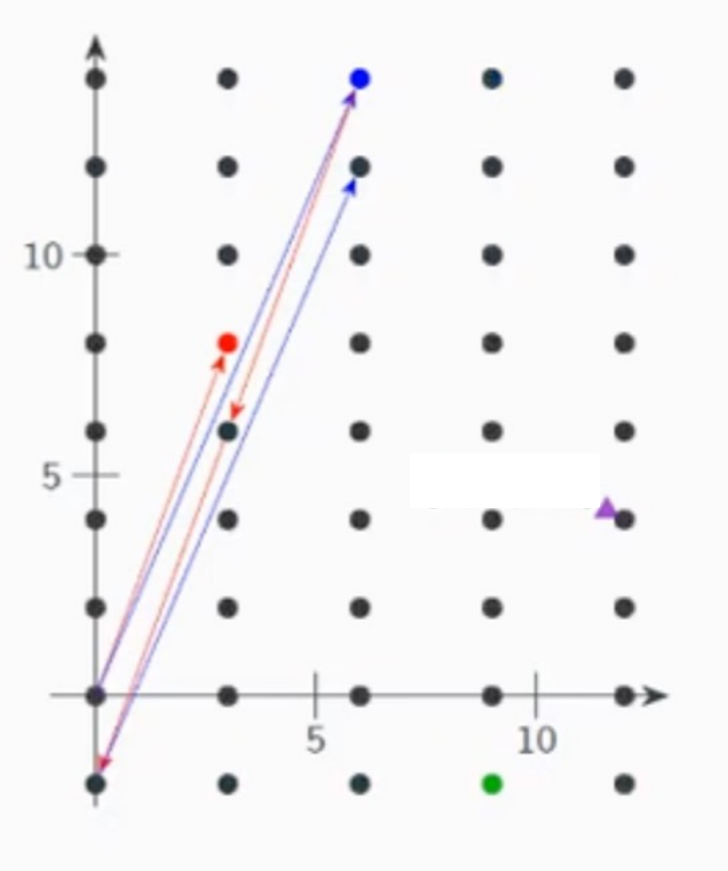
もう一つの基底



この基底 (3,8) , (6,14) は、次のように、先のラティスの各点をすべてカバーする。

先のラティスの各点をすべてカバーする。





ラティス問題

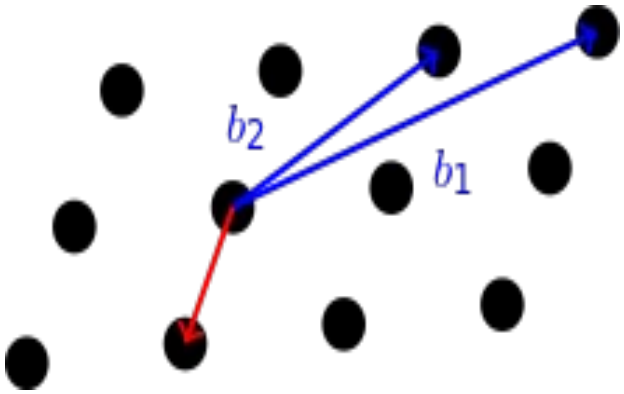
ラティス問題

ラティスに関して、いくつか基本的な問題がある。ここでは次の二つの問題を紹介する。

- Shortest vector problem (SVP)
- Closest vector problem (CVP)

これらの問題は、次元が高くなると一般には解くのが難しい。その難しさが、ラティス暗号の基礎になっている。

Shortest vector problem (SVP)

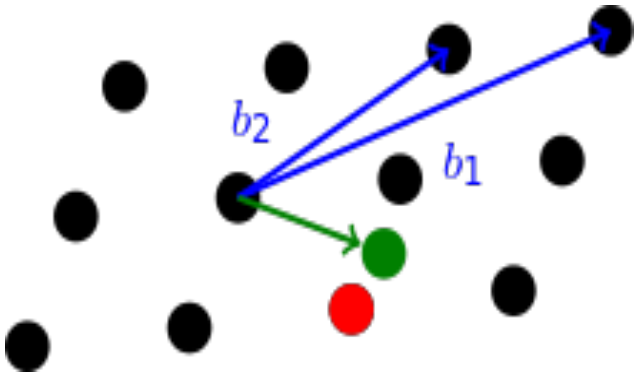


左の図のように基底b_1, b_2で張られるラティスがあるとする。

このラティス上の点で、互いに一番近い二点を求めよ。

（答は、赤い線で結ばれた二点である）

Closest vector problem (CVP)



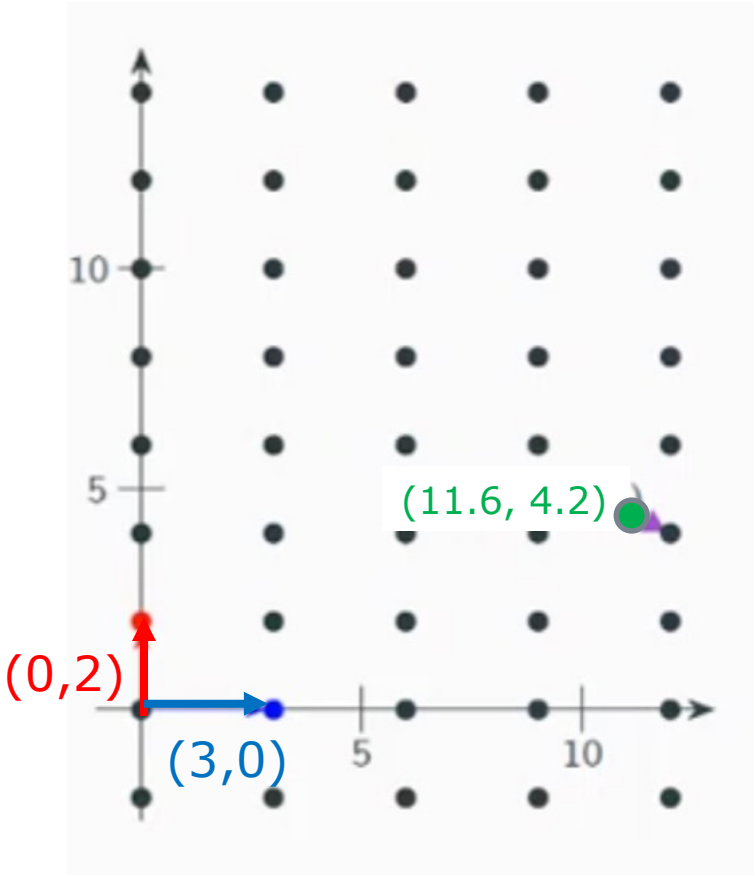
左の図のように基底b_1, b_2で張られるラティスがあるとする。

この平面上に、ラティスに属さない点を一つとる。（緑の点）

ラティス上の点で、この緑の点に一番近い点を求めよ。

（答は、赤い点である）

基底(3, 0) と(0,2)で張られるラティスで (11.6, 4.2) という点に一番近いラティスの点は？



$m(3,0) + n(0,2) = (11.6, 4.2)$

を解いてみる。

$3m = 11.6\,, 2n = 4.2$

$m = 3.87, n = 2.1$

m, n は整数だから、一番近い整数に丸めると、

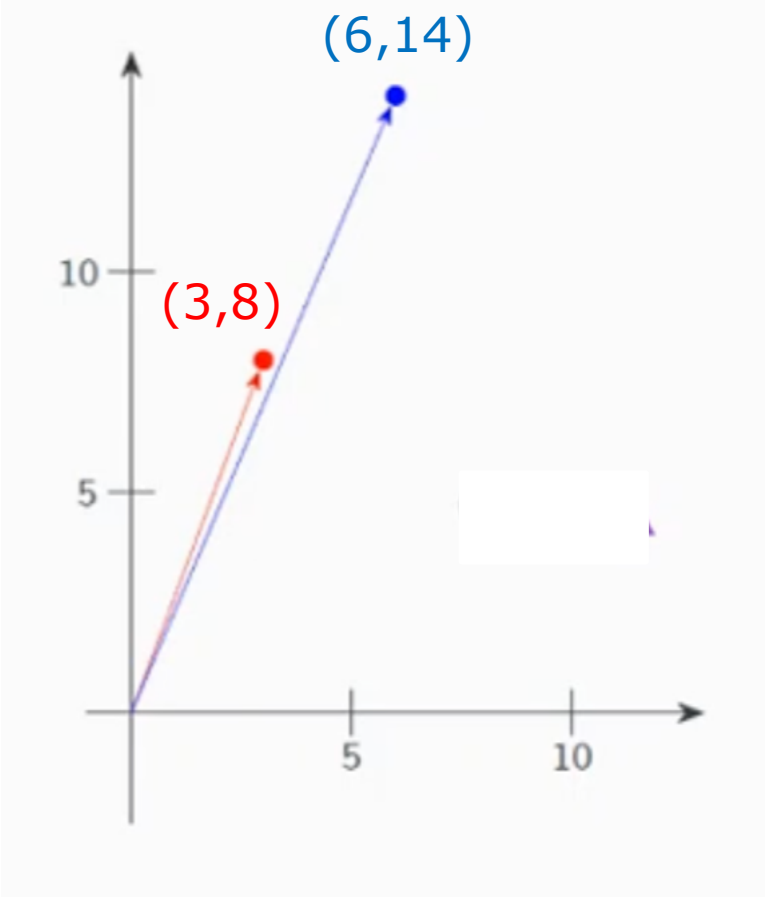
$m = 4, n = 2$。この時、

$4(3,0) + 2(0,2) = (12, 4)$

$\approx (11.6, 4.2)$

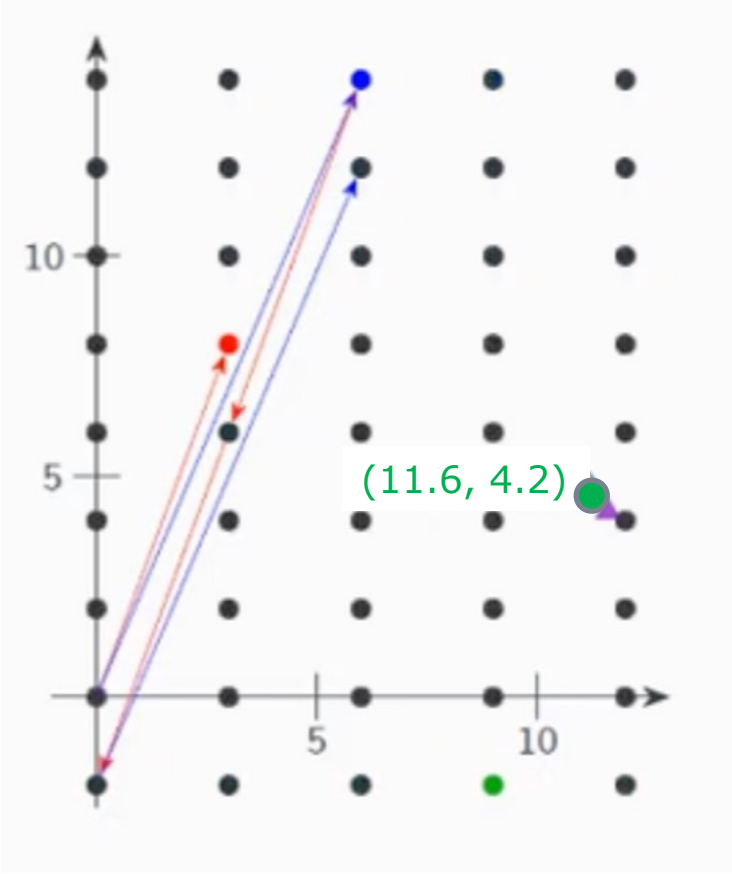
Babaiのアルゴリズム

もう一つの基底



この基底 (3,8) , (6,14) は、次のように、先のラティスの各点をすべてカバーする。

基底(3, 8) と(6,14)で張られるラティスで (11.6, 4.2) という点に一番近いラティスの点は？



$m(3,8) + n(6,14) = (11.6, 4.2)$

を解いてみる。

$$3m + 6n = 11.6$$
$$8m + 14n = 4.2$$

答えを整数に丸めると、

$m = -23, n = 13$。

この時、

$$-23(3,8) + 13(6,14)$$
$$= (9, -2)$$

さっきの点と違う。

$$-24(3,8) + 14(6,14)$$
$$= (12, 4)$$

で、先の結果と一致する。

良い基底と悪い基底

Babaiのアルゴリズムの整数近似解で Closest vector が正しく求まる基底を「良い基底」、そうでないなら「悪い基底」と、仮に呼ぼう。

どうやら、二つのベクトルの角度が大きい時に「良い基底」に、二つのベクトルの角度が小さい時に「悪い基底」になりそうである。

二つのベクトルが作る角度θ
（二次元の場合）

ベクトルを成分で表して$a = (a_1, a_2), b = (b_1, b_2)$とする。

ベクトルa, bの長さを$|a|, |b|$とすると

$$|a|^2 = a_1^2 + a_2^2,\ \ |b|^2 = b_1^2 + b_2^2$$

ベクトルa, bの内積$a \cdot b$は、ベクトルa, bが作る角度をθとすれば、

$$a \cdot b = |a||b|cos\theta = a_1 b_1 + a_2 b_2$$

$$cos\theta = \frac{a \cdot b}{|a||b|} = \frac{a_1 b_1 + a_2 b_2}{\sqrt{a_1^2 + a_2^2}\sqrt{\mathrm{b}_1^2 + \mathrm{b}_2^2}}$$

二つのベクトル(5,1),(-2,8)が作る角度θを求める

$$cos\theta = \frac{a \cdot b}{|a||b|} = \frac{5 \cdot (-2) + 1 \cdot 8}{\sqrt{5^2 + 1^2}\sqrt{(-2)^2 + 8^2}}$$

$$= \frac{-2}{\sqrt{26}\sqrt{68}}$$

$$\approx -0.05$$

$cos\theta$ は、ゼロに近いので、この基底(5,1),(-2,8)については、Babaiのアルゴリズムは機能しそうである。

先に失敗した基底(3, 8) と(6,14)が作る角度θを求める

$$cos\theta = \frac{a \cdot b}{|a||b|} = \frac{3 \cdot 6 + 8 \cdot 14}{\sqrt{3^2 + 8^2}\sqrt{6^2 + 14^2}}$$

$$= \frac{18 + 112}{\sqrt{9 + 64}\sqrt{36 + 196}} = \frac{130}{\sqrt{73}\sqrt{232}} \approx \frac{130}{8.54 \cdot 15.2} = \frac{130}{129.8}$$

$$\approx 1$$

$cos\theta$ は、ほぼ1である。

この基底(3, 8) ,(6,14)については、Babaiのアルゴリズムは機能しそうもない。

同じラティスを生成する複数の基底

同じラティスを生成する「等価」な基底

基底 $b_1, b_2, \cdots, b_n$ から作られるラティスを、次のように表す。

$$L = L(b_1, b_2, \cdots, b_n)$$

ラティスLは、複数の基底を持つ。
もう一つの基底を $b'_1, b'_2, \cdots, b'_n$ とすれば、

$$L = L(b_1, b_2, \cdots, b_n) = L(b'_1, b'_2, \cdots, b'_n)$$

である。

この時、基底$(b_1, b_2, \cdots, b_n)$と基底$(b'_1, b'_2, \cdots, b'_n)$は、$L$について「等価」であると呼んで、次のように表そう。

$$(b_1, b_2, \cdots, b_n) \Leftrightarrow (b'_1, b'_2, \cdots, b'_n)$$

等価な基底の間には、どんな関係があるかを考えてみよう。

$(b_1, b_2, \cdots, b_n)$がLの基底であること

ベクトルのn個の並び $(b_1, b_2, \cdots, b_n)$がLの基底であるということは、次のことを意味する。

Lの全ての点は、n個の基底ベクトル$b_1, b_2, \cdots, b_n$と n個の整数 $n_1, n_2, \cdots, n_n$ で、次のように表される。

$$n_1 b_1 + n_2 b_2 + \cdots + n_n b_n$$

$n_1 b_1 + n_2 b_2 + \cdots + n_n b_n \in L(b_1, b_2, \cdots, b_n)$である
整数の組 $(n_1, n_2, \cdots, n_n) \in \mathbb{Z}^n$ が存在する。

基底の順番を入れ替えたものも基底である

$L(b_1, b_2, \cdots, b_i \cdots, b_j, \cdots, b_n)$ は

$n_1 b_1 + n_2 b_2 + \cdots + n_i b_i + \cdots + n_j b_j + \cdots + n_n b_n \in L$ である

整数の組 $(n_1, n_2, \cdots, n_n) \in \mathbb{Z}^n$ が存在するということ。

$$n_1 b_1 + n_2 b_2 + \cdots + n_i b_i + \cdots + n_j b_j + \cdots + n_n b_n$$
$$= n_1 b_1 + n_2 b_2 + \cdots + n_j b_j + \cdots + n_i b_i + \cdots + n_n b_n \in L$$

これから、

$(b_1, b_2, \cdots, b_i \cdots, b_j, \cdots, b_n) \Leftrightarrow (b_1, b_2, \cdots, b_j, \cdots, b_i, \cdots, b_n)$

が言える。

一つの基底の符号を変えたものも基底である

$L(b_1, b_2, \cdots, -b_i \cdots, , b_n)$ は
$n_1 b_1 + n_2 b_2 + \cdots + n_i(-b_i) + \cdots + b_n \in L$ である
整数の組 $(n_1, n_2, \cdots, n_n) \in \mathbb{Z}^n$ が存在するということ。

$$n_1 b_1 + n_2 b_2 + \cdots + n_i(-b_i) + \cdots + n_j b_n =$$
$$n_1 b_1 + n_2 b_2 + \cdots + (-n_i) b_i + \cdots + n_n b_n \in L$$

これから、

$$(b_1, b_2, \cdots, b_i, \cdots, b_n) \Leftrightarrow (b_1, b_2, , \ldots, -b_i, \cdots, b_n)$$

がいえる。

基底の一つを，その基底に別の基底の整数倍を加えたものに置き換えたものも基底である

$L(b_1, b_2, \cdots, b_i \cdots, , b_n)$ は

$n_1 b_1 + n_2 b_2 + \cdots + n_i b_i + \cdots + n_j b_j + \cdots + n_n b_n \in L$ である

整数の組 $(n_1, n_2, \cdots, n_n) \in \mathbb{Z}^n$ が存在するということ。

$$n_1 b_1 + n_2 b_2 + \cdots + n_i (b_i + k b_j) + \cdots + n_j b_j + \cdots + n_n b_n =$$
$$n_1 b_1 + n_2 b_2 + \cdots + n_i b_i + \cdots + (n_i k + 1) b_j + \cdots + n_j b_n \in L$$

これから、kを整数とすれば、

$$(b_1, b_2, \cdots, b_i, \cdots, b_n) \Leftrightarrow (b_1, b_2, , \ldots, b_i + k b_j, \cdots, b_n)$$

がいえる。

基底を行列で表現する

基底を構成するベクトルを列ベクトルで表し、それを並べて行列を作る。ラティスの基底は、$n \times n$の行列Bで表現される。
$L = L(b_1, b_2, \cdots, b_n) = L(B)$と表そう。

この時、先の等価な基底についてのルールは、基底の等価性を保ったままでの行列の操作として、次のようにまとめられる。

1. 行列Bの列$\mathrm{v_i}$を$\mathrm{v_j}$と交換したものをB′とすると、$\mathrm{L(B) = L(B')}$
2. 行列Bの一列$\mathrm{v_i}$に-1に掛けたものをB′とすると、$\mathrm{L(B) = L(B')}$
3. 行列Bの一列$\mathrm{v_i}$を$\mathrm{v_i + kv_j}$にかえたものをB′とすると、$\mathrm{L(B) = L(B')}$

行列の操作で等価な基底を見つける

例えば、基底(3, 0) と(0,2)、基底(3, 8) と(6,14)は同じラティスを作る。

基底を列ベクトルで表すと

$$\mathrm{L}\left(\begin{pmatrix}3\\0\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}\right) = \mathrm{L}\left(\begin{pmatrix}3\\8\end{pmatrix}, \begin{pmatrix}6\\14\end{pmatrix}\right)$$

行列で基底を表し、

$$L\left(\begin{bmatrix}3 & 0\\0 & 2\end{bmatrix}\right) = L\left(\begin{bmatrix}3 & 6\\8 & 14\end{bmatrix}\right)$$

となることを、行列の操作で確かめてみよう。

行列の操作で等価な基底を見つける

$$\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 0 \\ 8 & 2 \end{bmatrix}$$

1列＋4×2列

$$\begin{pmatrix} 3 \\ 8 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} + 4\begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\begin{bmatrix} 3 & 0 \\ 8 & 2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 0 \\ 8 & -2 \end{bmatrix}$$

−1×2列

$$\begin{pmatrix} 0 \\ -2 \end{pmatrix} = -\begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\begin{bmatrix} 3 & 0 \\ 8 & -2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 6 \\ 8 & 14 \end{bmatrix}$$

2×1列＋2列

$$\begin{pmatrix} 6 \\ 14 \end{pmatrix} = 2\begin{pmatrix} 3 \\ 8 \end{pmatrix} + \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

$$\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 0 \\ 8 & 2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 0 \\ 8 & -2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 6 \\ 8 & 14 \end{bmatrix}$$

ラティスの基底の変換

基底を行列で表現する

基底を構成するベクトルを列ベクトルで表し、それを並べて行列を作る。ラティスの基底は、$n \times n$の行列Bで表現される。
$L = L(b_1, b_2, \cdots, b_n) = L(B)$と表そう。

この時、先の等価な基底についてのルールは、基底の等価性を保ったままでの行列の操作として、次のようにまとめられる。

1. 行列Bの列$\mathrm{v_i}$を$\mathrm{v_j}$と交換したものをB′とすると、$\mathrm{L(B) = L(B')}$
2. 行列Bの一列$\mathrm{v_i}$に-1に掛けたものをB′とすると、$\mathrm{L(B) = L(B')}$
3. 行列Bの一列$\mathrm{v_i}$を$\mathrm{v_i + kv_j}$にかえたものをB′とすると、$\mathrm{L(B) = L(B')}$

行列の操作で等価な基底を見つける

$$\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 0 \\ 8 & 2 \end{bmatrix}$$

1列＋4×2列

$$\begin{pmatrix} 3 \\ 8 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} + 4\begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\begin{bmatrix} 3 & 0 \\ 8 & 2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 0 \\ 8 & -2 \end{bmatrix}$$

−1×2列

$$\begin{pmatrix} 0 \\ -2 \end{pmatrix} = -\begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\begin{bmatrix} 3 & 0 \\ 8 & -2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 6 \\ 8 & 14 \end{bmatrix}$$

2×1列＋2列

$$\begin{pmatrix} 6 \\ 14 \end{pmatrix} = 2\begin{pmatrix} 3 \\ 8 \end{pmatrix} + \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

$$\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 0 \\ 8 & 2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 0 \\ 8 & -2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 6 \\ 8 & 14 \end{bmatrix}$$

行列の操作を、行列の積で表現する

同じラティスを張る等価な基底の例

$$\begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix} \Leftrightarrow \begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix} \Leftrightarrow \begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix} \Leftrightarrow \begin{bmatrix}3 & 6\\ 8 & 14\end{bmatrix}$$

$$\begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix}$$

$$\begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix} = \begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\ 4 & 1\end{bmatrix} \qquad \det\begin{vmatrix}1 & 0\\ 4 & 1\end{vmatrix} = 1$$

$$\begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix} = \begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \qquad \det\begin{vmatrix}1 & 0\\ 0 & -1\end{vmatrix} = -1$$

$$\begin{bmatrix}3 & 6\\ 8 & 14\end{bmatrix} = \begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix}\begin{bmatrix}1 & 2\\ -4 & -7\end{bmatrix} \qquad \det\begin{vmatrix}1 & 2\\ -4 & -7\end{vmatrix} = 1$$

同じラティスを張る等価な基底の例

$$\begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix} \Leftrightarrow \begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix} \Leftrightarrow \begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix} \Leftrightarrow \begin{bmatrix}3 & 6\\ 8 & 14\end{bmatrix}$$

$$\begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix} = \begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\ -4 & 1\end{bmatrix} \qquad \det\begin{vmatrix}1 & 0\\ -4 & 1\end{vmatrix} = 1$$

$$\begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix} = \begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \qquad \det\begin{vmatrix}1 & 0\\ 0 & -1\end{vmatrix} = -1$$

$$\begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix} = \begin{bmatrix}3 & 6\\ 8 & 14\end{bmatrix}\begin{bmatrix}1 & -2\\ 0 & 1\end{bmatrix} \qquad \det\begin{vmatrix}1 & -2\\ 0 & 1\end{vmatrix} = 1$$

$$\begin{bmatrix}3 & 6\\ 8 & 14\end{bmatrix}$$

同じラティスを張る等価な基底の例

$$\begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix} \Leftrightarrow \begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix} \Leftrightarrow \begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix} \Leftrightarrow \begin{bmatrix}3 & 6\\ 8 & 14\end{bmatrix}$$

$$\begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix}$$

$$\begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix} = \begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\ -4 & 1\end{bmatrix}$$

$$\begin{bmatrix}1 & 0\\ -4 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 4 & 1\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix} = \begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\ 4 & 1\end{bmatrix}$$

$$\begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix} = \begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

$$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix} = \begin{bmatrix}3 & 0\\ 8 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

$$\begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix} = \begin{bmatrix}3 & 6\\ 8 & 14\end{bmatrix}\begin{bmatrix}1 & -2\\ 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}1 & -2\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 2\\ -4 & -7\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}3 & 6\\ 8 & 14\end{bmatrix} = \begin{bmatrix}3 & 0\\ 8 & -2\end{bmatrix}\begin{bmatrix}1 & 2\\ -4 & -7\end{bmatrix}$$

$$\begin{bmatrix}3 & 6\\ 8 & 14\end{bmatrix}$$

Unimodular 行列

整数からなる $n \times n$ の行列 U で、
同じく整数からなる $n \times n$ の行列 V で、
次の式を満たす V が存在する時、U をユニモジュラー という

$$UV = VU = I$$

この時、次が成り立つ。

- もし、U がユニモジュラーなら U^{-1} もユニモジュラーである
- もし、U と V がユニモジュラーなら UV もユニモジュラーである
- U がユニモジュラーになるのは、$\det(U) = \pm 1$ の場合に限る

ラティスの基底の変換とユニモジュラー行列

Bがラティスの基底なら、Bに右からユニモジュラー行列Uをかけたも基底になる。

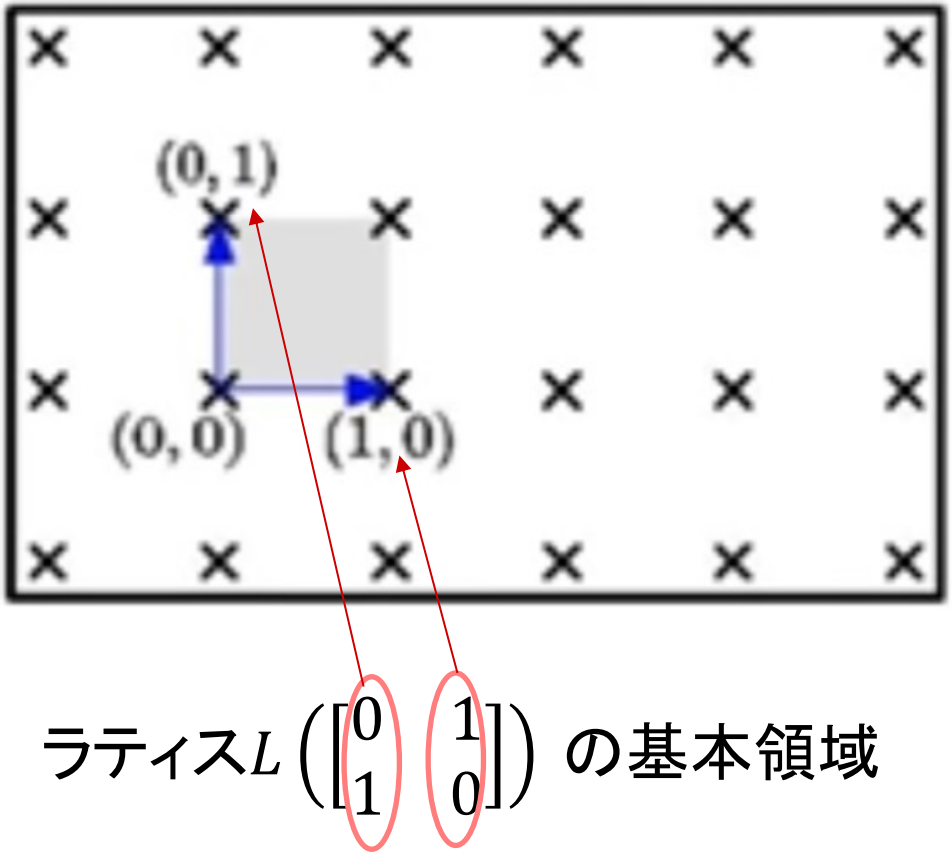
BとBUは、同じラティスLを張る。

ユニモジュラー行列Uが与えられた時、

$$L(B) = L(BU)$$

ラティスの「基本領域」

ラティス $L(B)$ の基本領域



ラティス$L\left(\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\right)$ の基本領域

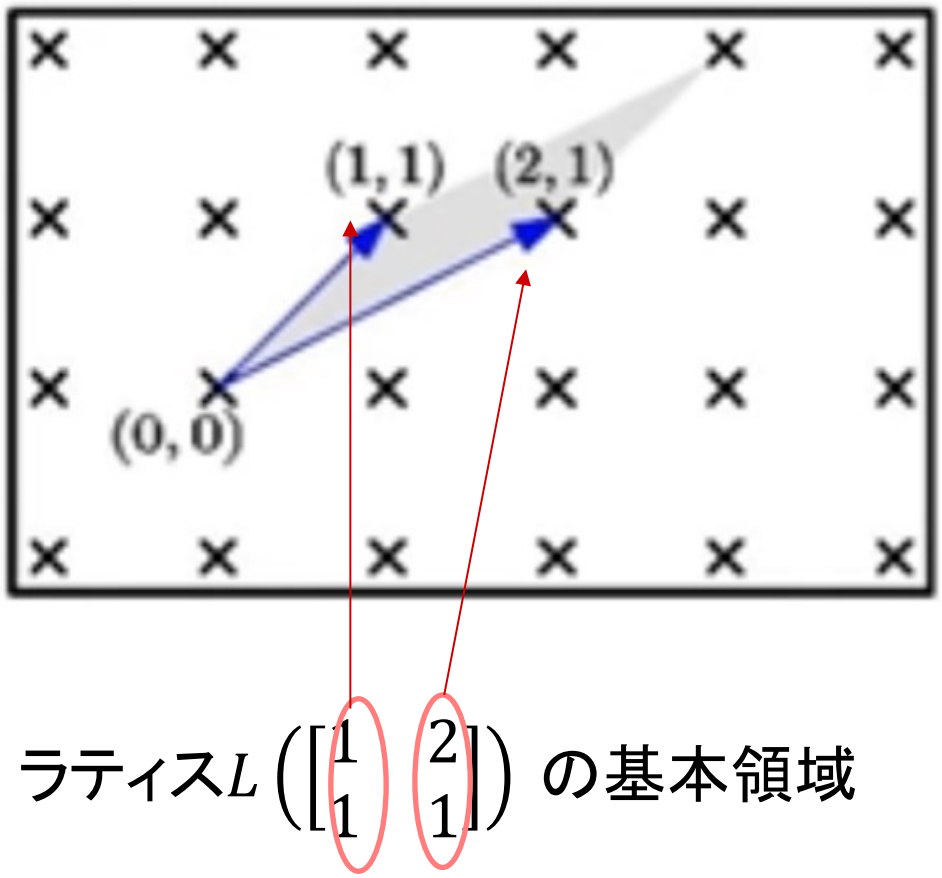
$B = (b_1, b_2, \ldots, b_n)$ を基底として、ラティス$L(B)$が張る平面上の、次の条件を満たす領域を、ラティス$L(B)$の「基本領域」という。

$a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$

で、$a_i \in [0,1)$

$a_i = 0$ は、この領域に含まれるが
$a_i = 1$は、この領域に含まれない

ラティス $L(B)$ の基本領域



ラティス$L\left(\begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}\right)$ の基本領域

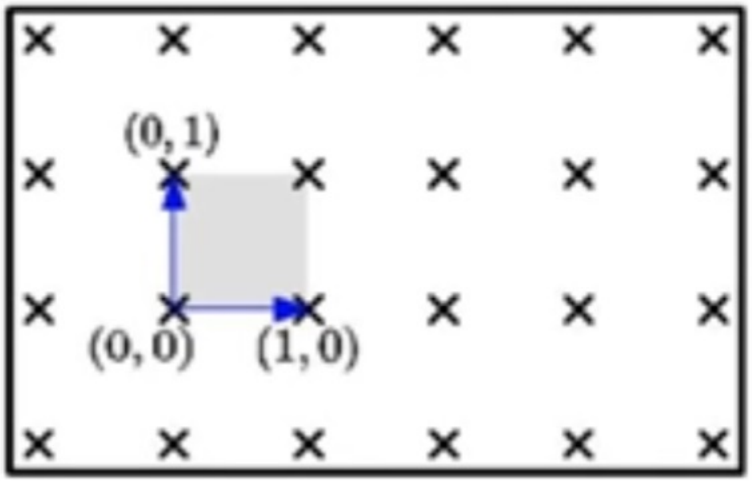
$B = (b_1, b_2, \ldots, b_n)$ を基底として、ラティス$L(B)$が張る平面上の、次の条件を満たす領域を、ラティス$L(B)$の「基本領域」という。

$a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$

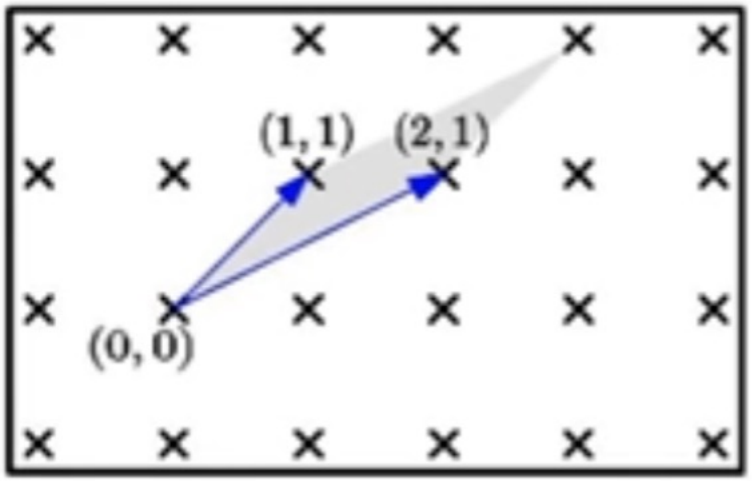
で、$a_i \in [0,1)$

$a_i = 0$ は、この領域に含まれるが
$a_i = 1$は、この領域に含まれない

同じラティスを張る基底



ラティス$L\left(\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\right)$ の基本領域



ラティス$L\left(\begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}\right)$ の基本領域

$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} = \begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}\begin{bmatrix}2 & -1\\ -1 & 1\end{bmatrix}$ $\det\left(\begin{bmatrix}2 & -1\\ -1 & 1\end{bmatrix}\right)=1$ でユニモジュラーである

基底 $\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$ と基底$\begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}$は、同じラティスを張る。 $L\left(\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\right)=L\left(\begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}\right)$

ラティスの行列式

ラティス$L(B)$の行列式を$\det(L)$で表し、$\det(B)$で定義する。

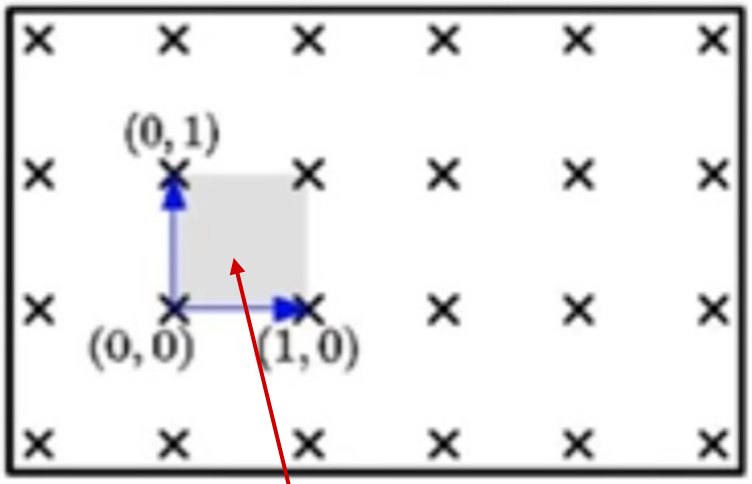
ラティスLの異なる基底をB_1, B_2とする。$L(B_1) = L(B_2)$。
この時、$|\det(U)| = 1$であるユニモジュラー行列Uが存在して、
$B_1 = B_2 U$と表すことができる。

$$|\det(B_1)| = |\det(B_2 U)| = |\det(B_2)\det(U)| = |\det(B_2)|$$

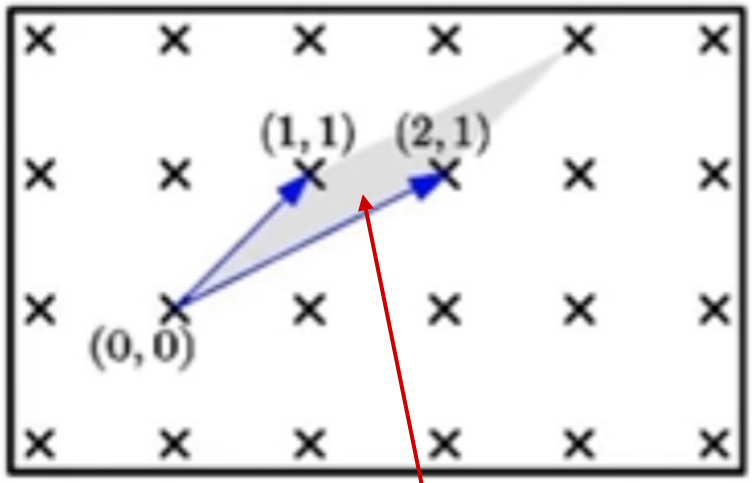
ラティスLの行列式$\det(L)$の値は、基底B_1, B_2の取り方によらず一定である。

$\det(B)$は、ベクトルBが張る図形の面積（体積）なので、
基底によらず、その基底が張る
ラティスLの基本領域の面積（体積）は全て等しい。

同じラティスを張る基底の基本領域



ラティス$L\left(\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\right)$の基本領域



ラティス$L\left(\begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}\right)$の基本領域

$$\det\left(\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\right) = \det\left(\begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}\begin{bmatrix}2 & -1\\ -1 & 1\end{bmatrix}\right) = \det\left(\begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}\right)$$

二つの基本領域の面積は等しい

$\mathbb{Z}$のラティスと$\mathbb{Z}_q$のラティス

整数$\mathbb{Z}$のqによる剰余類$\mathbb{Z}_q$

ある整数nを整数qで割った余りをrとする。nをrと同一視することで得られる集合を、整数$\mathbb{Z}$のqによる剰余類と呼び、$\mathbb{Z}_q$で表す。

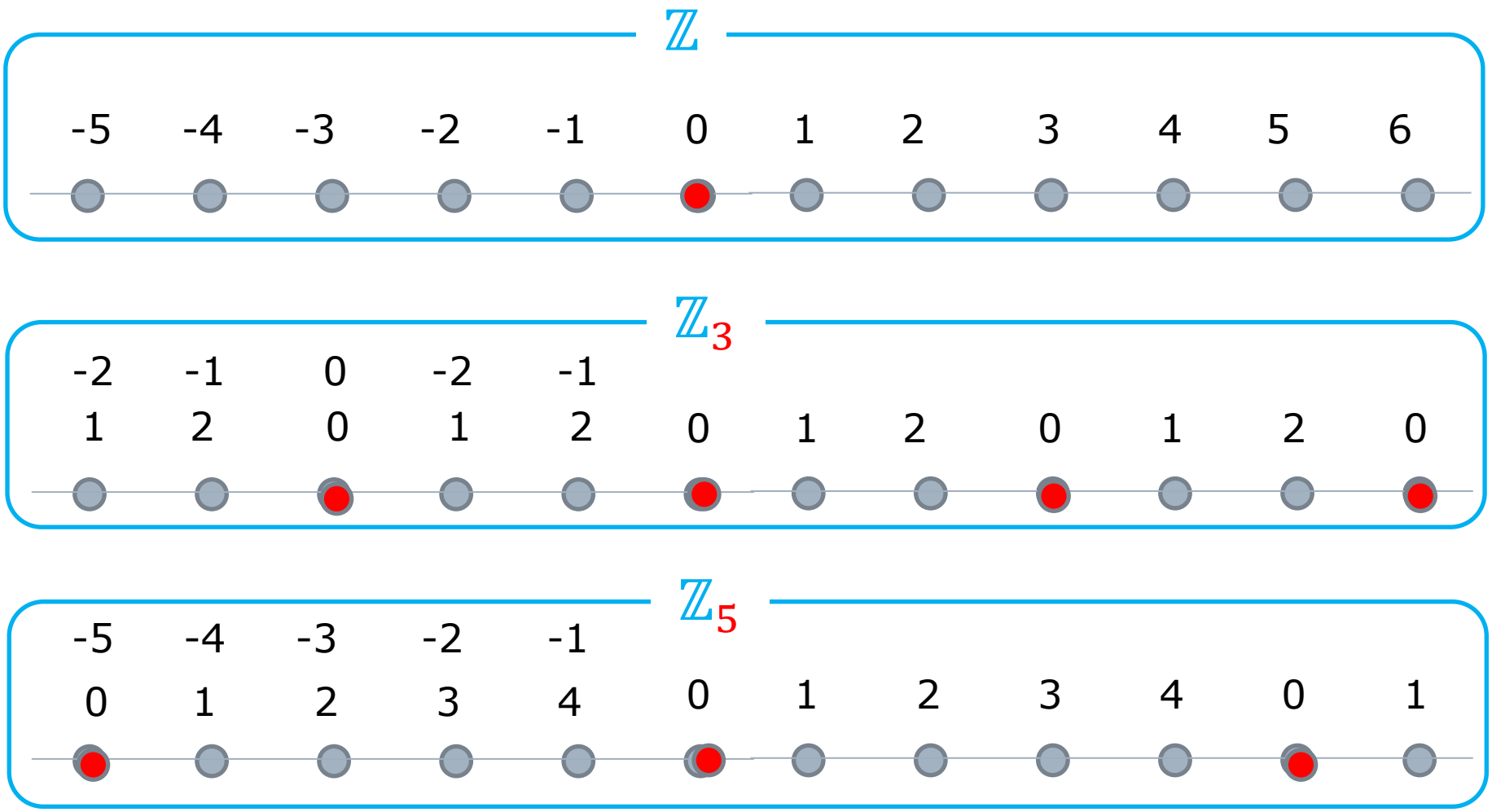
qで割った余りは、$\{0, 1, \dots, q-1\}$のq個しかないので、$\mathbb{Z}_q$は、q個の要素からなる。また、どんな整数も、このq個の類のどれかに属することになる。

例えば、$q = 3$の場合、

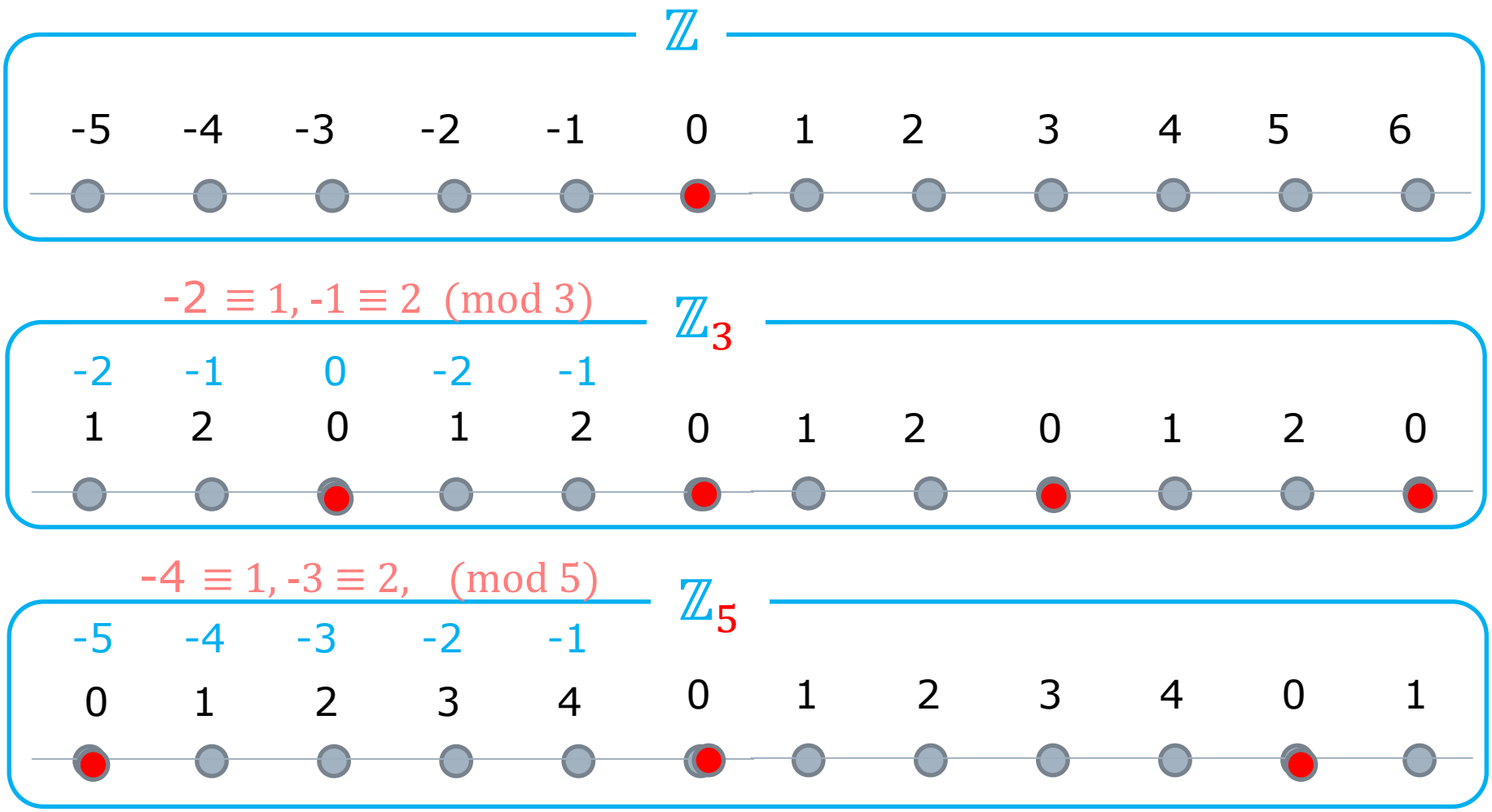
$$0 \equiv 0, \quad 1 \equiv 1, \quad 2 \equiv 2 \quad \boldsymbol{mod(3)}$$
$$3 \equiv 0, \quad 4 \equiv 1, \quad 5 \equiv 2 \quad \boldsymbol{mod(3)}$$
$$6 \equiv 0, \quad 7 \equiv 1, \quad 8 \equiv 2 \quad \boldsymbol{mod(3)}$$
$$\dots\dots$$

ここで、$\{ 0, 1, 2 \} \in \mathbb{Z}_3$ である。

$\mathbb{Z}$ と $\mathbb{Z}_3$ と $\mathbb{Z}_5$



$\mathbb{Z}$ と $\mathbb{Z}_3$ と $\mathbb{Z}_5$



$\mathbb{Z}^n$ 上のラティス

以前に、n次元の$\mathbb{Z}^n$上のラティスを次のように定義した。
n次元のラティスの基底を$b_1, b_2, \ldots, b_n$とする。
ラティスの各点は、n個の基底$b_1, b_2, \ldots, b_n$の整数倍の和で表される。

$$L(b_1, b_2, \ldots, b_n) = \sum_{i=1}^{n} b_i \mathbb{Z}$$

すなわち、$\mathbb{Z}^n$上のラティスの格子点は、

$$n_1, n_2, \ldots, n_n \in \mathbb{Z} \text{ の時、}$$
$$L(b_1, b_2, \ldots, b_n) = b_1 n_1 + b_2 n_2 + \cdots + b_n n_n$$

と表される。

$\mathbb{Z}_q^n$上のラティス

n次元の$\mathbb{Z}_q^n$上のラティスを次のように定義する。

n次元のラティスの基底を$b_1, b_2, \dots, b_n$とする。

ラティスの各点は、n個の基底$b_1, b_2, \dots, b_n$の$\mathbb{Z}_q$倍の和で表される。

$$L(b_1, b_2, \dots, b_n) = \sum_{i=1}^{n} b_i \mathbb{Z}_q$$

すなわち、$\mathbb{Z}_q^n$上のラティスの格子点は、

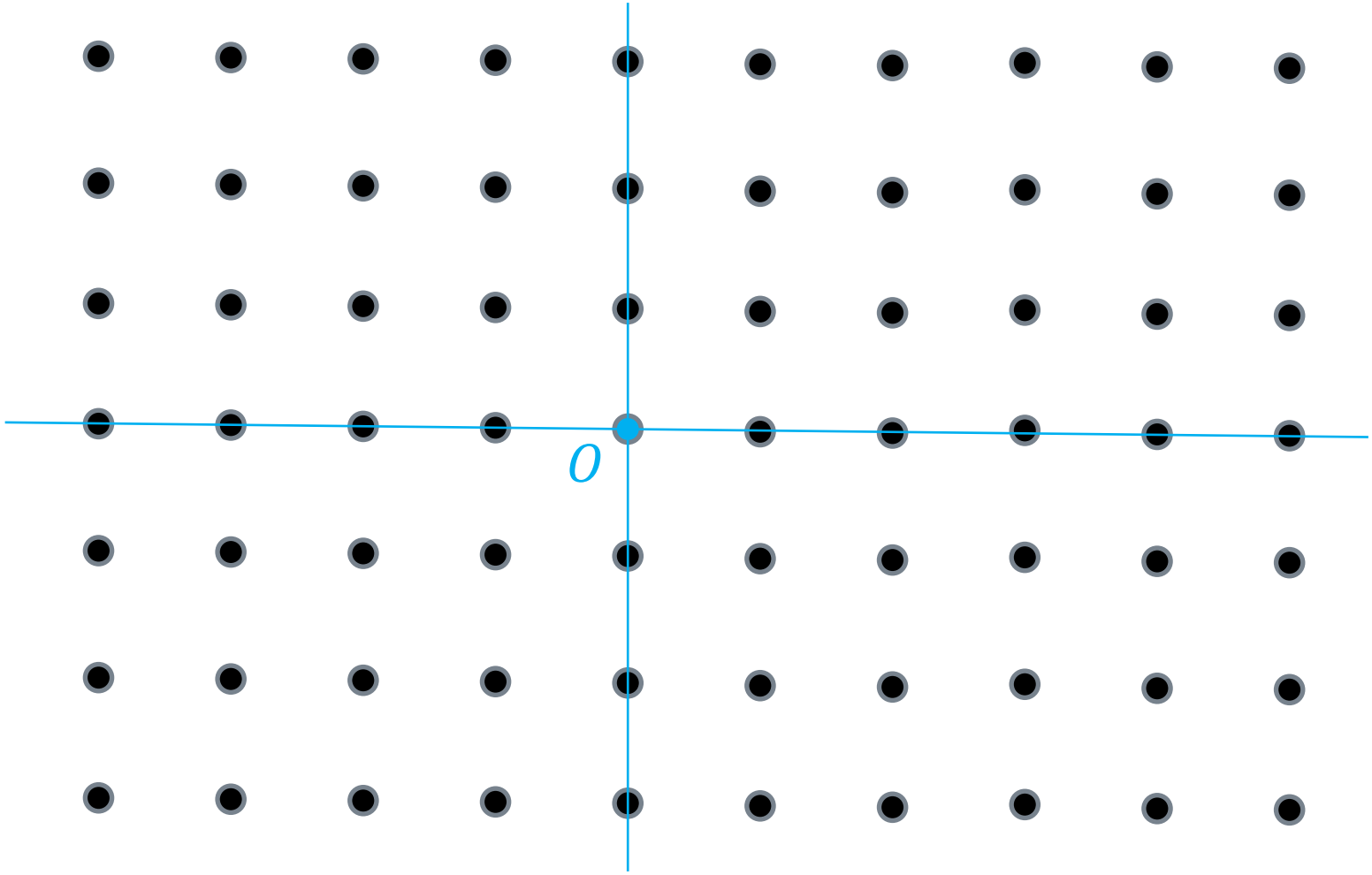
$n_1, n_2, \dots, n_n \in \mathbb{Z}_q$ の時、

$$L(b_1, b_2, \dots, b_n) = b_1 n_1 + b_2 n_2 + \cdots + b_n n_n$$

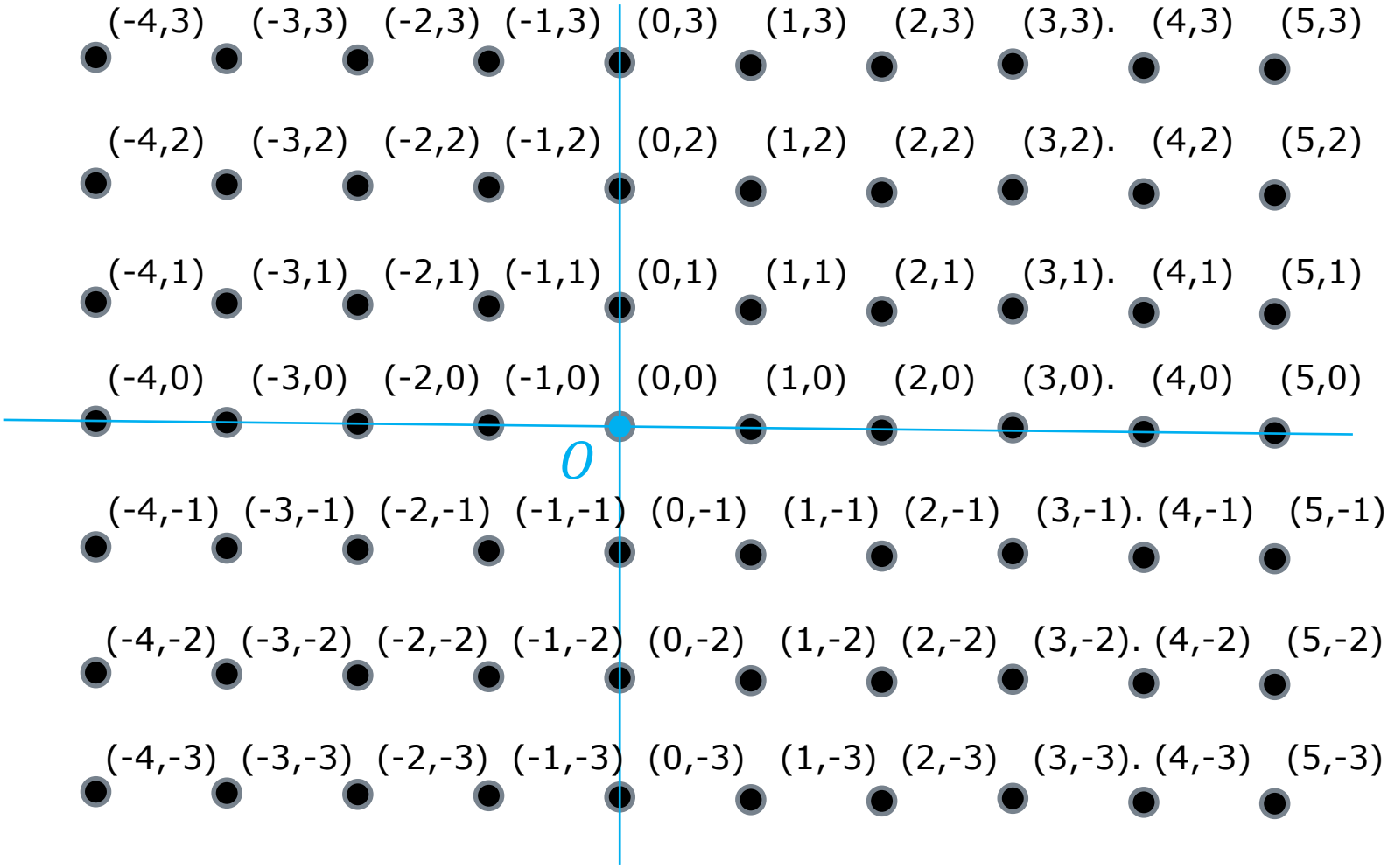
と表される。

$\mathbb{Z}^2$ 上のラティスの格子点

$n_1, n_2 \in \mathbb{Z}$ 、$L(b_1, b_2) = b_1 n_1 + b_2 n_2$

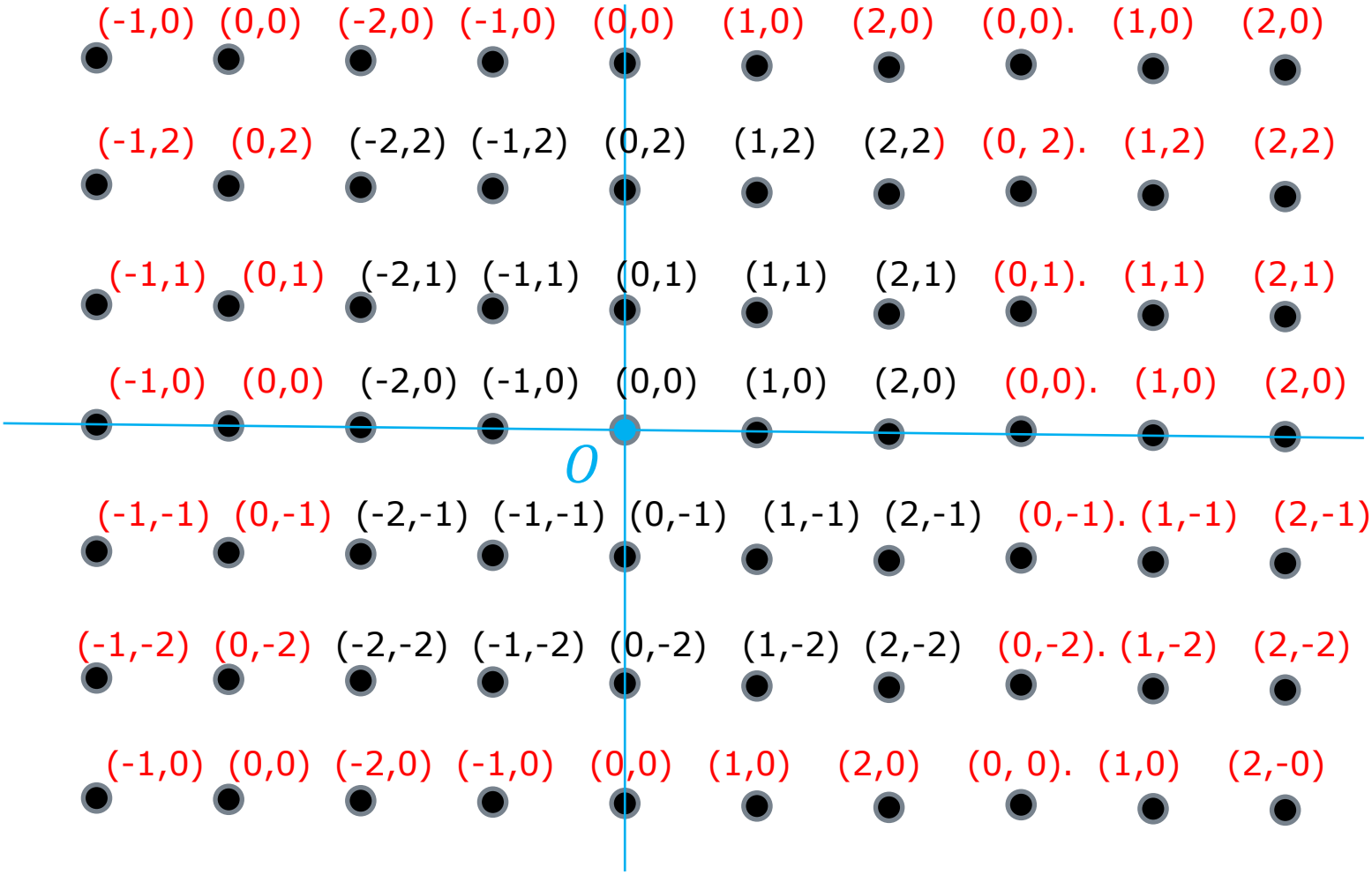


$\mathbb{Z}^2$ 上のラティス
格子点に整数の係数n_1, n_2を入れる

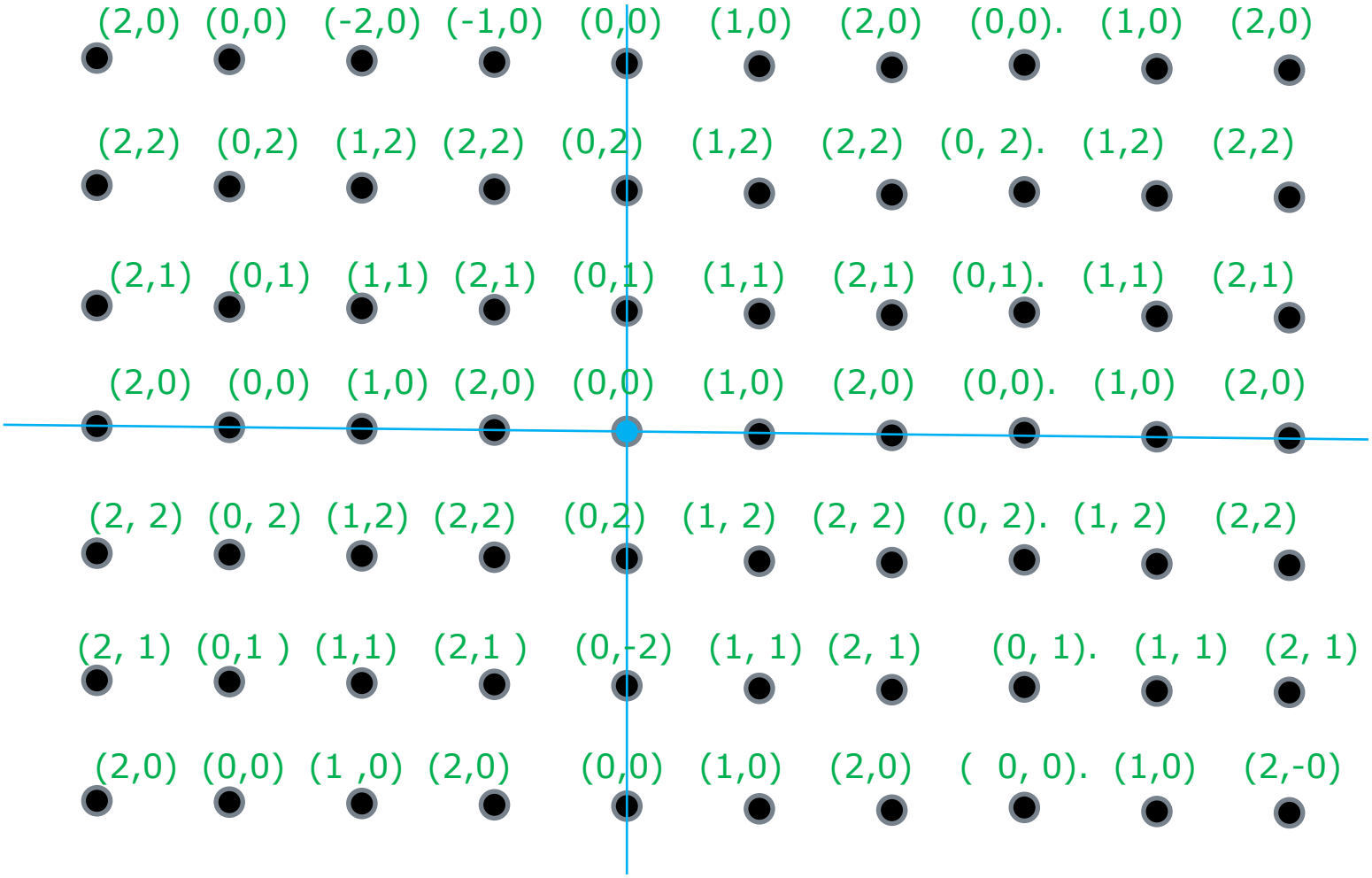


$\mathbb{Z}^2$ 上のラティス→ $\mathbb{Z}_3^2$ 上のラティス

$n_i \leq -3, n_i \geq 3$ を$\mathbb{Z}_3$の要素に

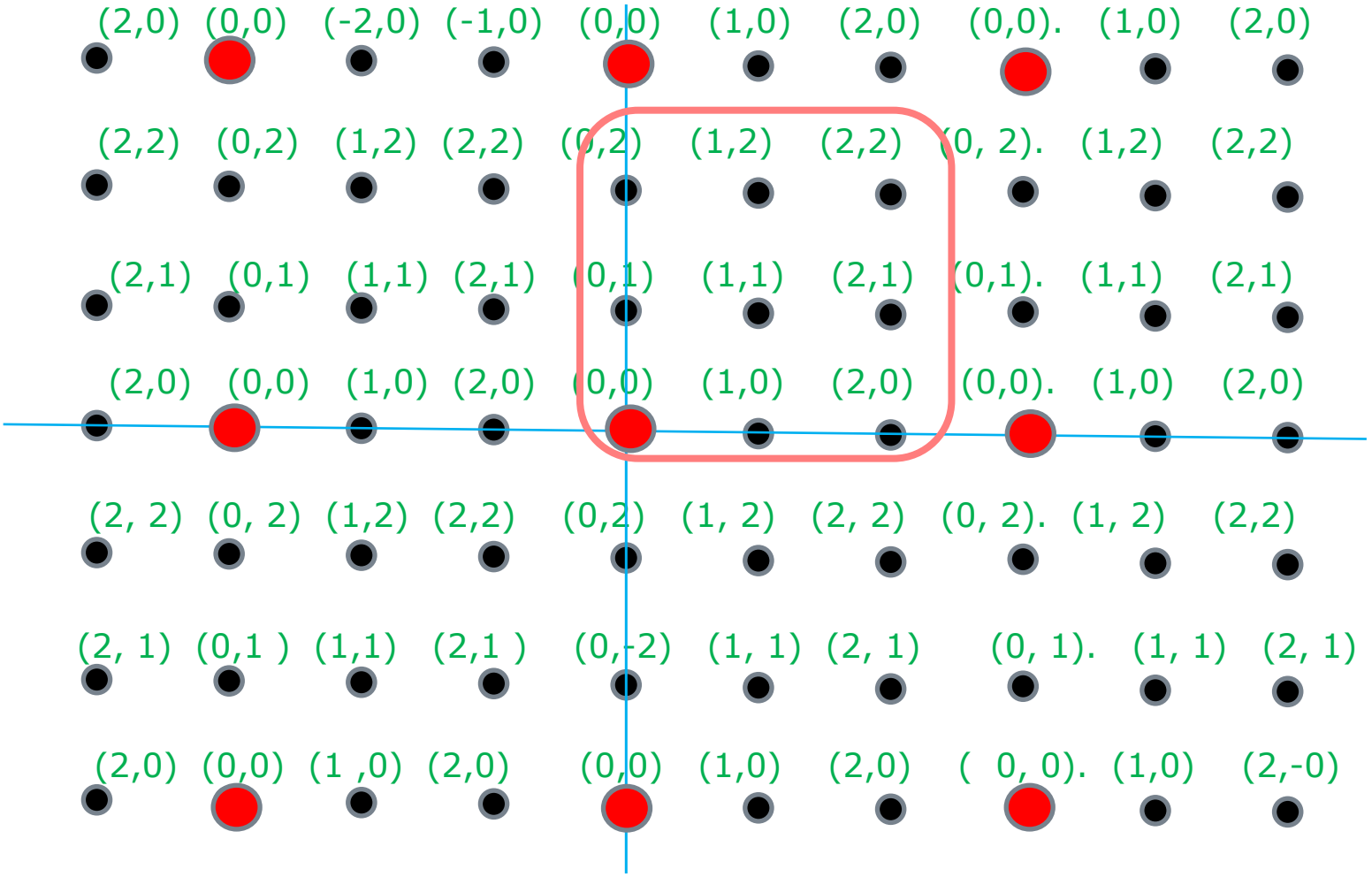


$\mathbb{Z}^2$ 上のラティス→ $\mathbb{Z}_3^2$ 上のラティス
$-2 \to 1, \quad -1 \to 2$

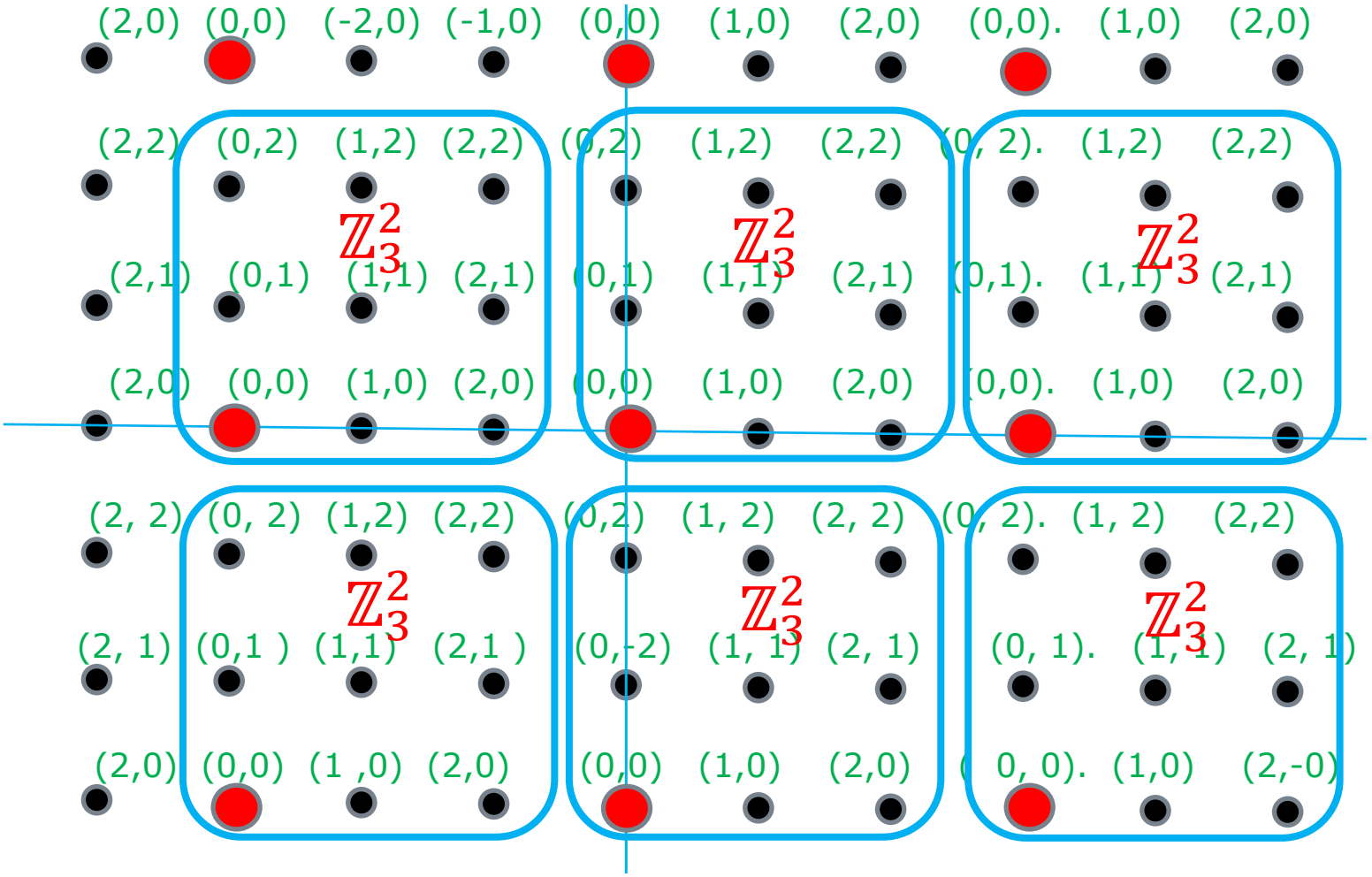


$\mathbb{Z}_3^2$上のラティス

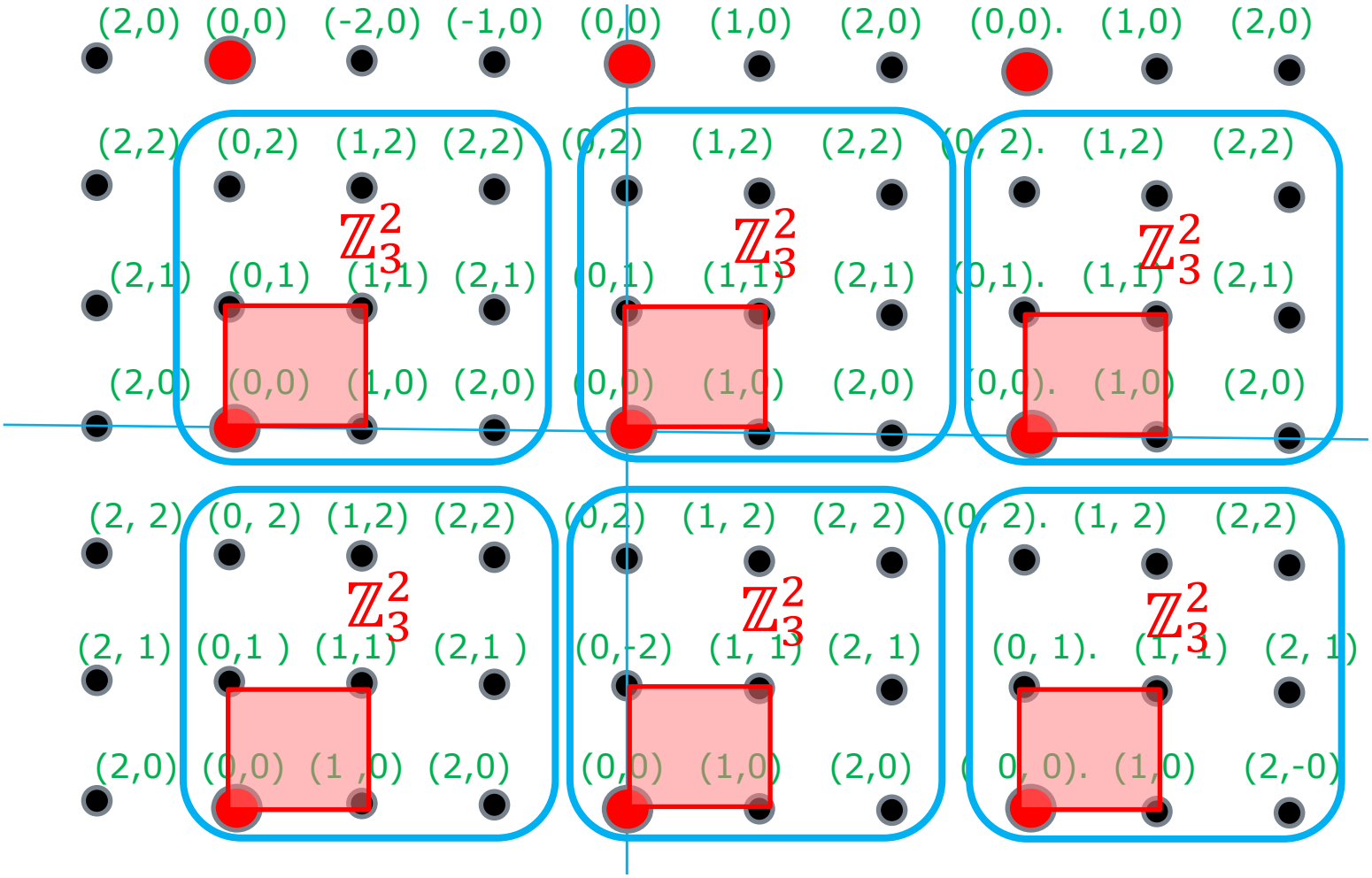
$n_1, n_2 \in \mathbb{Z}_3$ 、$L(b_1, b_2) = b_1 n_1 + b_2 n_2$



$\mathbb{Z}_3^2$上のラティス



$\mathbb{Z}_3^2$上のラティスとその基本領域



Gram-Schmidt 直交化

Gram-Schmidt 直交化
ベクトル空間の基底を直交化する

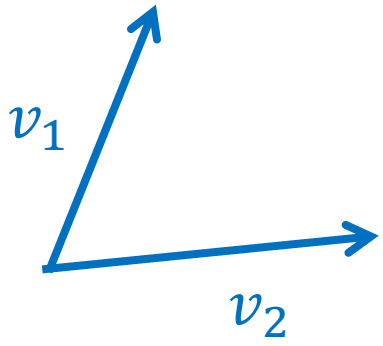
ここでは、ラティスの話題をいったん離れる。

一般のベクトル空間の基底を「直交化」する、Gram-Schmidtの方法について述べる。

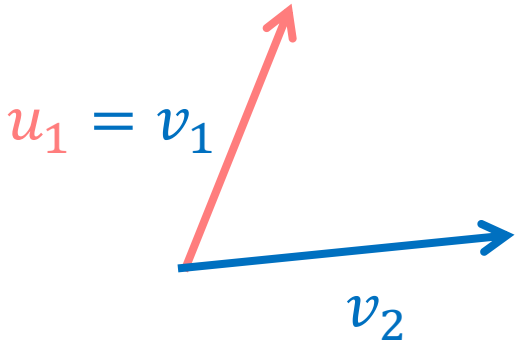
一般のベクトル空間では、与えられた任意の基底をGram-Schmidtの方法で直交化できる。

後で見るように、ラティスの基底をこの方法で直交化しても、その基底が同じラティスを張るとは限らない。

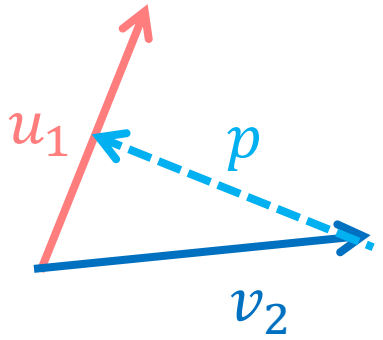
基底 $B(v_1, v_2)$ を $< u_1, u_2 > = 0$ となる直交する基底 $B(u_1, u_2)$ に変える



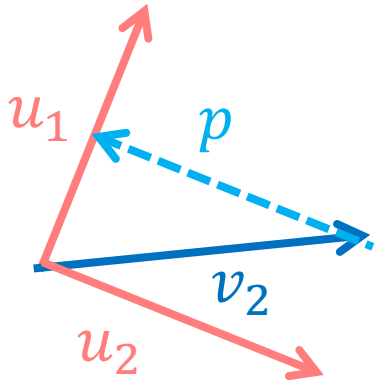
基底v_1は残してu_1とする

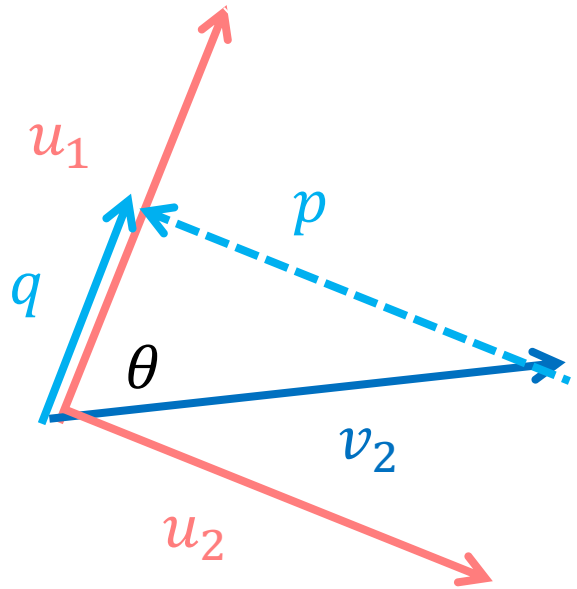


v_2からu_1に垂線pをおとす



Pを逆転したものをu_2とすればいい





u_1とv_2のなす角度をθ、 v_2からu_1に落とした垂線pの足をqとする。

$$q = \frac{|q|}{|u_1|}u_1 = \frac{|v_2|\cos\theta}{|u_1|}u_1$$

$q = v_2 + p$ だから、

$$p = q - v_2 = \frac{|v_2|\cos\theta}{|u_1|}u_1 - \mathrm{v}_2 = \frac{|v_2||u_1|\cos\theta}{|u_1||u_1|}u_1 - \mathrm{v}_2 = \frac{< u_1, v_2 >}{< u_1, u_1 >}u_1 - v_2$$

u_2はpの符号を逆にしたもの。$u_2 = -p$

$$u_2 = v_2 - \frac{<u_1,v_2>}{<u_1,u_1>}u_1$$

三次元ベクトルの場合の Gram-Schmidt 直交化

元の基底を v_1, v_2, v_3 とする。

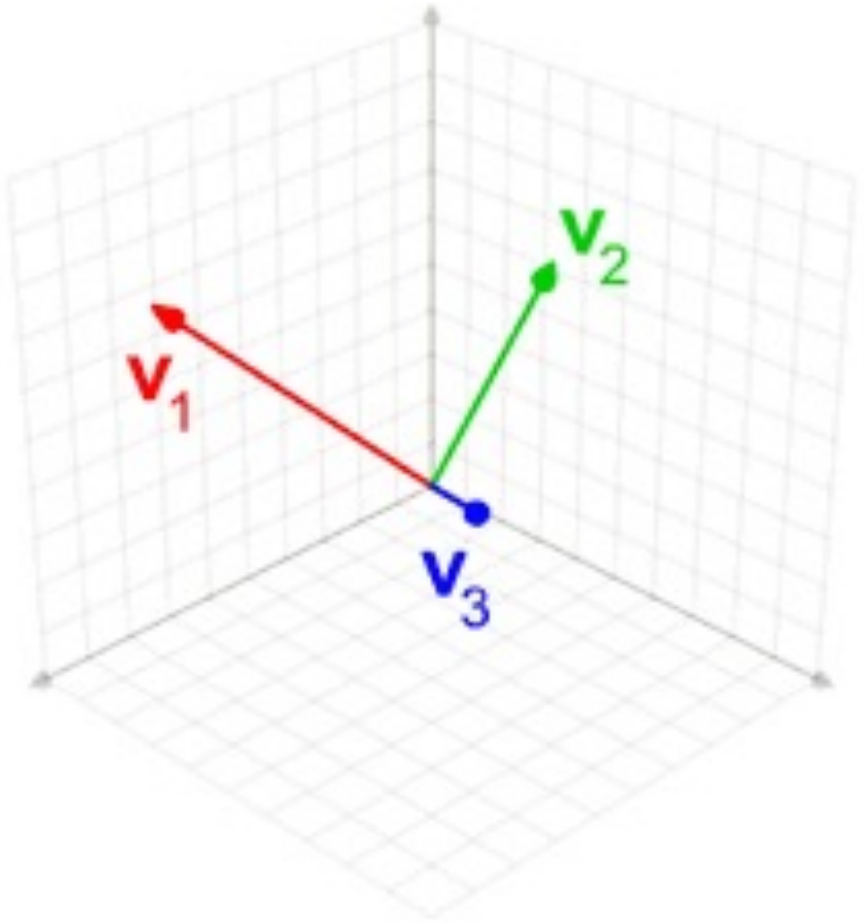
$$u_1 = v_1$$

$$u_2 = v_2 - \frac{<u_1, v_2>}{<u_1, u_1>} u_1$$

ここまでは、二次元の場合と同じである。
今度は、v_3から、u_1, u_2に垂線p_1, p_2をおろす。

$$u_3 = v_3 - \frac{< u_1, v_3 >}{< u_1, u_1 >} u_1 - \frac{< u_2, v_3 >}{< u_2, u_2 >} u_2$$

wikipediaの動画を見てみよう。


v2
v1
v3

基底 $\begin{bmatrix}0 & 2\\ 2 & 1\end{bmatrix}$ の直交化 $\Rightarrow \begin{bmatrix}0 & 2\\ 2 & 0\end{bmatrix}$

$$B\left(\begin{bmatrix}0 & 2\\ 2 & 1\end{bmatrix}\right), v_1 = \begin{bmatrix}0\\ 2\end{bmatrix}, v_2 = \begin{bmatrix}2\\ 1\end{bmatrix}$$

$$u_1 = v_1$$

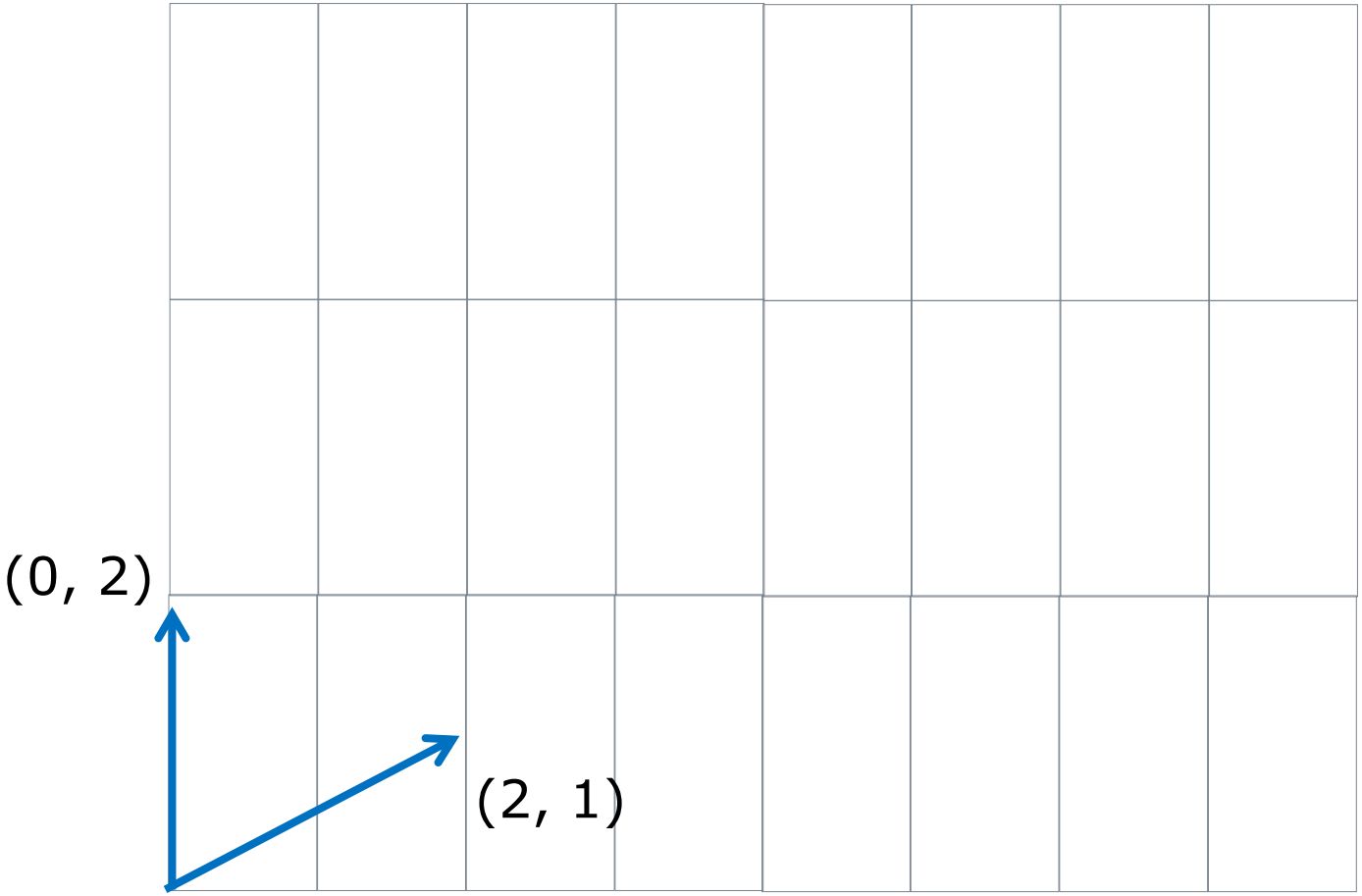
$$u_2 = v_2 - \frac{< u_1, v_2 >}{< u_1, u_1 >} u_1$$

$$u_2 = v_2 - \frac{< \begin{bmatrix}0\\ 2\end{bmatrix}, \begin{bmatrix}2\\ 1\end{bmatrix} >}{< \begin{bmatrix}0\\ 2\end{bmatrix}, \begin{bmatrix}0\\ 2\end{bmatrix} >} u_1 = v_2 - \frac{0\times 2 + 2\times 1}{0\times 0 + 2\times 2} u_1$$

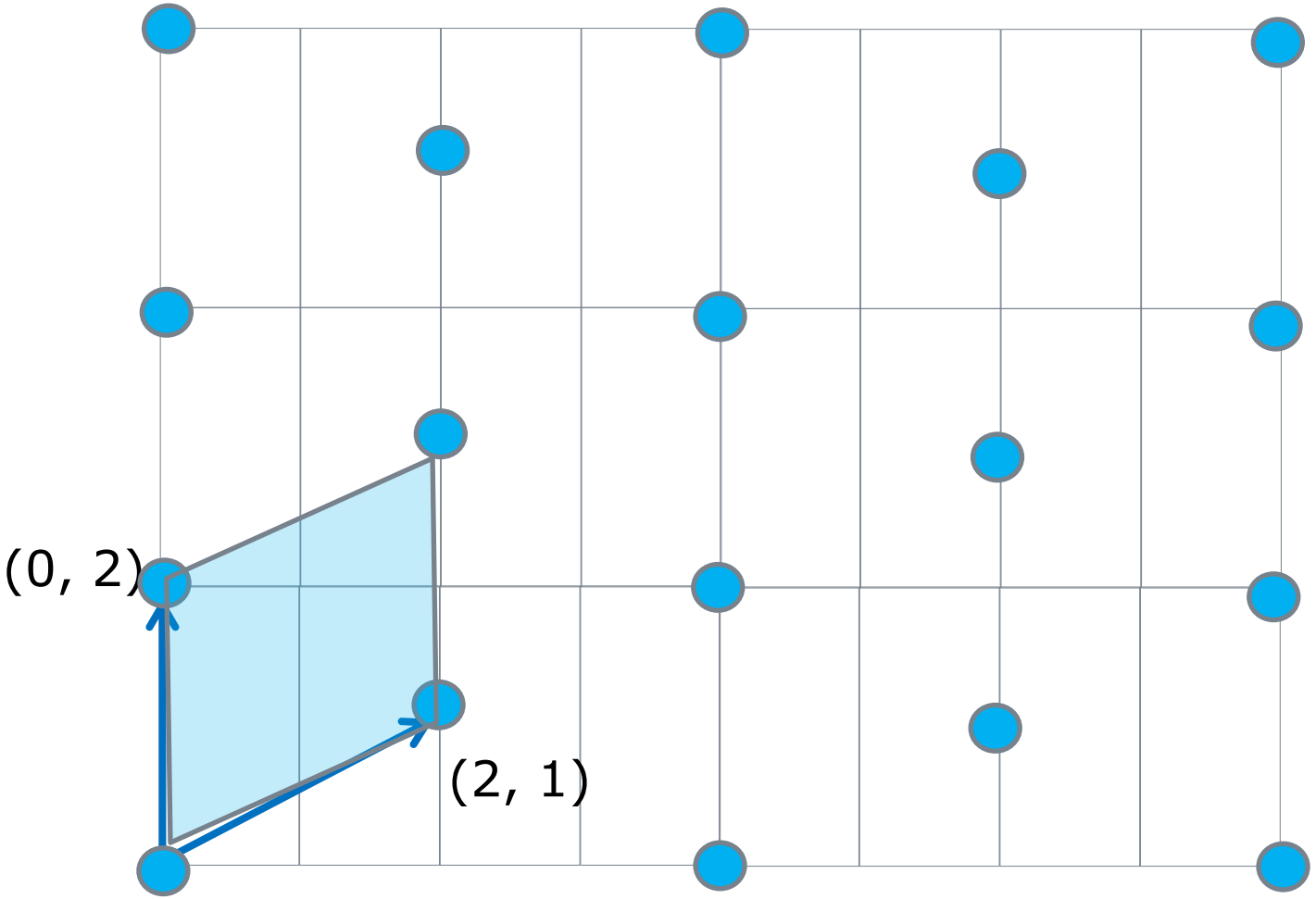
$$= v_2 - \frac{2}{4} u_1 = \begin{bmatrix}2\\ 1\end{bmatrix} - \frac{1}{2}\begin{bmatrix}0\\ 2\end{bmatrix} = \begin{bmatrix}2\\ 1\end{bmatrix} - \begin{bmatrix}0\\ 1\end{bmatrix} = \begin{bmatrix}2\\ 0\end{bmatrix}$$

$$B_{\perp}\left(\begin{bmatrix}0 & 2\\ 2 & 1\end{bmatrix}\right) = B(u_1, u_2) = B\left(\begin{bmatrix}0 & 2\\ 2 & 0\end{bmatrix}\right)$$

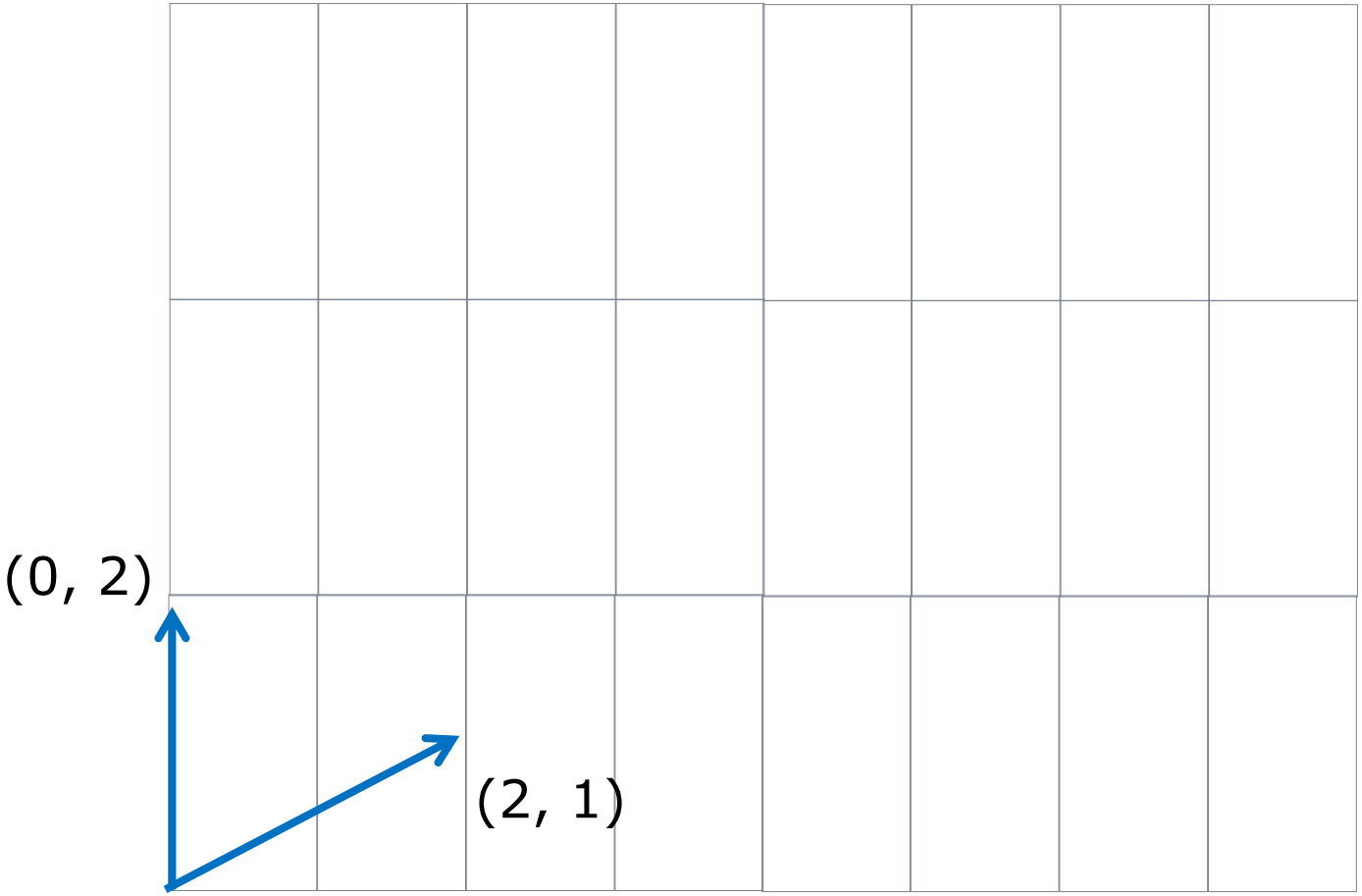
基底 $\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix}$ で張られるラティス



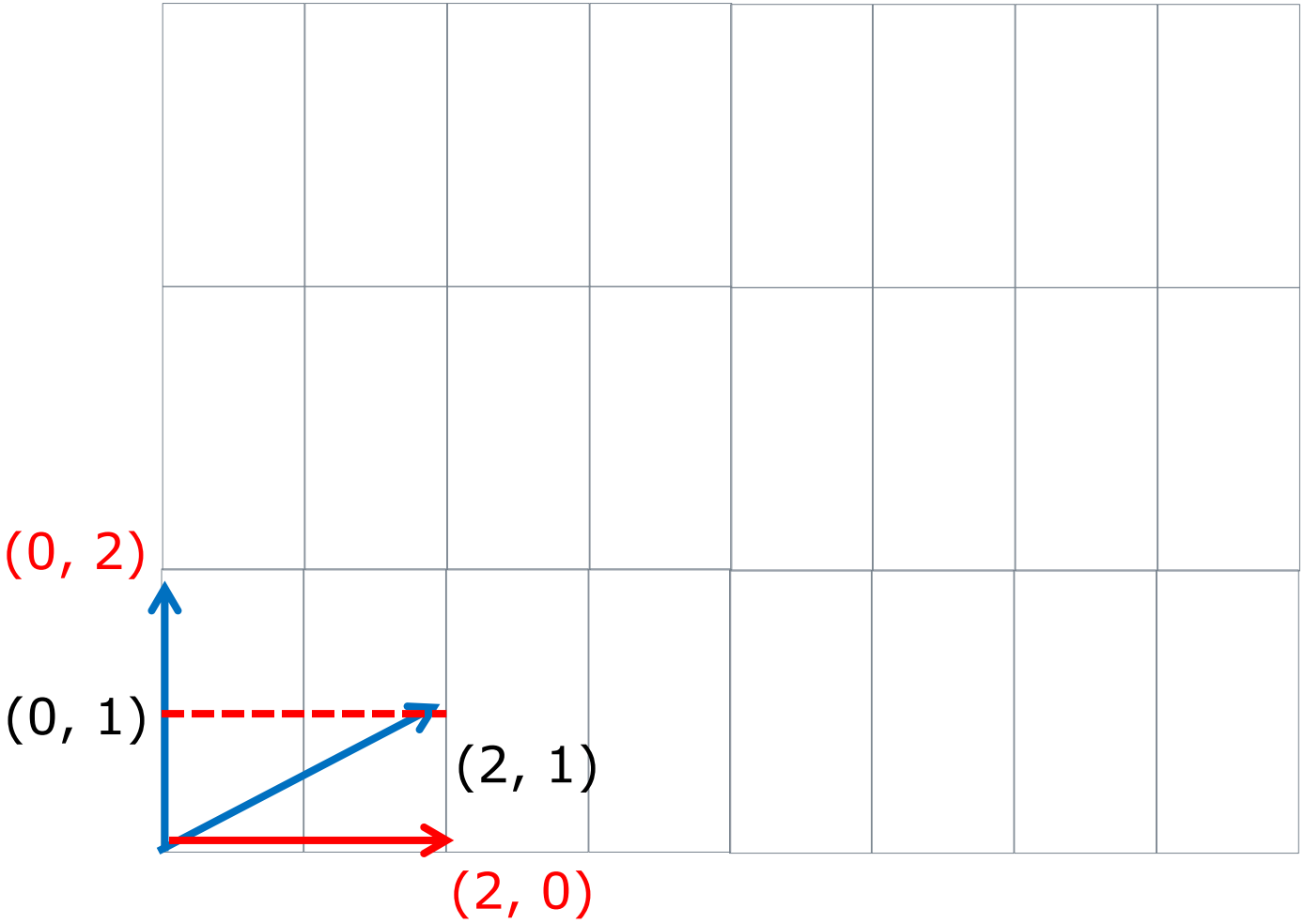
ラティス $l\left(\begin{bmatrix}0 & 2\\ 2 & 1\end{bmatrix}\right)$



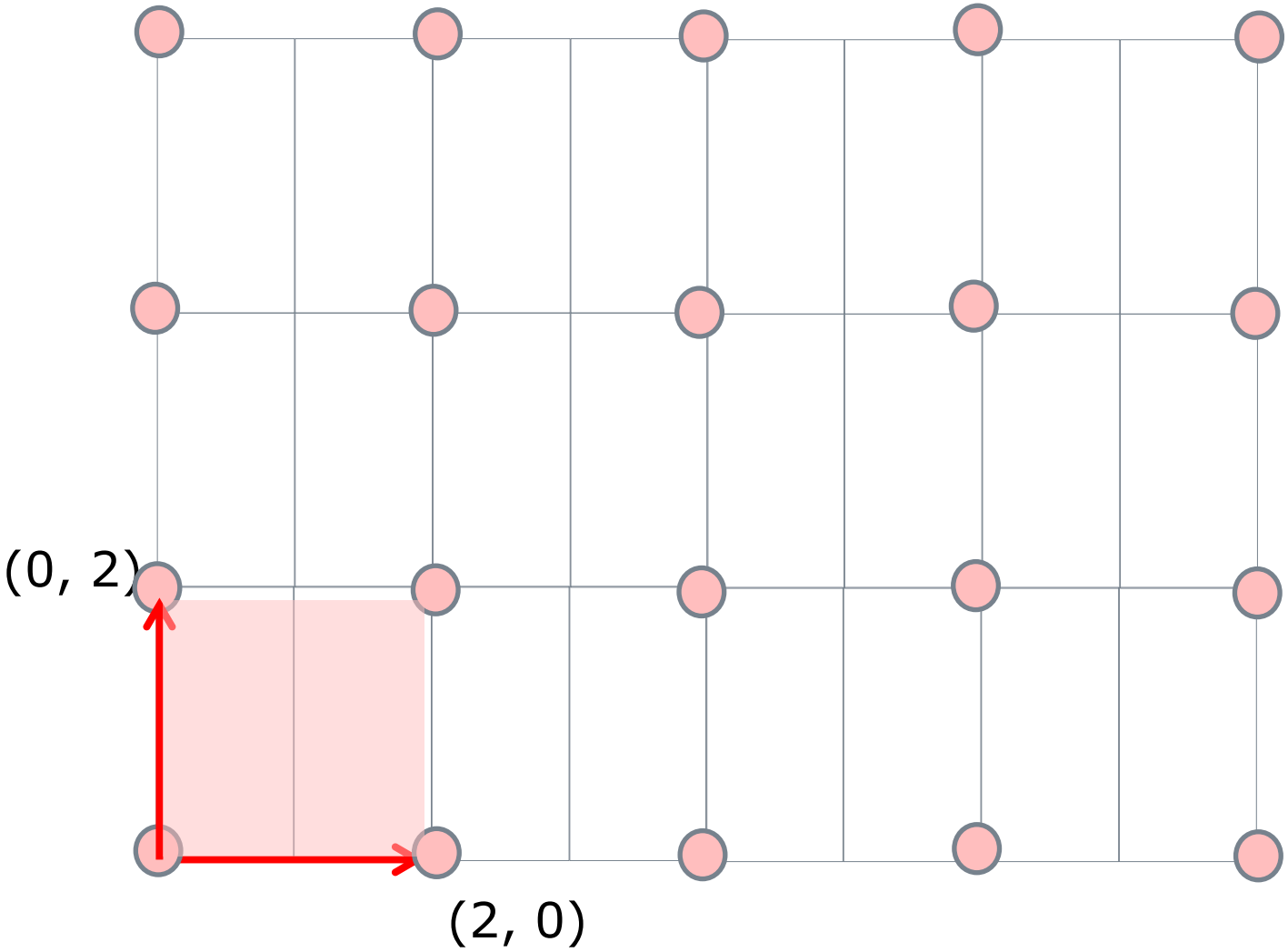
基底 $\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix}$ で張られるベクトル空間



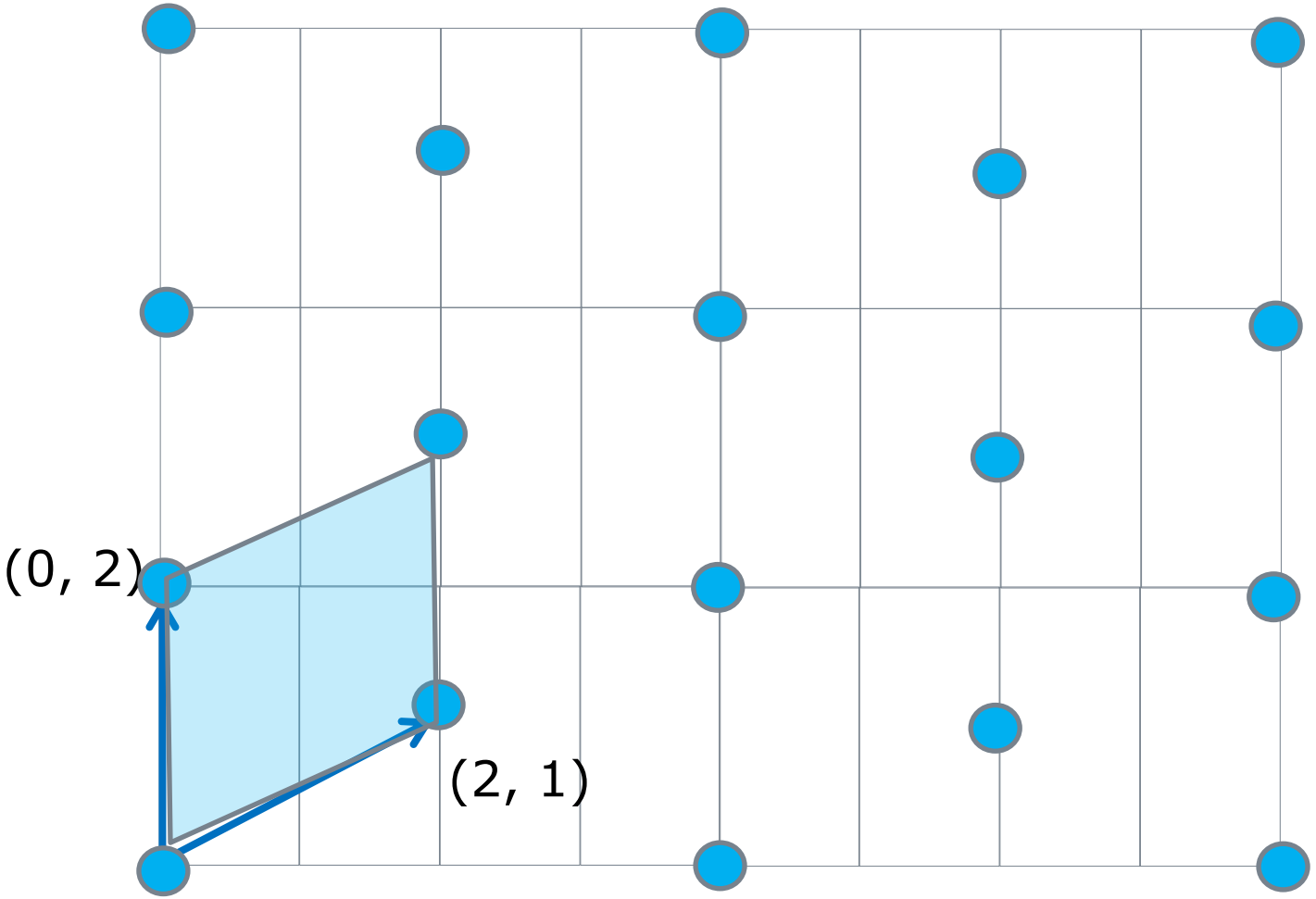
基底 $\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix}$ の直交基底 $\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$



直交基底 $\begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix}$ の張るラティス



基底 $\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix}$ で張られるラティス




© NASA, ESA/CSA/Judy Sch


Part II
ラティス暗号 LWE

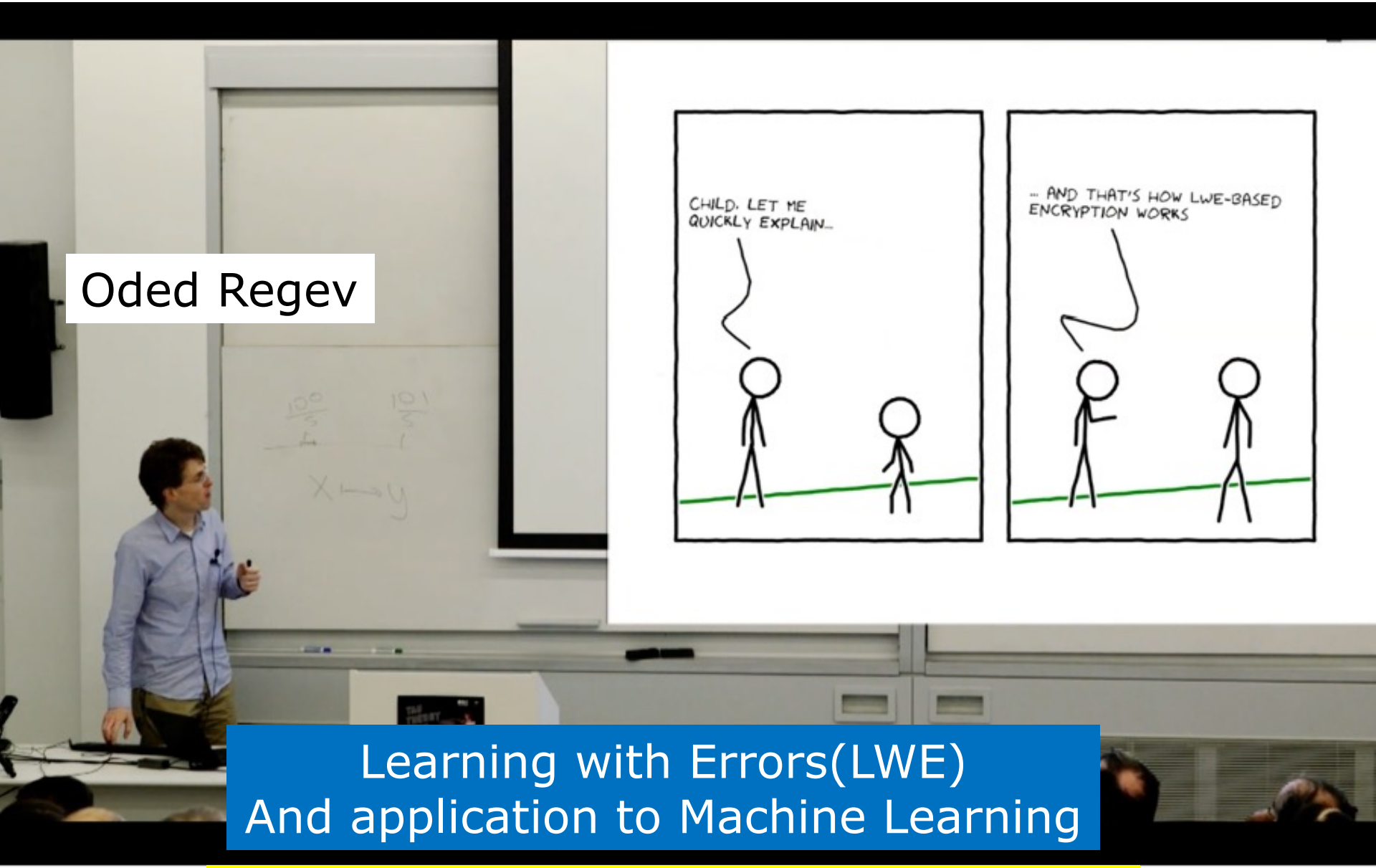
単純な例で学ぶ LWE (1)

ラティス暗号 Learning with Errors (LWE)

このセッションでは、ラティス暗号の代表格である Learning with Errors (LWE) を、分かりやすいサンプルで説明しようと思う。

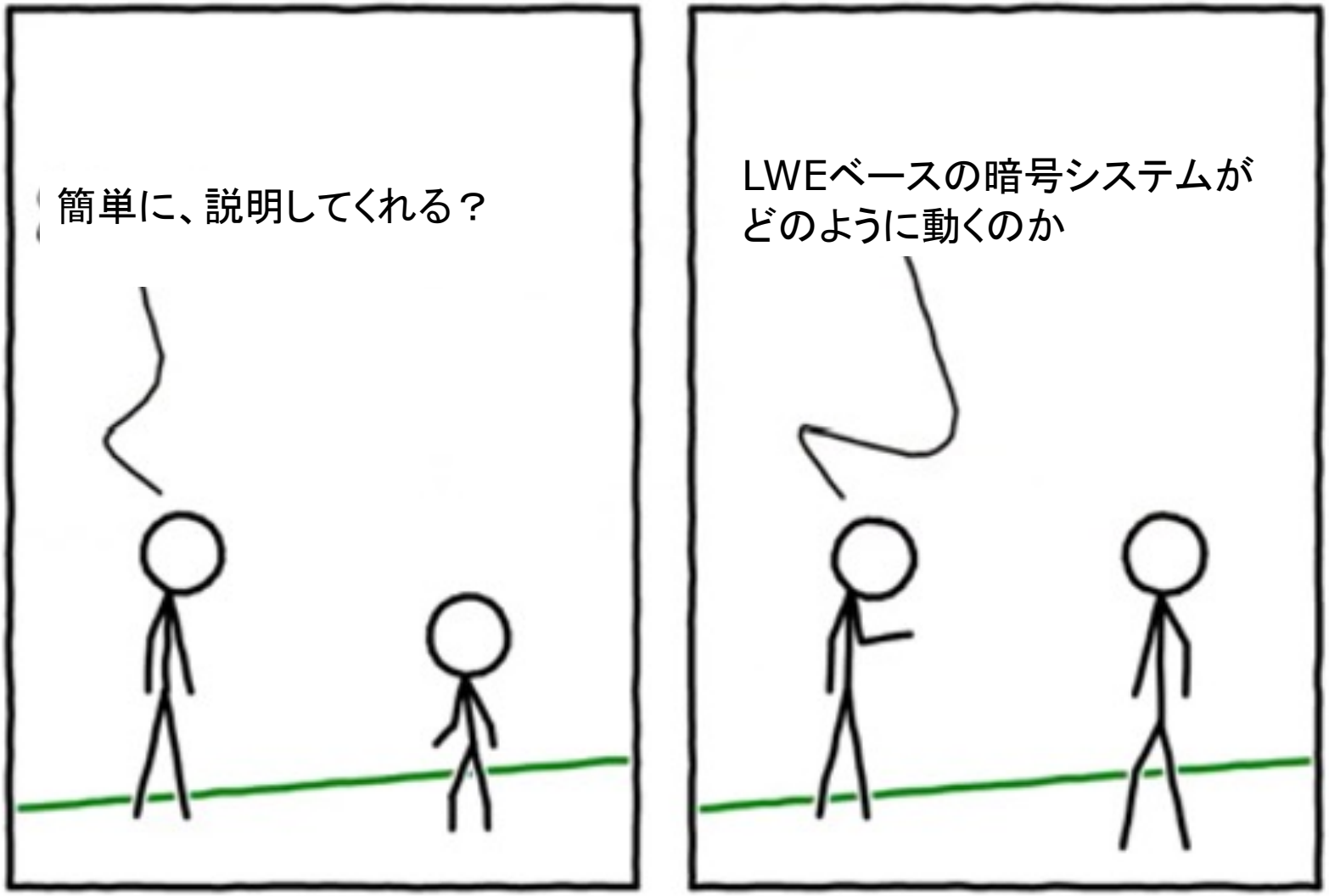
まず、LWE暗号の働きの基本的なイメージを持つことが大事だと考えている。単純なサンプルをいくつか見た後で、あとのセッションで理論的に整理したい。

最初は、LWE暗号の創始者 Oded Regev 自身による、「高校生にもわかる」という「もっとも簡単なLWEのサンプル」を紹介する。



Learning with Errors(LWE) And application to Machine Learning

https://www.youtube.com/watch?v=Ut1FPvxC7mA


簡単に、説明してくれる？
LWEベースの暗号システムが
どのように動くのか

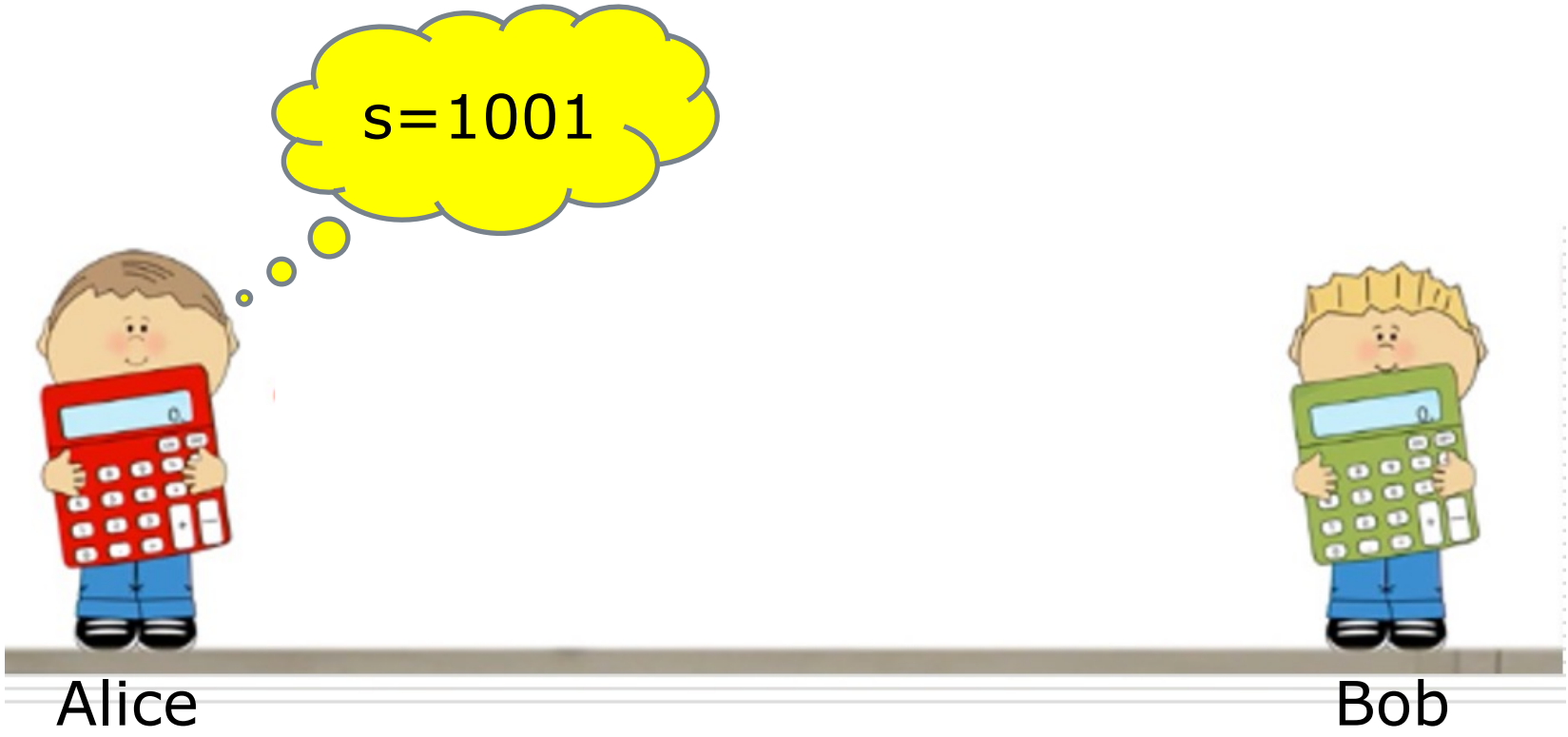
もっとも単純な LWEベースの公開キー暗号システム

Aliceが秘密キーを選ぶ

- Aliceは、秘密キー s を選ぶ。

この例では、秘密キーはランダムな奇数の整数とする。

Aliceは、秘密キーとして、s=1001 を選んだとしよう。



Aliceは公開キーを作ってBobに送る

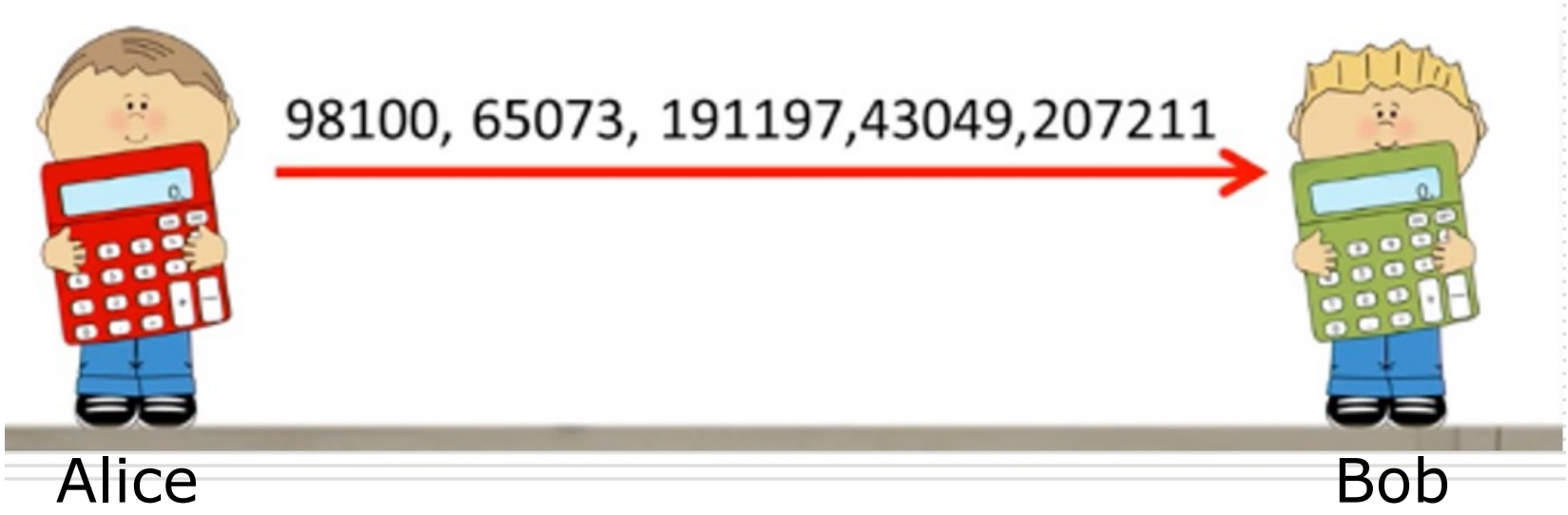
- Aliceは公開キーを作ってBobに送る

公開キーは、秘密キー sにランダムな整数を掛けたものに、小さなランダムな偶数を加えたもののリストである。

Aliceは公開キーを作ってBobに送る

- Aliceは公開キーを作ってBobに送る

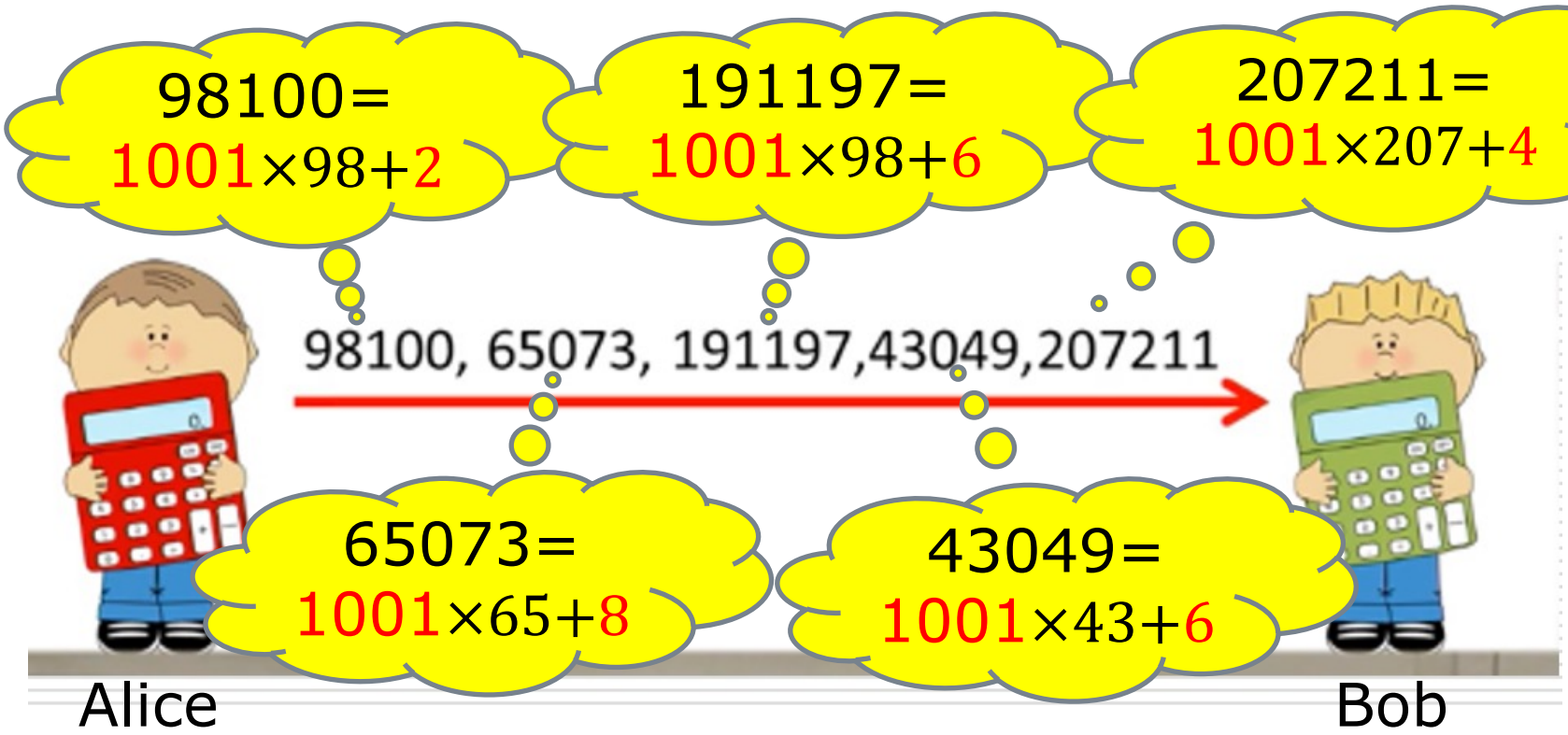
公開キーは、秘密キー sにランダムな整数を掛けたものに、小さなランダムな偶数を加えたもののリストである。



Aliceは公開キーを作ってBobに送る

- Aliceは公開キーを作ってBobに送る

公開キーは、秘密キー sにランダムな整数を掛けたものに、小さなランダムな偶数を加えたもののリストである。

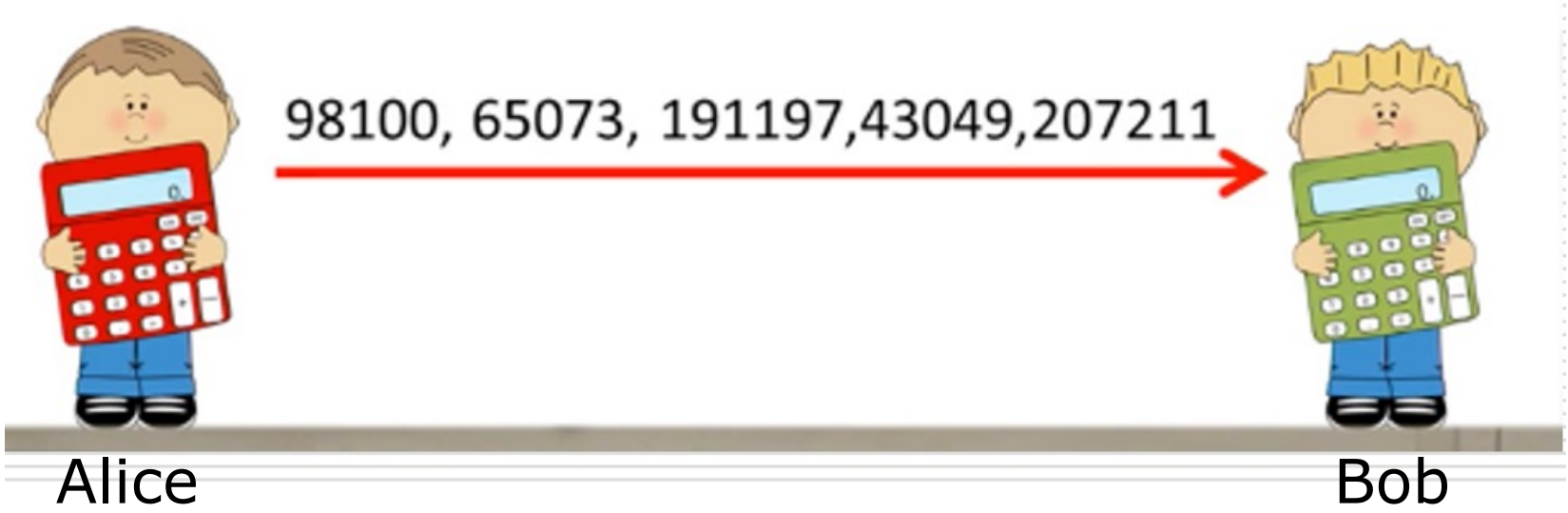


Bobは $b \in \{0,1\}$ をエンコードする

- Bobは $b \in \{0,1\}$ をエンコードする

Bobは、公開キーのリストから、何個かの数字をランダムに選ぶ。この例では、公開キーのリストは、5個の数字からなるので、ランダムに2個から5個までの数字を選ぶことができる。

この数字を全部加えて、さらにbを加えたものxが、エンコードされた値になる。



Bobは $b \in \{0,1\}$ をエンコードする サンプル 1

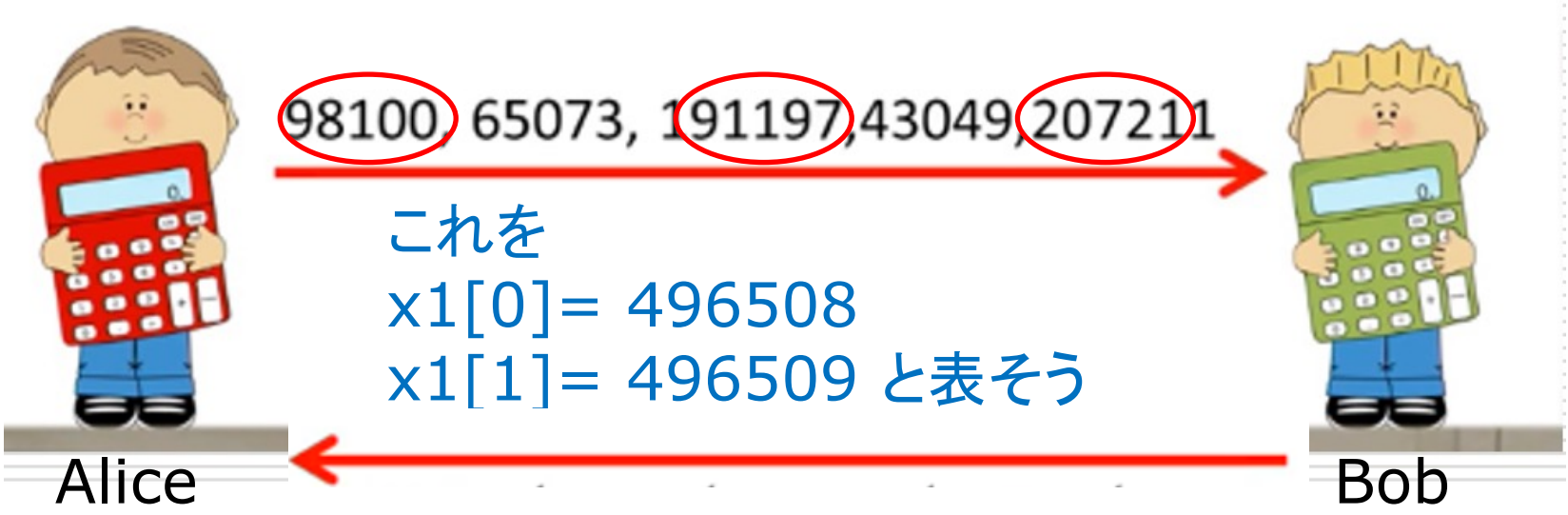
- Bobは $b \in \{0,1\}$ をエンコードする

Bobは、公開キーのリストから、98100と191197と207211を選んだとしよう。

98100＋191197＋207211＝496508 だから

b=0 は、x=496508+0 = 496508

b=1 は、x=496508+1 = 496509 とエンコードされる。



Bobは $b \in \{0,1\}$ をエンコードする サンプル 2

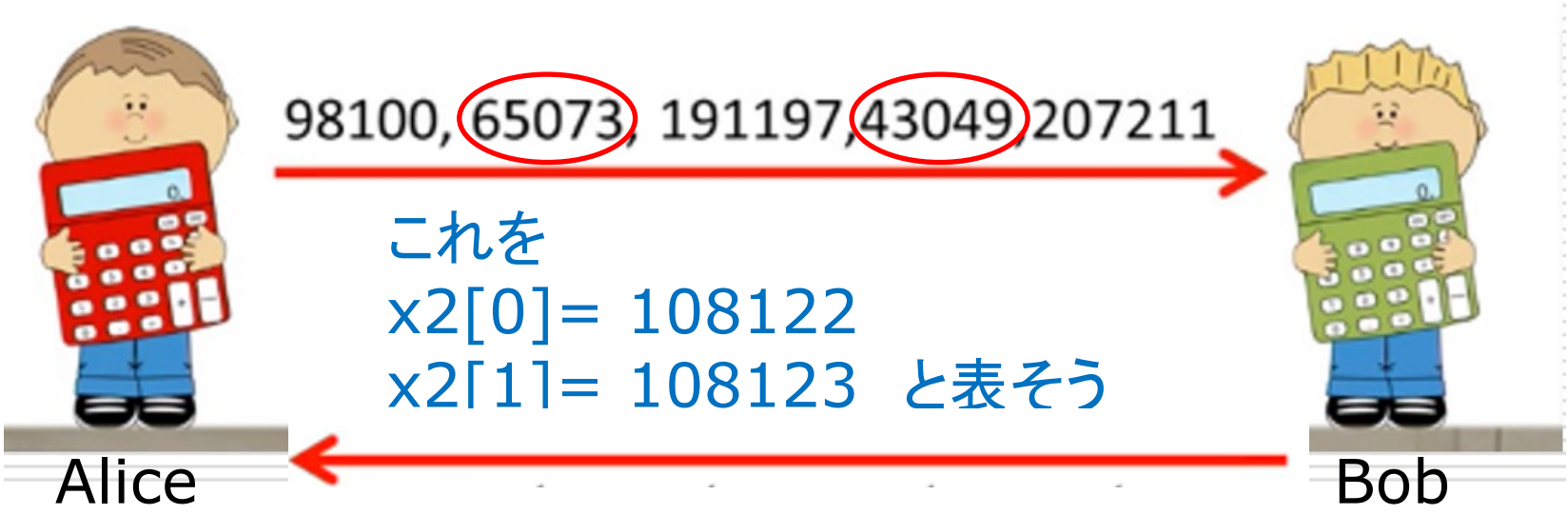
- Bobは $b \in \{0,1\}$ をエンコードする

Bobは、公開キーのリストから、65073と43049を選んだとしよう。

65073＋43049＝108122 だから

b=0 は、x=108122+0 = 108122

b=1 は、x=108122+1 = 108123 とエンコードされる。



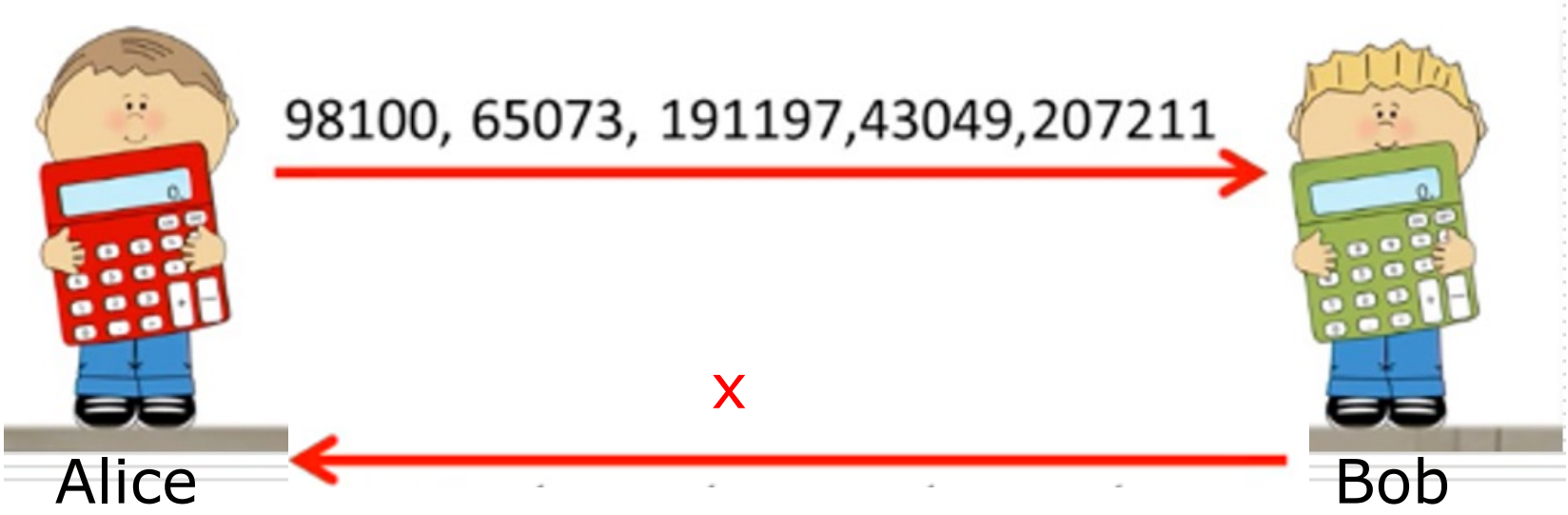
Aliceは受け取ったxをデコードする

- Aliceは受け取ったxをデコードする

Aliceは、受け取ったxを秘密キーのsでわって、その余りをチェックする。

もしも余りが奇数なら、xは1に、

もしも余りが偶数なら、xは0に、デコードされる。



Aliceは受け取ったxをデコードする
サンプル1の場合

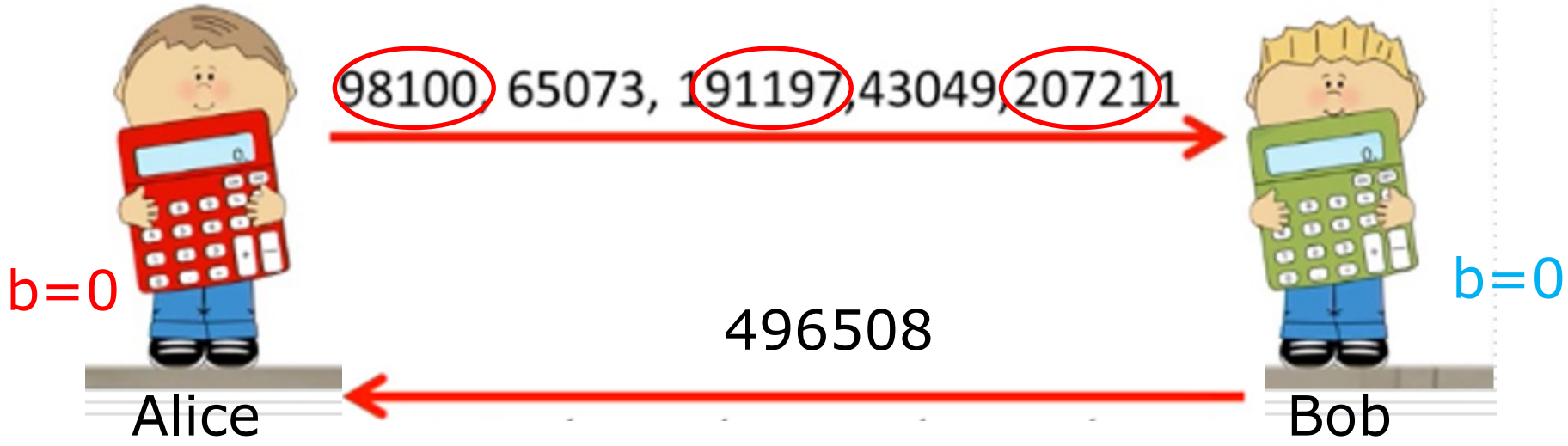
- Aliceは受け取ったxをデコードする

Aliceは、受け取ったx1[0]=496508を秘密キーの1001でわって、その余りをチェックする。

496508=496x1001+12

余りは偶数だからデコード値は0

エンコード値とデコード値は一致する。



Aliceは受け取ったxをデコードする
サンプル1の場合

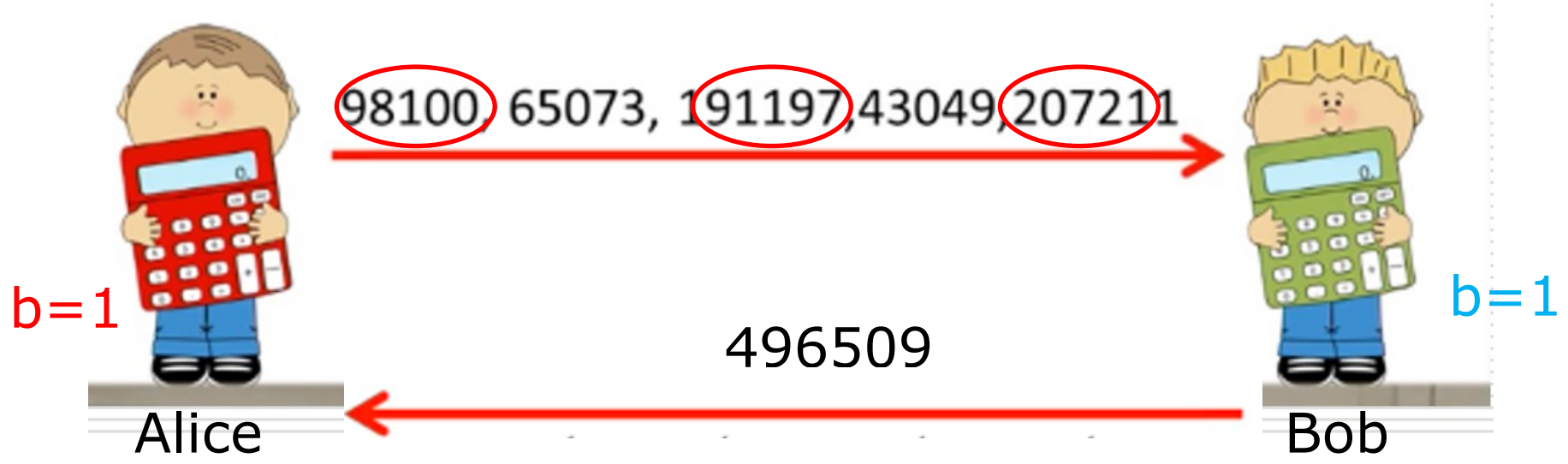
- Aliceは受け取ったxをデコードする

Aliceは、受け取ったx1[1]=496509を秘密キーの1001でわって、その余りをチェックする。

496509=496x1001+13

余りは奇数だからデコード値は1

エンコード値とデコード値は一致する。



Aliceは受け取ったxをデコードする
サンプル2の場合

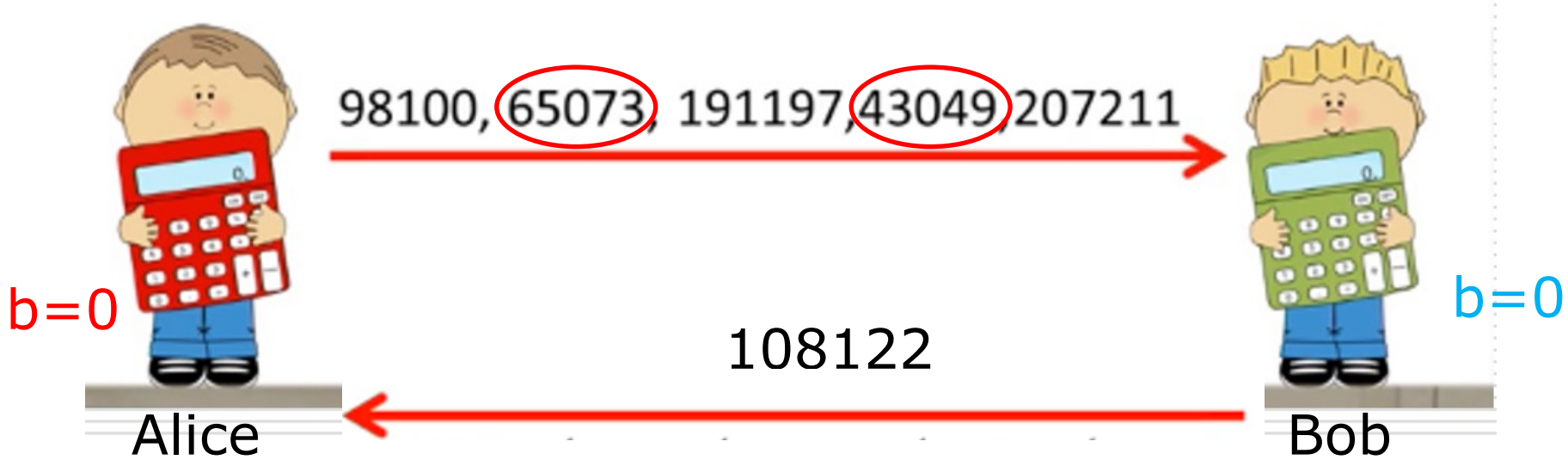
- Aliceは受け取ったxをデコードする

Aliceは、受け取ったx2[0]= 108122を秘密キーの1001でわって、その余りをチェックする。

108122=108x1001+14

余りは偶数だからデコード値は0

エンコード値とデコード値は一致する。



Aliceは受け取ったxをデコードする
サンプル2の場合

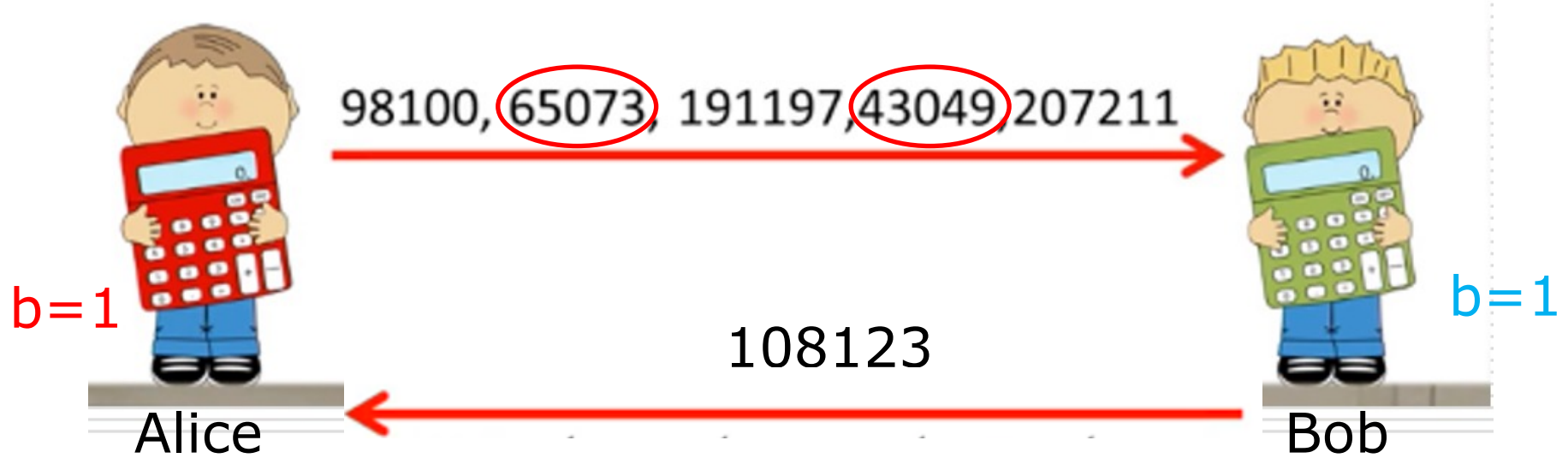
- Aliceは受け取ったxをデコードする

Aliceは、受け取ったx2[1]= 108123を秘密キーの1001でわって、その余りをチェックする。

108123=108x1001+15

余りは奇数だからデコード値は1

エンコード値とデコード値は一致する。



なぜか？

公開キーのリストは、次の要素B_iから構成されている。

$$B_i = A_i s + e_i$$

ただし、条件から、sはプライベートキーで奇数、e_iは偶数である。

$b \in \{0,1\}$のエンコードされたxは、次の形をしている。

$$x = \left(\sum A_k\right) s + \left(\sum e_k\right) + b$$

ただし、kは選択されたリストの要素のみを走るものとする。

もしも、e_iがsと比べて十分に小さければ、xをsで割った余りは、

$\left(\sum e_k\right) + b$ と考えることができる。

e_iは偶数であるので、この余りは、$b = 0$の時偶数に、$b = 1$の時奇数になる。

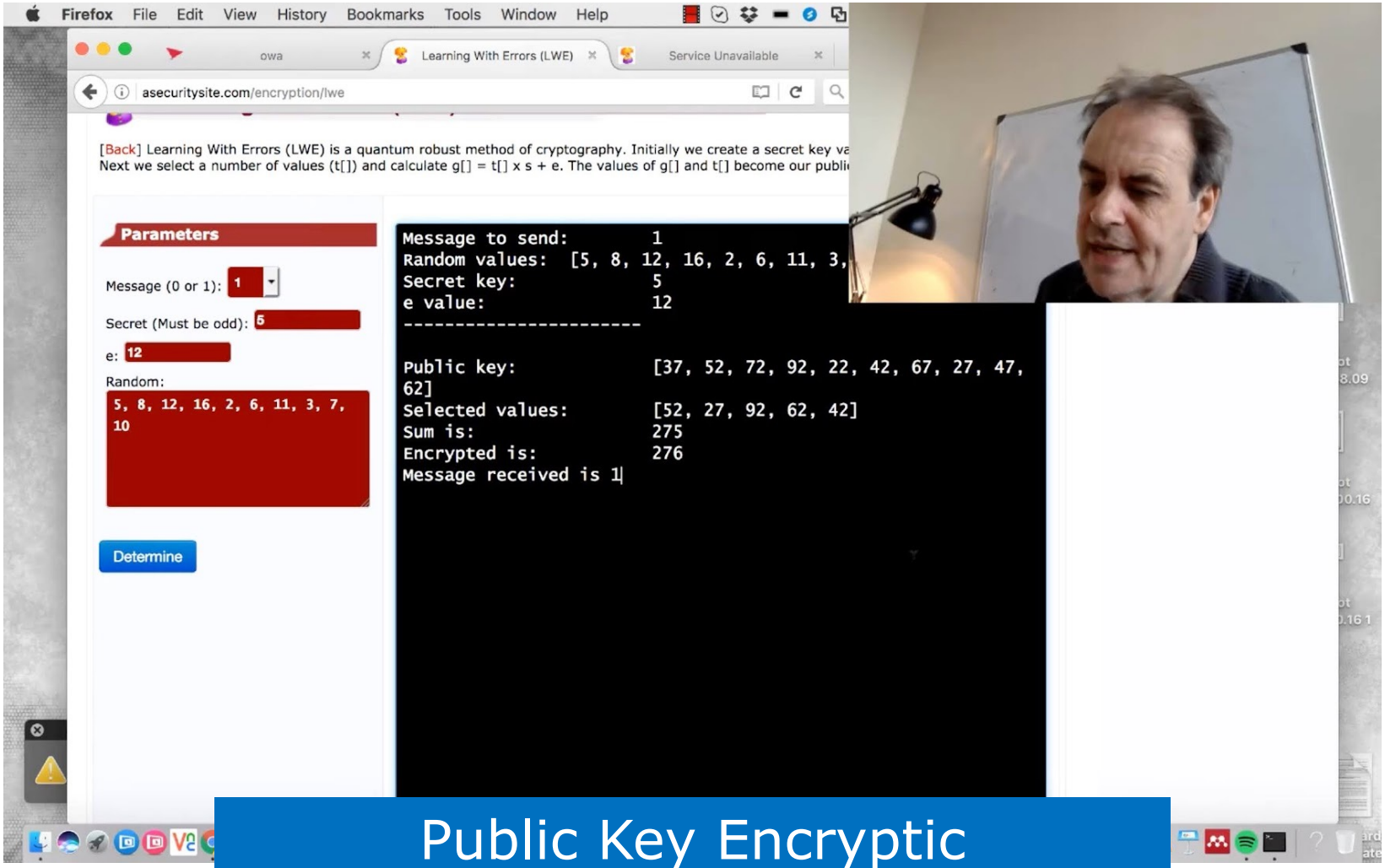
単純な例で学ぶ LWE (2)

前回のサンプルとの違い

このサンプルは、次の点で、前回のサンプルと異なっている。

- 整数$\mathcal{Z}$ではなく 整数の剰余類$\mathbb{Z}_q$を使っている。
- 公開キーのリストBが、秘密キーs と小さな数字e_iについて $B_i = A_i s + e_i$ の形の要素から構成されているのは前回と同じだが、Bに加えて、 A_i からなるリストAも公開キーに加えている。(A,B)のペアが公開キーになる。
- 公開キーの一部をサンプリングしてその和をエンコードに使うのは前回と同じだが、エンコードの仕方は、少し異なる。
- デコードの仕方も、前回とは異なっている。

このサンプルは、データの構造は単純だが、行っている処理は、本物のLWEとほとんど同じものである。



Public Key Encryptic Using Learning With Errors (LWE)

Bill Buchanan

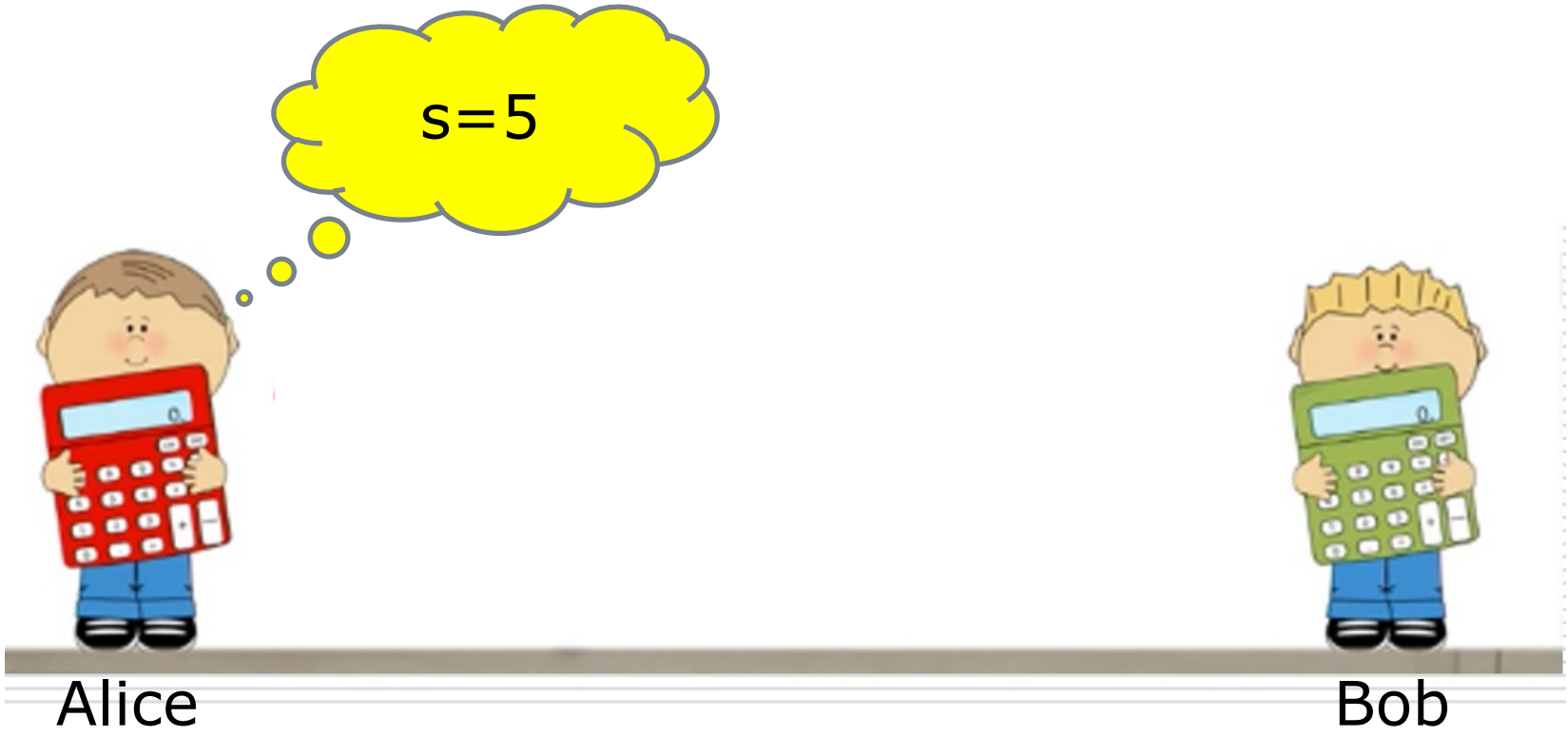
https://www.youtube.com/watch?v=sXvoX9uDr8Q

$\mathbb{Z}_q$を使った単純な LWEベースの公開キー暗号システム

Aliceが秘密キーを選ぶ

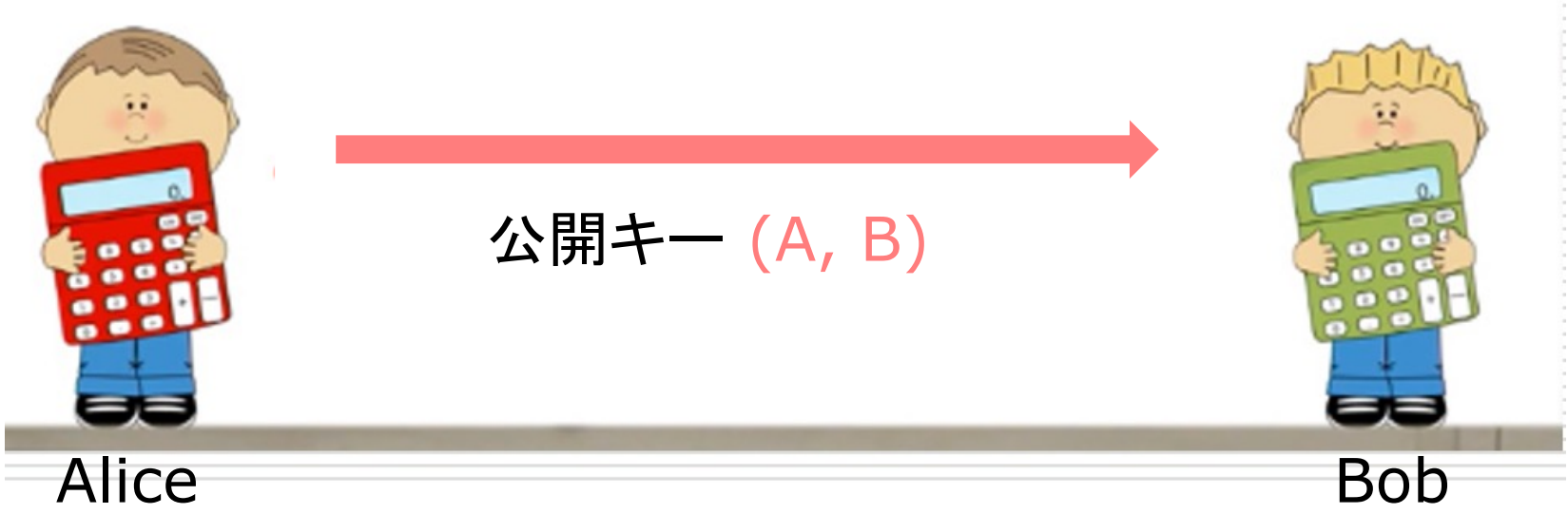
- Aliceは、秘密キー s を選ぶ。

Aliceは、秘密キーとして、s=5 を選んだとしよう。



Aliceは公開キーを作ってBobに送る

- Aliceは、まずn個の整数をランダムに選んでリストAを作る。
- また、n個の小さな整数をランダムに選んでリストeを作る。
- ただし、A, eの要素はすべて$\mathbb{Z}_q$の要素である。
- 次に、$B_i = A_i s + e_i$ を要素とするリストBをつくる。
- リスト A,Bの組 (A, B)を公開キーとしてBobに送る。



Aliceは公開キーを作ってBobに送る データ･サンプル

q=97, n=20, s=5 としよう。A, e の要素をランダムに20個えらぶ。

A= [80, 86, 19, 62, 2, 83, 25, 47, 20, 58, 45, 15, 30, 68, 4, 13, 8, 6, 42, 92]

e= [3, 3, 4, 1, 3, 3, 4, 4, 1, 4, 3, 3, 2, 2, 3, 2, 4, 4, 1, 3]

$$B_i = A_i s + e_i$$

$$B_1 = A_1 s + e_1 = 80 \times 5 + 3 = 15 \ (mod\ 97)$$
$$B_2 = A_2 s + e_2 = 86 \times 5 + 3 = 45 \ (mod\ 97)$$
$$B_3 = A_3 s + e_3 = 19 \times 5 + 4 = 2 \ \ (mod\ 97)$$

...

$$B_{20} = A_{20} s + e_{20} = 92 \times 5 + 3 = 75 \ (mod\ 97)$$

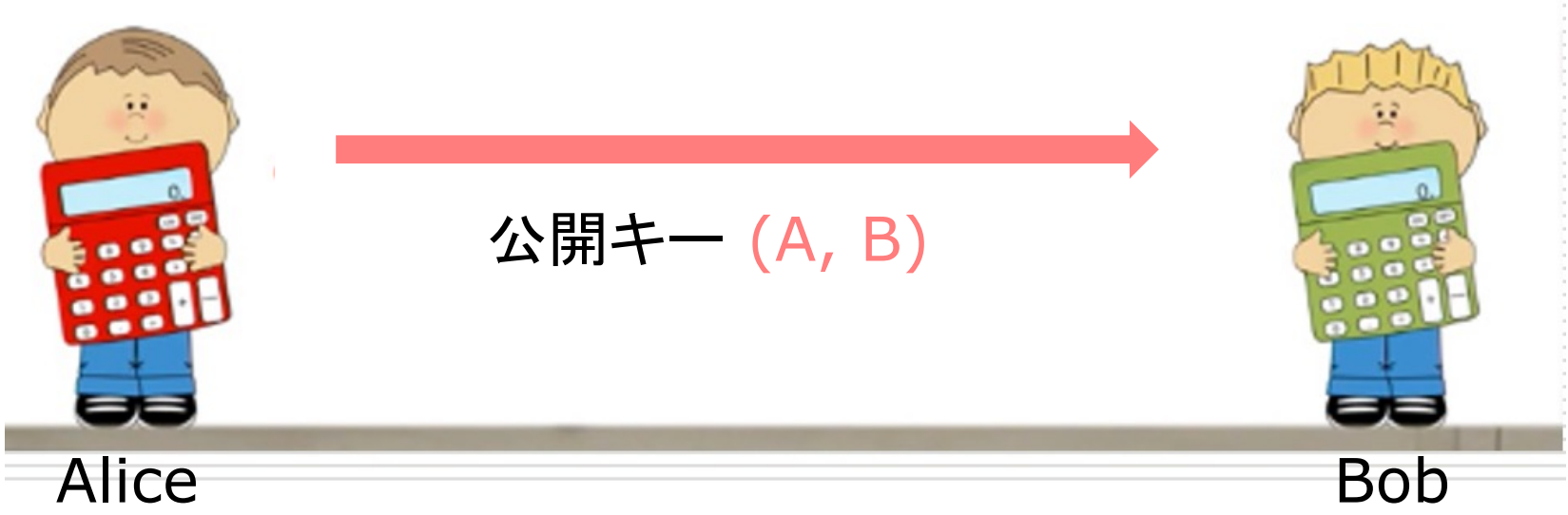
B= [15, 45, 2, 20, 13, 30, 32, 45, 4, 3, 34, 78, 55, 51, 23, 67, 44, 34, 17, 75]

Aliceは公開キーを作ってBobに送る
データ･サンプル

A= [80, 86, 19, 62, 2, 83, 25, 47, 20, 58, 45, 15, 30, 68, 4, 13, 8, 6, 42, 92]

B= [15, 45, 2, 20, 13, 30, 32, 45, 4, 3, 34, 78, 55, 51, 23, 67, 44, 34, 17, 75]

の組 (A,B)を公開キーとしてBobに送る



Bobは $b \in \{0,1\}$ をエンコードする

- 公開キーA,Bのリストから、何個かの数字をサンプリングする
- サンプリングされた数字から、次のようにu,vを定義する。

$$u = \sum A_k$$

$$v = \sum B_k + \frac{q}{2} b$$

Bobは $b \in \{0,1\}$ をエンコードする データ・サンプル

ランダムに次の添字が選ばれたとしよう。

Sampling [18, 5, 8, 13, 11]

A= [80, 86, 19, 62, 2, 83, 25, 47, 20, 58, 45, 15, 30, 68, 4, 13, 8, 6, 42, 92]

B= [15, 45, 2, 20, 13, 30, 32, 45, 4, 3, 34, 78, 55, 51, 23, 67, 44, 34, 17, 75]

Bobは $b \in \{0,1\}$ をエンコードする データ・サンプル

Sampling [18, 5, 8, 13, 11]

$$u = A_{18} + A_5 + A_8 + A_{13} + A_{11} = 6 + 2 + 47 + 30 + 45 = 34$$

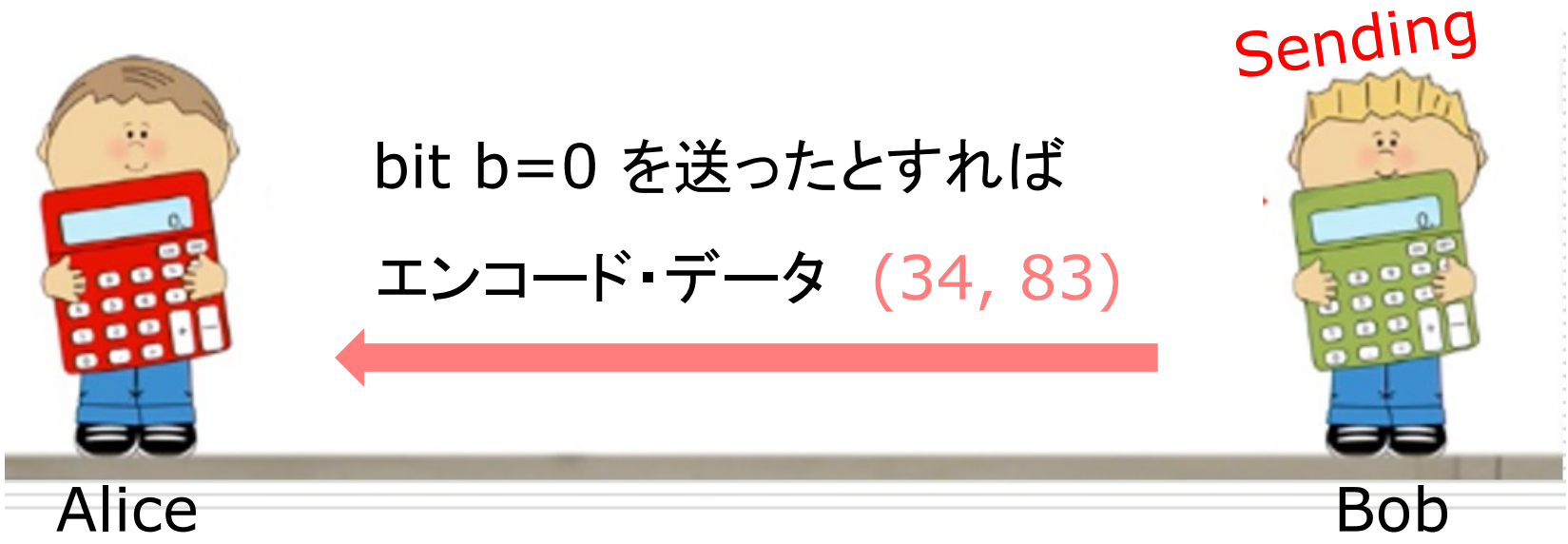
$$v_0 = B_{18} + B_5 + B_8 + B_{13} + B_{11} = 34 + 13 + 45 + 55 + 34 = 83$$

$$v[0] = 83 + \frac{97}{2} \times 0 = 83 \ (mod\ 97)$$

$$v[1] = 83 + \frac{97}{2} \times 1 = 83 + 48 = 34 \ (mod\ 97)$$

Bobはエンコードしたデータ(u,v)を Aliceに送る データ･サンプル

$$u = 6 + 2 + 47 + 30 + 45 = 34\ (mod\ 97)$$
$$v_0 = 34 + 13 + 45 + 55 + 34 = 83\ (mod\ 97)$$
$$v[0] = 83 + \frac{97}{2} \times 0 = 83\ (mod\ 97)$$
$$v[1] = 83 + \frac{97}{2} \times 1 = 83 + 48 = 34\ (mod\ 97)$$



Aliceは受け取った(u,v)をデコードする

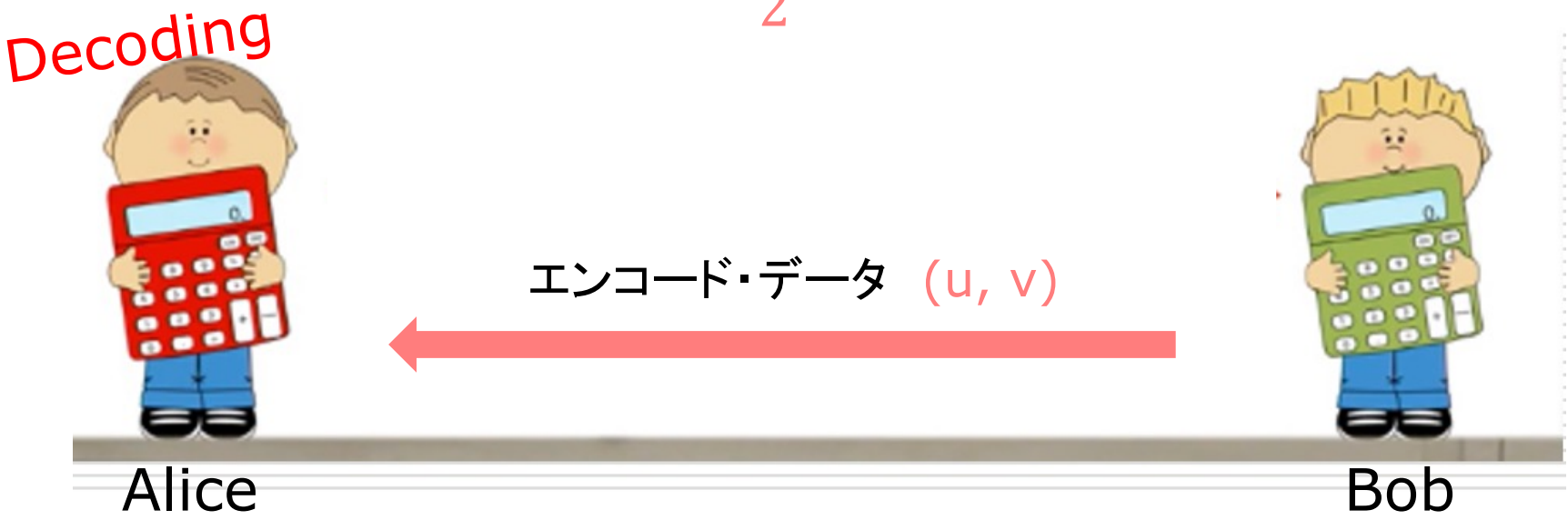
Aliceは、受け取った(u,v)から次の値を計算する

$$Dec = v - su \ (mod\ q)$$

この時

$$Dec < \frac{q}{2} \rightarrow \ b = 0$$

$$Dec \geq \frac{q}{2} \rightarrow \ b = 1$$



Decについて成り立つ式

$$Dec = v - su = \left(\sum B_k + \frac{q}{2}b\right) - s\left(\sum A_k\right)$$

$$= \left(\sum (sA_k + e_k) + \frac{q}{2}b\right) - s\left(\sum A_k\right) = \sum e_k + \frac{q}{2}b$$

$$Dec < \frac{q}{2} \Leftrightarrow \sum e_k + \frac{q}{2}b < \frac{q}{2} \Leftrightarrow \sum e_k < \frac{q}{2}(1-b)$$

$$Dec \geq \frac{q}{2} \Leftrightarrow \sum e_k + \frac{q}{2}b \geq \frac{q}{2} \Leftrightarrow \sum e_k \geq \frac{q}{2}(1-b)$$

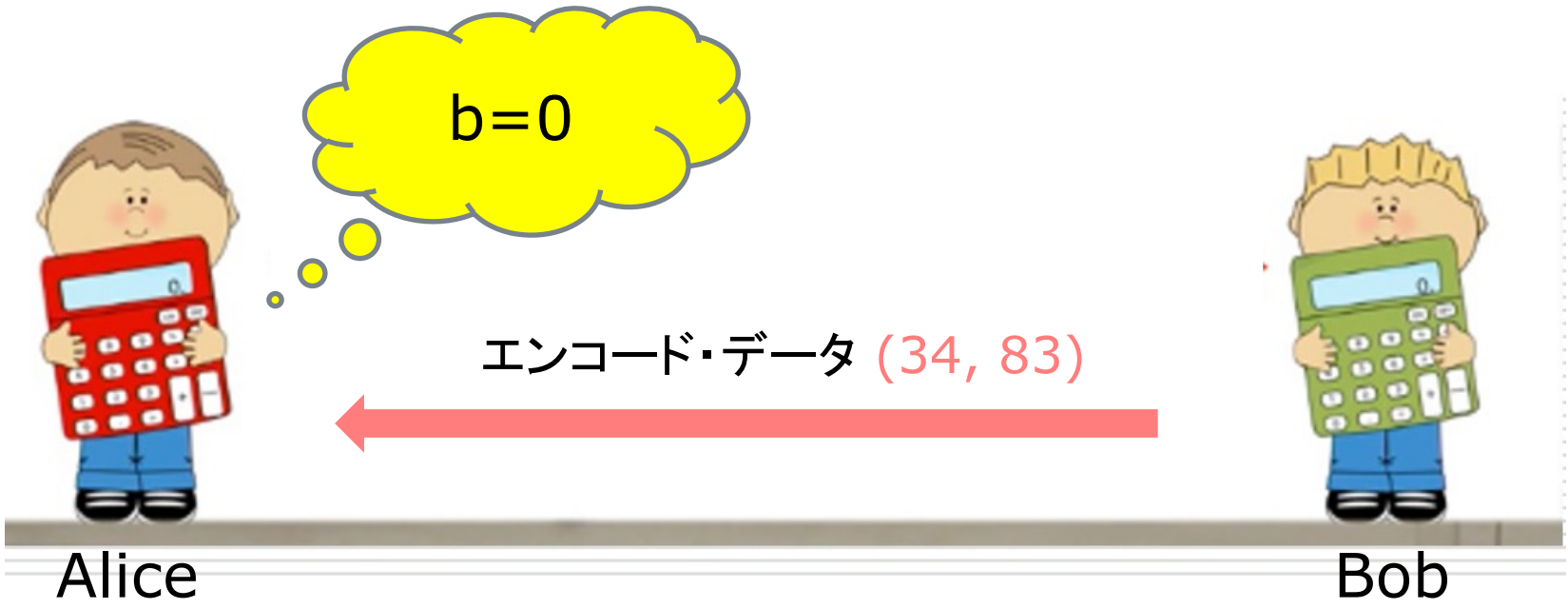
Aliceは受け取った(u,v)をデコードする
データ･サンプル

Aliceは、(34, 83)を受け取ったとする

$$Dec = v - su = 83 - 5\times 34 = -87 = 10 \ (mod\ 97)$$

この時

$10 < \frac{97}{2}$ だから、$Dec < \frac{q}{2}$ となるので $b = 0$



まとめ

- 公開キー Bの構成

$$B_i = A_i s + e_i$$

- エンコーディング

$$u = \sum A_k$$

$$v = \sum B_k + \frac{q}{2} b$$

- デコーディング

$$Dec = v - su < \frac{q}{2} \rightarrow \quad b = 0$$

$$Dec = v - su \geq \frac{q}{2} \rightarrow \quad b = 1$$

単純なサンプルからLWEへ

-- 準備編 --

単純なサンプルを拡大する
基本的な目標 -- 秘密キーの拡大

これまで見てきた単純なサンプルを、実際的な暗号システムに拡大することを考える。

二つのサンプルでは、いずれも、秘密キー s は一つの数だった。それは、少し単純すぎる。ここでは、n個の数$s_1, s_2, \ldots, s_n$の集まりとして秘密キーs を拡大しようと思う。

$s = s_1$ ではなく $s = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}$ あるいは、$s = (s_1, s_2, \ldots, s_n)$

にしようということだ。

今回のセッションでは、こうした方向での拡大を、準備する。

振り返り　単純なサンプルでの公開キーA,Bの構成

前回見たサンプルでは、公開キーA,Bは、次のように構成されていた。

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_n \end{pmatrix}$$

A_iはランダムに選ばれた数

$$B = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_n \end{pmatrix} \times s + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{pmatrix}$$

sはプライベートキー

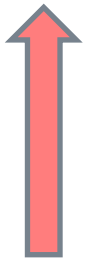
e_iはランダムに選ばれた小さな数

エラー項

Learning with Errors の名前は、この項からきている。

この式のままでは、sの拡大はできない

$$B = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_n \end{pmatrix} \times s + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{pmatrix}$$



表記法を統一しよう

前回のサンプルでは、例えば、公開キー A,B の要素を横に並べて「リスト」として表現した。これは、「行ベクトル」として、複数の要素を表現したことになる。

ただ、以前の「ラティス入門」のセッションでは、ラティスの基底をその要素を縦に並べて「列ベクトル」として表現していた。

文脈上でまぎれがなければ、どちらの表現を用いてもいいのだが、行列とベクトルの積がでてくるこのセッションでは、どちらかを基本にしたほうが、計算式の意味が分かりやすくなる。

列ベクトルを基本としよう
transpose で行ベクトルを作る

このセッションでは、列ベクトルを基本としよう。ベクトルvと言った時、vは列ベクトルであるとする。（それは、量子論でも同じである。ケット$|\phi>$は列ベクトルである。）

残念ながら、このルールは、プログラミングには向いていない。プログラムは、一行単位で横に記述されるからである。基本の列ベクトルを、行ベクトルに変換する操作 transpose を次のように導入する。ベクトル変数の右肩にTをのせる。

列ベクトル $s = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}$ に対して、

s^Tで行ベクトル$(s_1, s_2, \ldots, s_n)$を表す。

ベクトルと行列

- n個の要素を持つ列ベクトル $s = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}$ は、n行1列からなる行列とみなすことができる。

- n個の要素を持つ行ベクトル $s^T = (s_1, s_2, \ldots, s_n)$は、1行n列からなる行列とみなすことができる。

- 行列Aが、n行 m列からなる時、$A^{n \times m}$と表す。

ベクトルの内積と行列の積

- n個の要素を持つベクトルa とm個の要素を持つベクトルb に内積 <a, b>が定義されるのは、m=n の場合に限る。
この時、

$$< a, b > = a^T \times b = (a_1, a_2, \ldots, a_n) \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \sum a_i b_i$$

$$a^{1\times n} \times b^{n\times 1} = c^{1\times 1}$$

- n行m列の行列 $A^{n\times m}$ と k行j列の行列$B^{k\times j}$ の積が定義されるのは、m＝kの場合に限る。
この時、行列A,Bの積の行列Cはn行j列の行列$C^{n\times j}$になる。

$$A^{n\times m} \times B^{m\times j} = C^{n\times j}$$

ノテーションの解釈

- $x \in \{0,1\}^m$
  0,1からなるm個の要素を持つ列ベクトルx
- $s \in \mathbb{Z}_q^n$
  $\mathbb{Z}_q$からなるn個の要素を持つ列ベクトルs
- $e \in \mathbb{Z}_q^{\mathrm{m}}$
  $\mathbb{Z}_q$からなるm個の要素を持つ列ベクトルe
- $A \in \mathbb{Z}_q^{n \times m}$
  $\mathbb{Z}_q$を要素とする n行m列の行列 A

ノテーションの解釈
$b^T = s^T A + e^T$

- $b^T = s^T A + e^T$

sを行ベクトルにしたものに行列Aをかけて、それにeを行ベクトルにしたものを加えて得られるのが、行ベクトルb^T 。

$s \in \mathbb{Z}_q^n$, $A \in \mathbb{Z}_q^{n \times m}$, $e \in \mathbb{Z}_q^m$ とすれば、次のような形の計算になる。$b^T = (s_1, s_2, \dots, s_n) \times A + (e_1, e_2, \dots, e_m)$

$$(s_1, s_2, \dots, s_n) \times \underbrace{\left[A^{n \times m} \right]}_{m} = (b_1, b_2, \dots, b_m)$$

n

$$s^{1 \times n} \times A^{n \times m} = b^{1 \times m}$$

ノテーションの解釈
$u = Ax$

- $u = Ax$

行列Aに列ベクトルx を掛けたものが、列ベクトルu。
$A \in \mathbb{Z}_q^{n \times m}$, $x \in \{0,1\}^m$ とすれば、次のような形の計算になる。

$$\mathrm{u} = \underbrace{\left.\begin{matrix} A^{n \times m} \\ \mathrm{n} \end{matrix}\right\}}_{\mathrm{m}} \times \begin{pmatrix} 0/1_1 \\ 0/1_2 \\ \vdots \\ 0/1_m \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

$$u_i = \sum A_{ik}$$

$$A^{n \times m} \times x^{m \times 1} = u^{n \times 1}$$

ノテーションの解釈

$$u' = b^T x + bit \cdot \frac{q}{2}$$

- $u' = b^T x + bit \cdot \frac{q}{2}$

まず、長さmの行ベクトルb^Tに同じ長さの列ベクトルx を掛ける。それはベクトルの内積で、ただの数が返る。それに、数 $bit \cdot \frac{q}{2}$ を加えたものが、u′。次のような形の計算になる。

$$\mathrm{u}' = (b_1, b_2, \ldots, b_m) \times \begin{pmatrix} 0/1_1 \\ 0/1_2 \\ \vdots \\ 0/1_m \end{pmatrix} + bit \cdot \frac{q}{2}$$

$$(b^T)^{1\times m} \times x^{m\times 1} + \left(\mathrm{bit} \cdot \frac{\mathrm{q}}{2}\right)^{1\times 1} = (u')^{1\times 1}$$

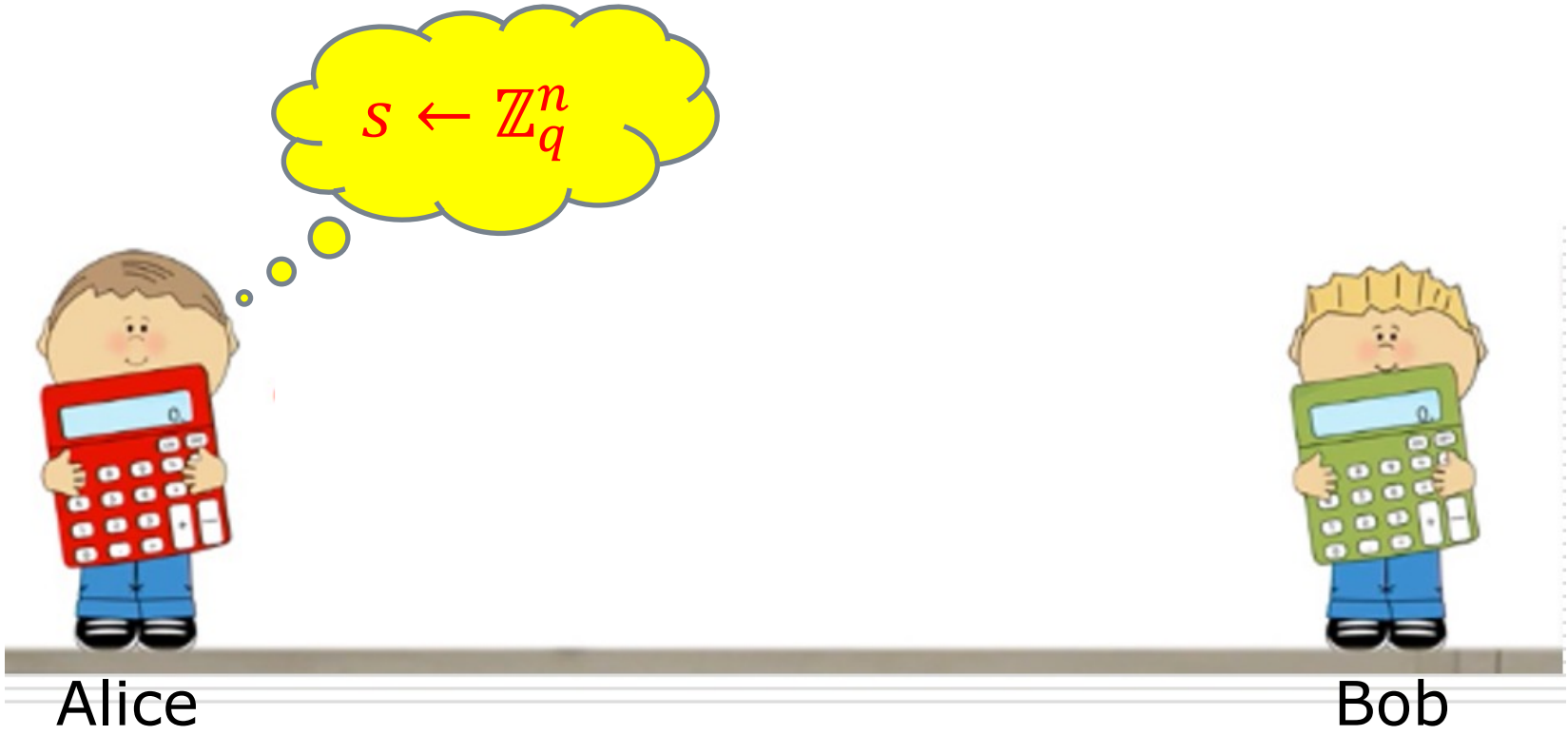
LWEの基本的なプロトコル

LWEのプロトコル

LWEの公開キー暗号には、いくつかのバリアントがあるのだが、
ここでは、もっとも基本的なプロトコルを見ておこう。

Alice：秘密キーの作成（1）

Aliceは、n個の$\mathbb{Z}_q$の要素をランダムに集めて、秘密キーsを作成する。



Alice: 秘密キーの作成 (2)

Aliceは、m個の$\mathbb{Z}_q$の小さな要素をランダムに集めて、エラー・キーeを作成する。

Alice：公開キー A の作成

Aliceは、$n \times m$個の$\mathbb{Z}_q$の要素をランダムに集めて、n行m列の行列Aを作成し、公開キーAとする。

Alice: 公開キー B の作成

Aliceは、公開キーAと秘密キーsとエラー・キーe から、次の式で公開キーBを作成する。

$$B^T = s^T A + e^T$$

$(X^T)^T = X$, $(XY)^T = Y^T X^T$ だから、この式は次の式に等しい。

$$B = A^T s + e$$

秘密キーsの拡大

$$B = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_n \end{pmatrix} \times s + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{pmatrix}$$

$$B = \underbrace{\overbrace{A^T}^{n}}_{m} \times \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_i \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_m \end{pmatrix}$$

公開キーBは、
m個の要素を持つ
列ベクトルである

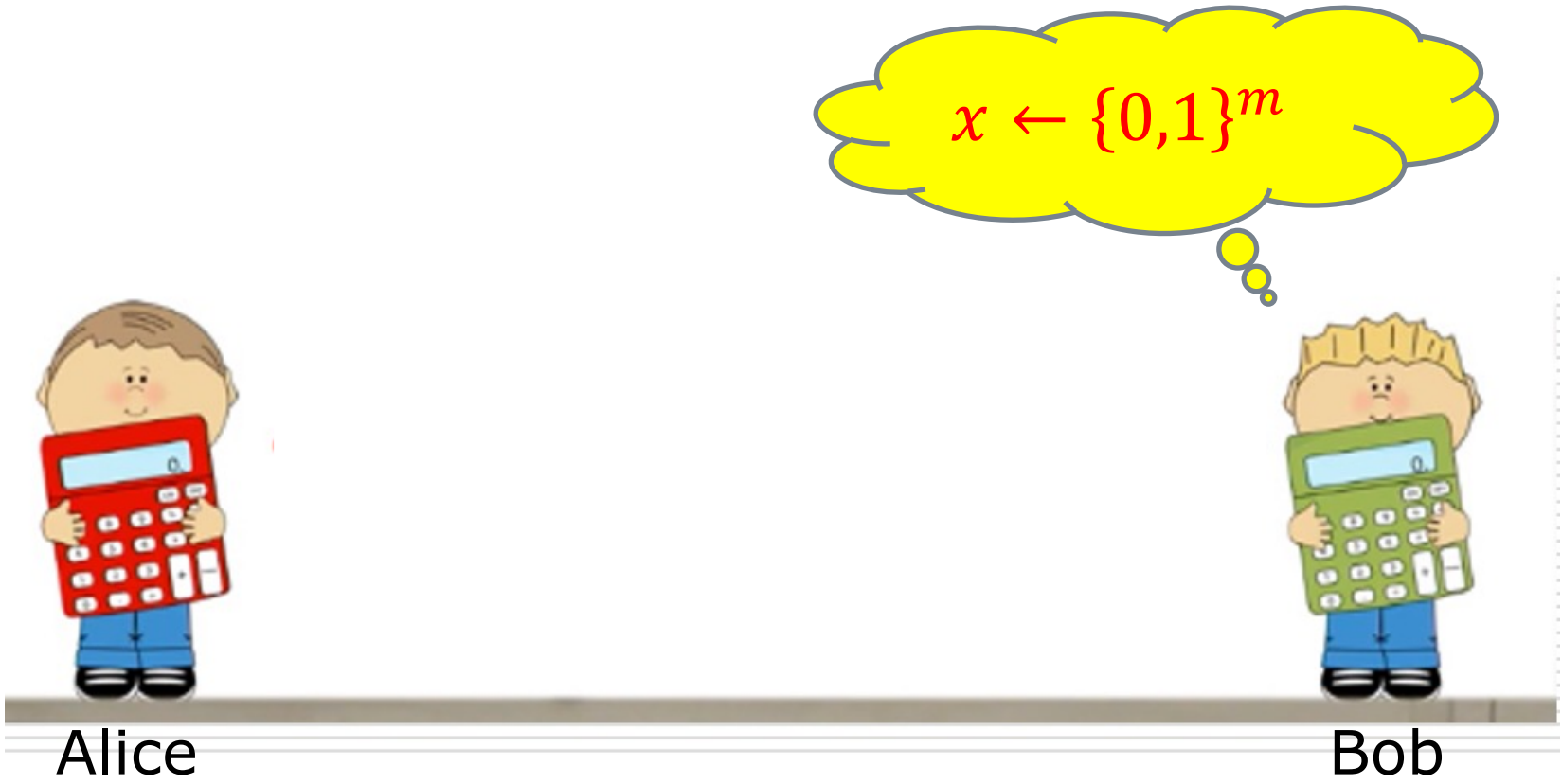
Alice: 公開キー B の作成

Aliceは、公開キーAと秘密キーsとエラー・キーe から、次の式で公開キーBを作成する。

$$B = A^T s + e$$

Bob: サンプリング 用のベクトル作成

Bobは、サンプリング用のベクトル$x \in \{0,1\}^m$を作成する。
1が立っているところがピックアップされる。



Bob：公開キーAからのサンプリング

Bobは、公開キーAとベクトルxを掛けて、サンプリング・データ u を作成する。

$$u = Ax$$

$u = Ax$

$$u = Ax$$

行列Aに列ベクトルx を掛けたものが、列ベクトルu。
$A \in \mathbb{Z}_q^{n \times m}$, $x \in \{0,1\}^m$ とすれば、次のような形の計算になる。

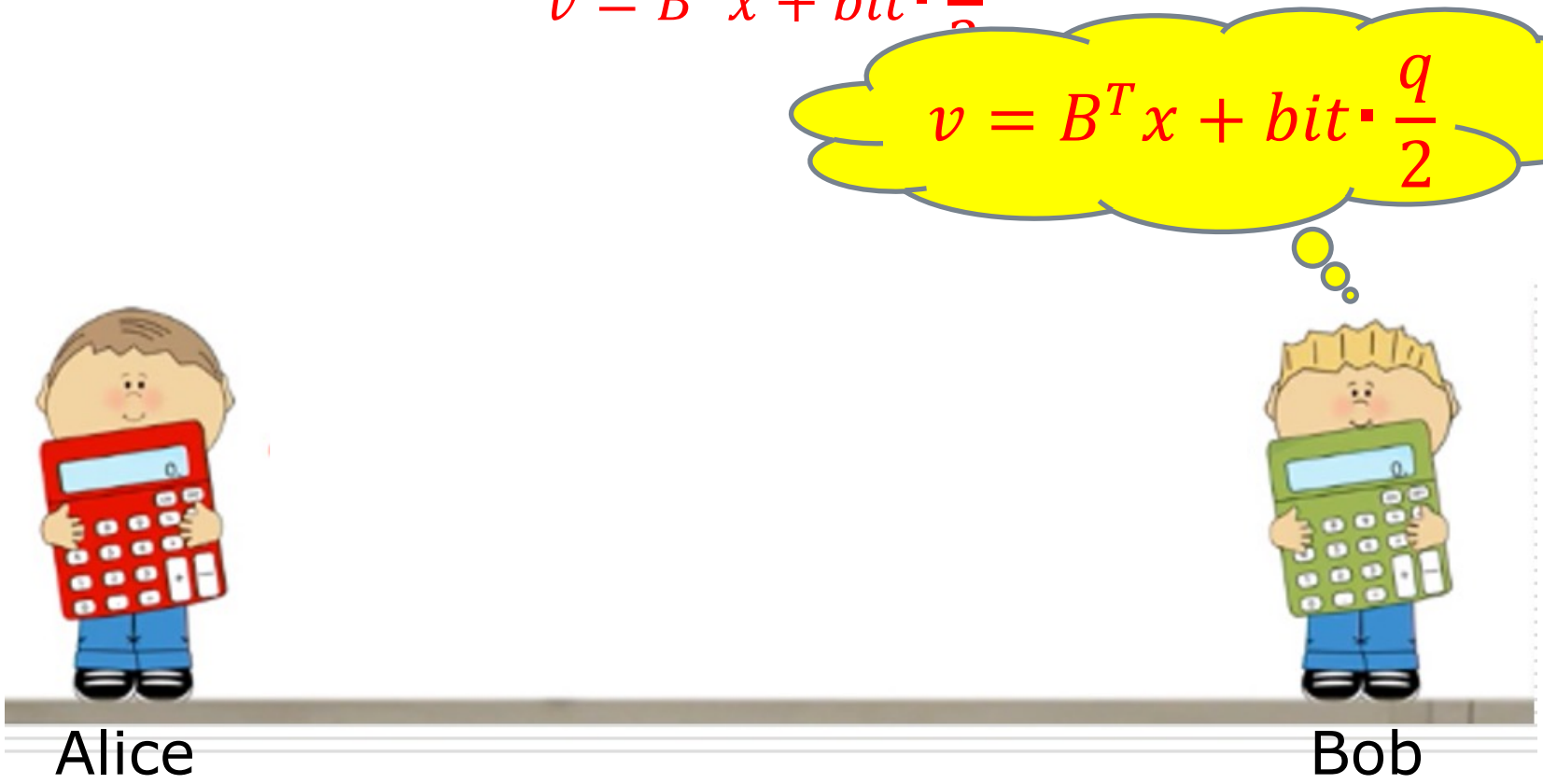
$$\mathrm{u} = \underbrace{\left.\begin{matrix} A^{n \times m} \\ \mathrm{n} \end{matrix}\right\}}_{\mathrm{m}} \times \begin{pmatrix} 0/1_1 \\ 0/1_2 \\ \vdots \\ 0/1_m \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

$$u_i = \sum A_{ik}$$

Bob: $bit \in \{0,1\}$ のエンコード

Bobは、公開キーBとサンプリング・ベクターx から、$bit \in \{0,1\}$ のエンコード・データ v を作成する。

$$v = B^T x + bit \cdot \frac{q}{2}$$



$$v = B^T x + bit \cdot \frac{q}{2}$$

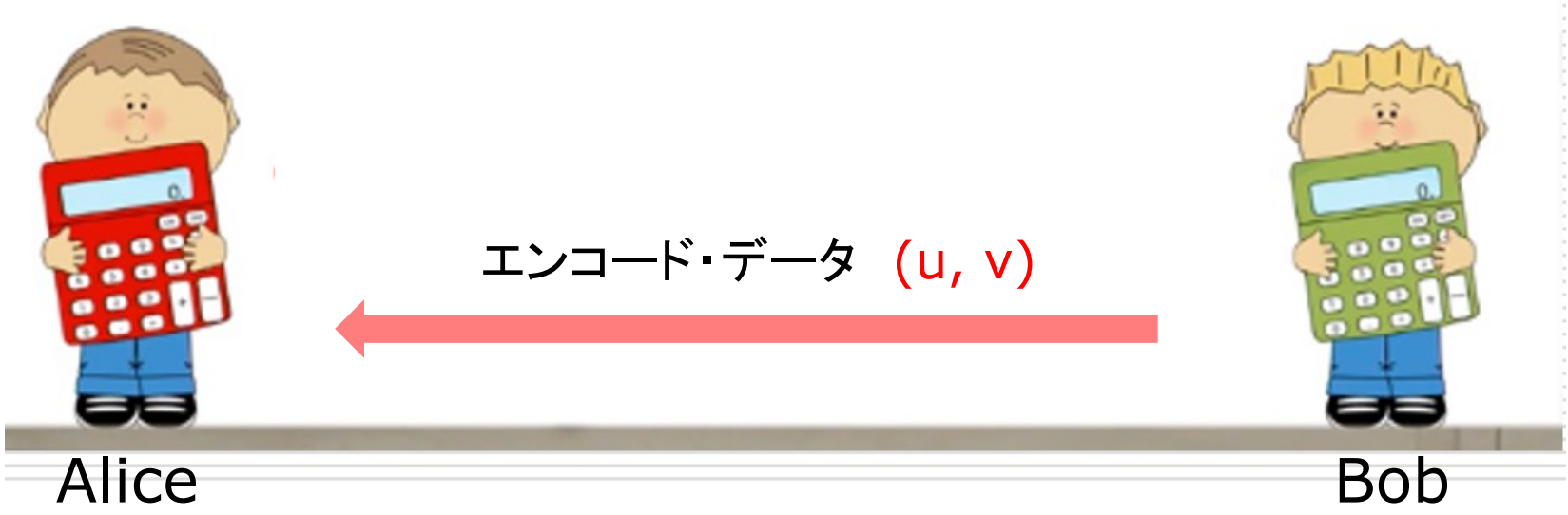
$$v = B^T x + bit \cdot \frac{q}{2}$$

まず、長さmの行ベクトルB^Tに同じ長さの列ベクトルx を掛ける。それはベクトルの内積で、ただの数が返る。それに、数 $bit \cdot \frac{q}{2}$ を加えたものが、v。次のような形の計算になる。

$$v = (b_1, b_2, \ldots, b_m) \times \begin{pmatrix} 0/1_1 \\ 0/1_2 \\ \vdots \\ 0/1_m \end{pmatrix} + bit \cdot \frac{q}{2}$$

Bob: データの送信

Bobは、エンコード・データ (u, v) をAliceに送信する。



Alice: デコード

Aliceは、受け取った(u,v)から次の値を計算する

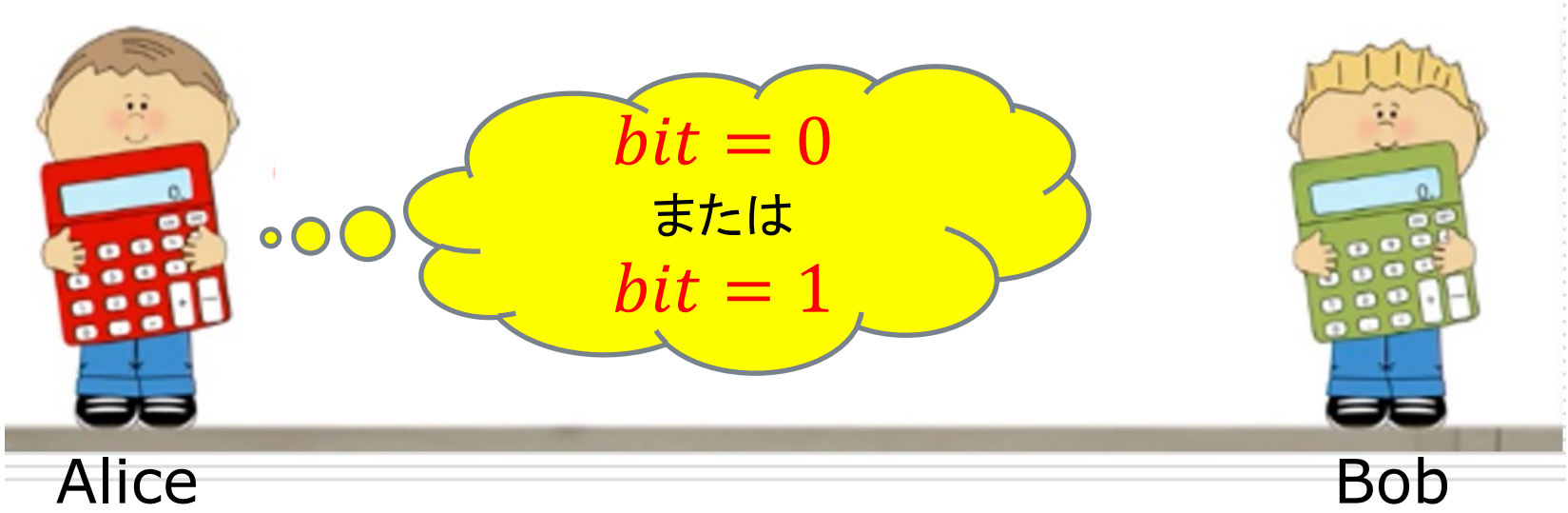
$$Dec = v - s^T u \ (mod\ q)$$

Alice: デコード

Aliceは、計算したDecから、次の式で bit をデコードする。

$$Dec < \frac{q}{2} \rightarrow \quad bit = 0$$

$$Dec \geq \frac{q}{2} \rightarrow \quad bit = 1$$



エラー項の役割

デコードのルールとエラー項の役割

前回のセッションで最後に見た 次のデコードのルールが分かりにくかったかと思う。

$$Dec < \frac{q}{2} \rightarrow \quad bit = 0$$
$$Dec \geq \frac{q}{2} \rightarrow \quad bit = 1$$

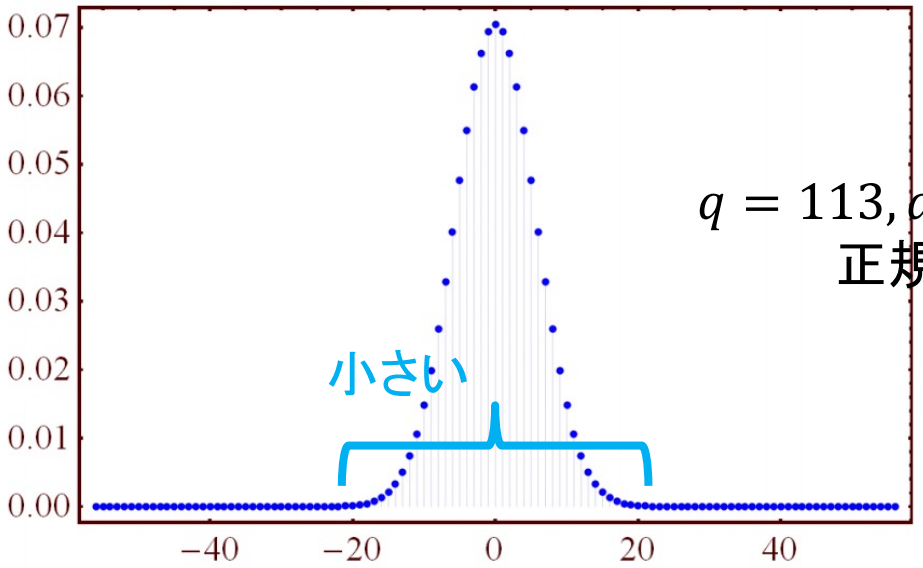
実は、これまできちんと説明してこなかったことがある。

それは、「小さな数をランダムに選ぶ」としてきたエラー項e がどのように選ばれ、LWEの中でどのような役割を果たしているかと言うことである。

エラー項は、正規分布に従うものとして選ばれる

エラー項eは、平均値 0 で標準偏差がαqである正規分布から、選ばれる。$e \in \mathbb{Z}_q^m$ であるので、正規分布と言ってもディスクリートな格子点上に選ばれる確率が定義されている。

エラー項としては、0が選ばれる確率が最も高い。平均値の0から左右に離れるとその値が選ばれる確率は急速に0に近づく。「小さな値」が選ばれると言うのは、そう言うことである。

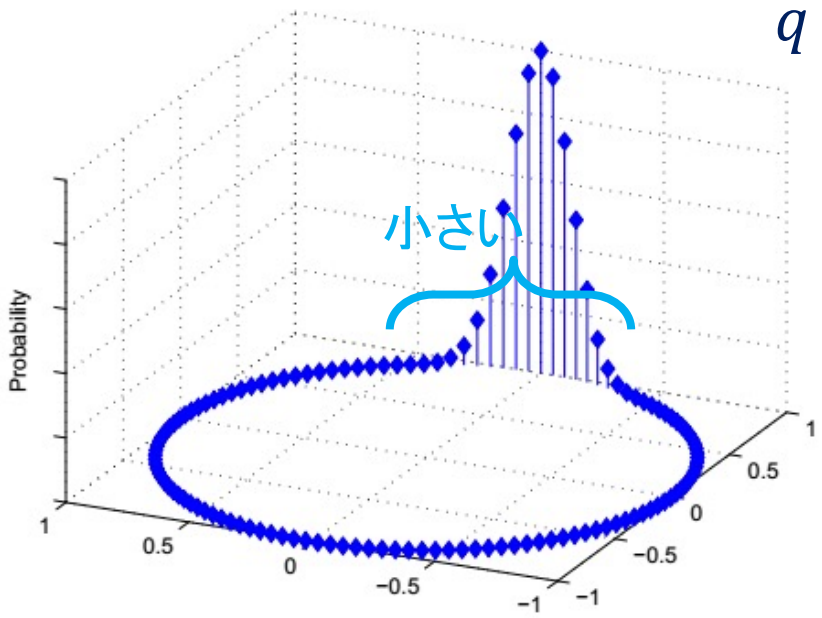


$q = 113, \alpha = 0.05$ の正規分布

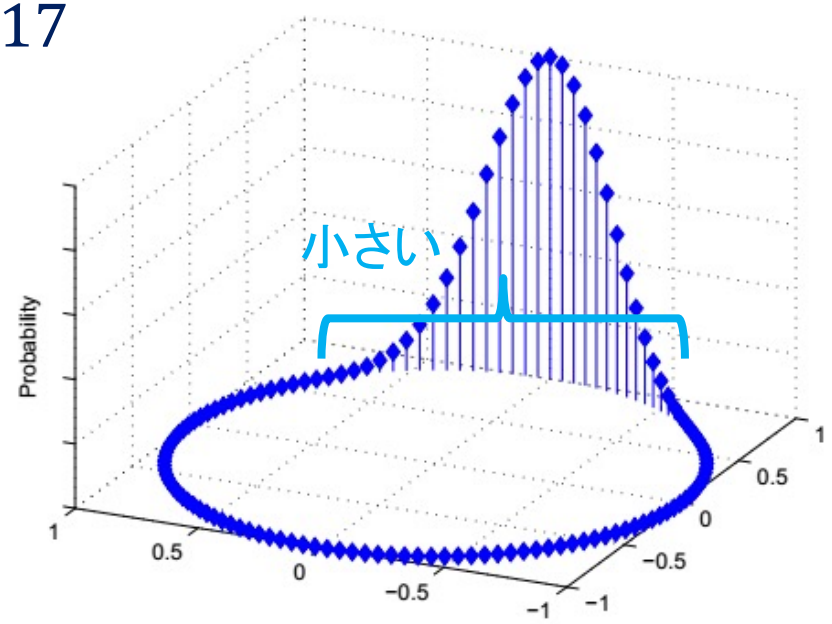
エラー項の分布

エラー項の要素は $\mathbb{Z}_q$なので巡回する円で表せる
その時のエラー項の正規分布は次のようになる

$q = 117$



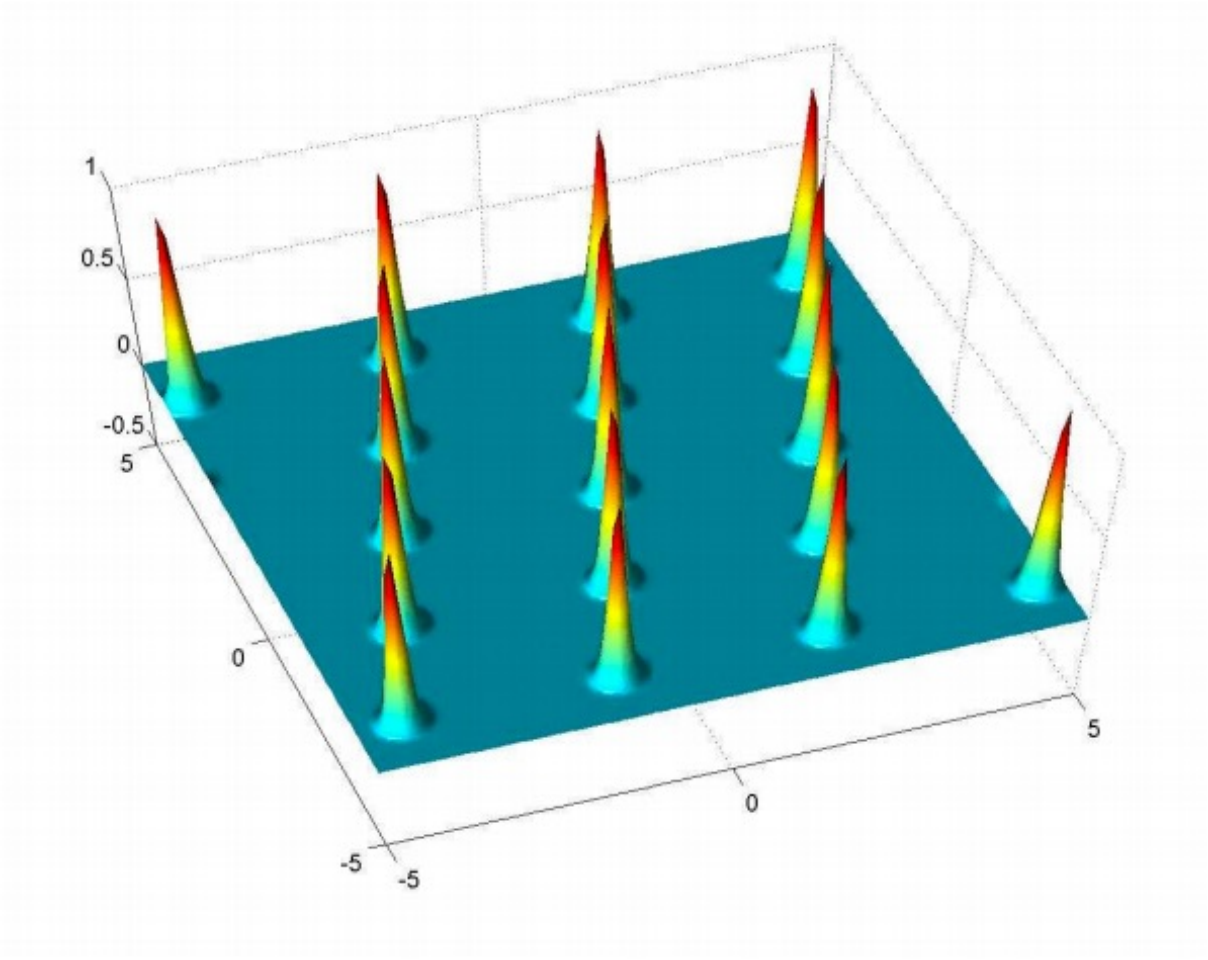
$\alpha = 0.05$



$\alpha = 0.1$

エラー項の分布
二次元の場合

二次元ラティス上の正規分布



エンコードされた(u, v)を振り返る

ベクトル$u \in \mathbb{Z}_q^n$は、公開キー$A \in \mathbb{Z}_q^{n \times m}$と
サンプリング用のベクトル$x \in \{0,1\}^m$を掛けたもの

$$u = Ax$$

スカラーvは、公開キー$B \in \mathbb{Z}_q^{m \times 1}$のトランスポーズ$B^T \in \mathbb{Z}_q^{1 \times m}$と
サンプリング用のベクトル$x \in \{0,1\}^m$を掛けたものに
数$bit \cdot \frac{q}{2}$を加えたもの

$$v = B^T x + bit \cdot \frac{q}{2}$$

$Dec = v - s^T u$ を計算する

$B = A^T s + e$ から、$B^T = s^T A + e^T$

左からxを掛けて、$B^T \mathrm{x} = s^T A \mathrm{x} + e^T \mathrm{x}$ 、

$u = Ax$ だから次の式が成り立つ。

$$B^T \mathrm{x} - s^T u = e^T \mathrm{x}$$

$v = B^T x + bit \cdot \frac{q}{2}$ だから

$$Dec = v - s^T u = \left(B^T x + bit \cdot \frac{q}{2}\right) - s^T u = e^T x + bit \cdot \frac{q}{2}$$

$$(v - s^T u) - bit \cdot \frac{q}{2} = e^T x = \sum e_k \approx 0$$

$$(v - s^T u) \approx bit \cdot \frac{q}{2}$$

eからk個の要素をサンプリングして和をとったもの

デコードのルール

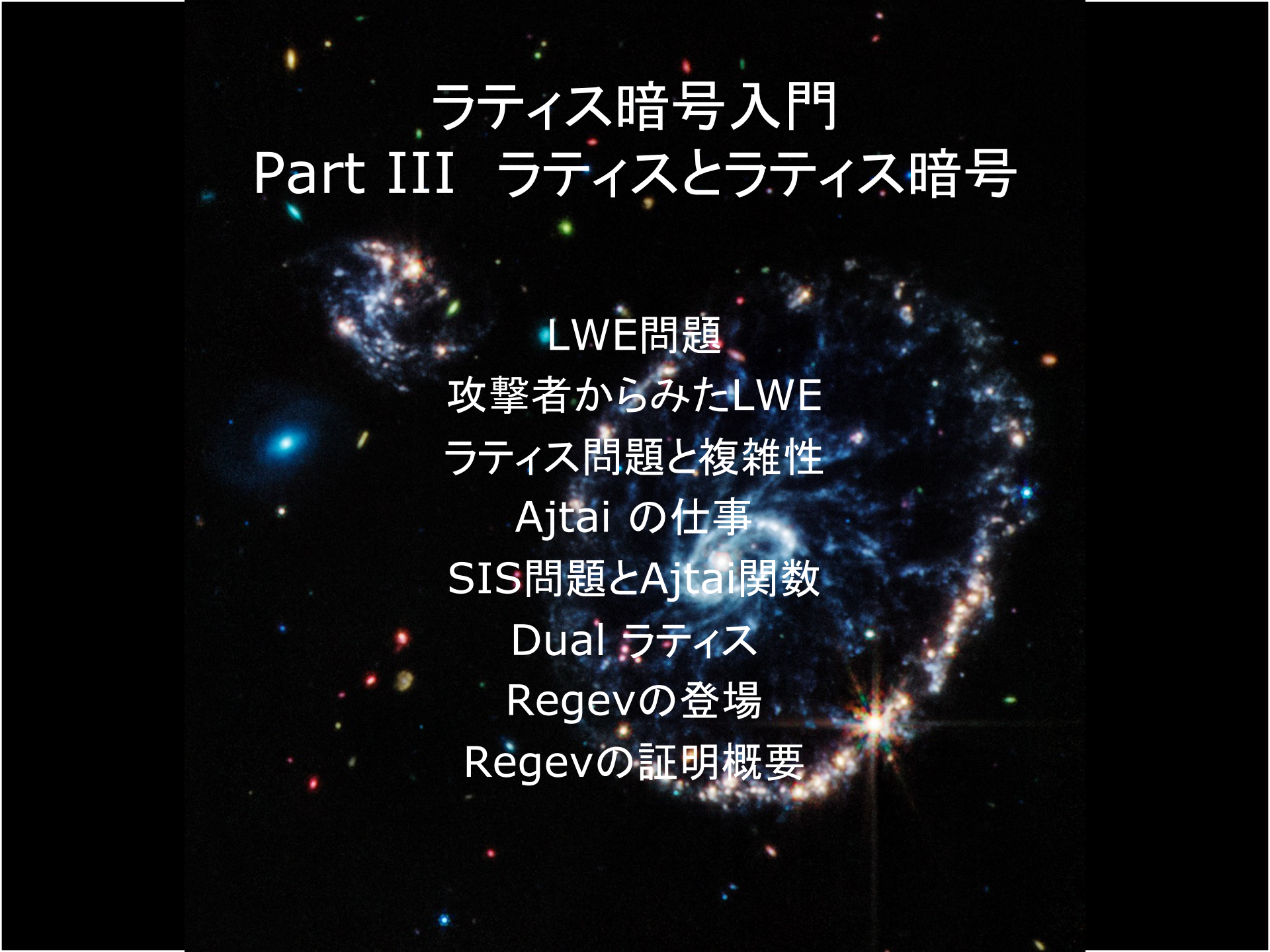
$Dec = v - s^T u$ を $v - < s, u >$と内積の形で書こう。どちらの項もスカラーであることがわかりやすい。

- $v - < s, u >$が $\left\lfloor \frac{q}{2} \right\rfloor$ より、0に近ければ、$bit = 0$
- そうでなければ、$bit = 1$


© NASA, ESA/CSA/Judy Sch

Part III

ラティスとラティス暗号



ラティス暗号入門
Part III　ラティスとラティス暗号

LWE問題

攻撃者からみたLWE

ラティス問題と複雑性

Ajtai の仕事

SIS問題とAjtai関数

Dual ラティス

Regevの登場

Regevの証明概要

LWE問題

LWE暗号の性質とLWE問題

この章から、ラティスとラティス暗号の関係について述べる。

まず、前回で概要を見たLWE暗号の性質を振り返ってみよう。

その上で、LWE問題という問題を紹介する。

LWE暗号の振り返り
Alice: 秘密キー / 公開キーの作成

- Aliceは、n個の$\mathbb{Z}_q$の要素をランダムに集めて、秘密キー s を作成する。

$$s \leftarrow \mathbb{Z}_q^n$$

- Aliceは、m個の$\mathbb{Z}_q$の小さな要素をランダムに集めて、エラー・キー e を作成する。

$$e \leftarrow \mathbb{Z}_q^m$$

- Aliceは、$n \times m$個の$\mathbb{Z}_q$の要素をランダムに集めて、n行m列の行列Aを作成し、公開キー A とする。

$$A \leftarrow \mathbb{Z}_q^{n \times m}$$

- Aliceは、公開キーAと秘密キーsとエラー・キーe から、次の式で公開キー B を作成する。

$$B = A^T s + e$$

LWE暗号の振り返り
Bob: エンコード・メッセージ (u, v) の作成と送信

- Bobは、サンプリング用のベクトル $x \in \{0,1\}^m$を作成する。

$$x \leftarrow \{0,1\}^m$$

- Bobは、公開キーAとベクトルxを掛けて、サンプリングされたデータ u を作成する。

$$u = Ax$$

- Bobは、公開キーBとサンプリング・ベクターx から、$bit \in \{0,1\}$のエンコード・データ v を作成する。 $v = B^T x + bit \cdot \frac{q}{2}$

- Bobは、エンコード・データ (u, v) をAliceに送信する。

LWE暗号の振り返り
Alice: エンコード・メッセージ (u, v) のデコード

- Aliceは、受け取った(u,v)から次の値を計算する。

$$Dec = v - s^T u$$

- Aliceは、計算したDecから、次の式で bit をデコードする。

$$Dec < \frac{q}{2} \rightarrow \quad bit = 0$$

$$Dec \geq \frac{q}{2} \rightarrow \quad bit = 1$$

メッセージ（u, v）の性質

$B = A^T s + e$ から、$B^T \mathrm{x} = s^T A \mathrm{x} + e^T \mathrm{x}$ 、
$u = Ax$ だから次の式が成り立つ。$B^T \mathrm{x} - s^T u = e^T \mathrm{x}$

$$B^T \mathrm{x} = s^T u + e^T x \implies b_k = < s, u > + e_k$$

$v = B^T x + bit \cdot \frac{q}{2}$ から

$$v - s^T u = \left(B^T x + bit \cdot \frac{q}{2}\right) - s^T u$$

$$= (s^T u + e^T \mathrm{x} + bit \cdot \frac{q}{2}) - s^T u = e^T x + bit \cdot \frac{q}{2}$$

$$= \sum e_k + bit \cdot \frac{q}{2} \approx bit \cdot \frac{q}{2}$$

$$v - s^T u = \prec s, u > \approx bit \cdot \frac{q}{2}$$

LWE問題

先の $B^T \mathrm{x} = s^T u + e^T x$ から導かれる $b_k = < s, u > + e_k$ という関係に注目しよう。b_k, e_k は、サンプリング用のベクトル $x \in \{0,1\}$ でサンプリングされたベクトルBとベクトルeの要素である。

e_kは、b_k に比べて十分小さいので、

$$< s, u > \approx b_k$$

と、近似的に表すことができる。

$u^T = \{u_1, u_2, \ldots, u_m\}, B^T = \{b_1, b_2, \ldots, b_m\}$としよう

これらのベクトルuとベクトルBの要素の値が具体的に与えられ、$< s, u > \approx b_k$の形の式が複数個与えられた時、ベクトルsの要素を求めよという問題を、LWE問題という。

次の問題は、LWE問題の例である

次の近似式が与えられた時、$s \in \mathbb{Z}_{17}^4$ を求めよという問題は、LWE問題である。

$$
\begin{aligned}
14s_1 + 15s_2 + 5s_3 + 2s_4 &\approx 8 \pmod{17} \\
13s_1 + 14s_2 + 14s_3 + 6s_4 &\approx 16 \pmod{17} \\
6s_1 + 10s_2 + 13s_3 + 1s_4 &\approx 3 \pmod{17} \\
10s_1 + 4s_2 + 12s_3 + 16s_4 &\approx 12 \pmod{17} \\
9s_1 + 5s_2 + 9s_3 + 6s_4 &\approx 9 \pmod{17} \\
3s_1 + 6s_2 + 4s_3 + 5s_4 &\approx 16 \pmod{17} \\
&\vdots \\
6s_1 + 7s_2 + 16s_3 + 2s_4 &\approx 3 \pmod{17}
\end{aligned}
$$

エラー項が存在しないとすれば、

エラー項が存在しないとすれば、先の近似式は、次のような連立方程式になって、最初の4つの式を解いて、簡単にsを求めることができる。s = (0, 13, 9, 11)。

$$
\begin{aligned}
14s_1 + 15s_2 + 5s_3 + 2s_4 &= 8 \pmod{17} \\
13s_1 + 14s_2 + 14s_3 + 6s_4 &= 16 \pmod{17} \\
6s_1 + 10s_2 + 13s_3 + 1s_4 &= 3 \pmod{17} \\
10s_1 + 4s_2 + 12s_3 + 16s_4 &= 2 \pmod{17} \\
9s_1 + 5s_2 + 9s_3 + 6s_4 &= 9 \pmod{17} \\
3s_1 + 6s_2 + 4s_3 + 5s_4 &= 16 \pmod{17} \\
&\vdots \\
6s_1 + 7s_2 + 16s_3 + 2s_4 &= 3 \pmod{17}
\end{aligned}
$$

エラー項を明示的に書けば

LWE問題とは、$< s, u > + e_k = b_k$ という式が、次のように与えられた時、sを求めよという問題である。

$$
\begin{aligned}
14s_1 + 15s_2 + 5s_3 + 2s_4 &+ e_1 = 8 \pmod{17} \\
13s_1 + 14s_2 + 14s_3 + 6s_4 &+ e_2 = 16 \pmod{17} \\
6s_1 + 10s_2 + 13s_3 + 1s_4 &+ e_3 = 3 \pmod{17} \\
10s_1 + 4s_2 + 12s_3 + 16s_4 &+ e_4 = 12 \pmod{17} \\
9s_1 + 5s_2 + 9s_3 + 6s_4 &+ e_5 = 9 \pmod{17} \\
3s_1 + 6s_2 + 4s_3 + 5s_4 &+ e_6 = 16 \pmod{17} \\
&\vdots \\
6s_1 + 7s_2 + 16s_3 + 2s_4 &+ e_m = \pmod{17}
\end{aligned}
$$

攻撃者からみたLWE

攻撃者から見たLWE

ここでは、攻撃者の視点からLWE暗号を見てみよう。

LWE暗号は、二つの公開キー (A, B)を使い、エンコードされたデータ (u, v) を送る。攻撃者は、これらのデータを知りうる。

LWE暗号では、Aliceの秘密キー sとエラーキー e、Bobのサンプリング・ベクトル x を、攻撃者は、直接には知り得ない。

ただ、次の関係がある。

$$u = Ax$$
$$B = A^T s + e$$

この時、公開キー (A,B)やuから、攻撃者がどのような情報を引き出せるか考えてみよう。

攻撃者が 公開キーAから知りうる情報

攻撃者は、公開キーAと$u = Ax$ から、Bobのみが知るxについての情報を知ることができるだろうか？

$A \in \mathbb{Z}_q^{n\times m}, x \in \{0,1\}^{m\times 1}$ より、$u \in \mathbb{Z}_q^{n\times 1}$ すなわち、$u = Ax$ はn個の要素を持つ列ベクトルである、$u^T = \{u_1, u_2, \dots, u_n\}$とすると、

$$u_i = \sum A_{ik}\, x_k$$

u_iは、行列Aのi行目のm個の要素の部分集合の和である。一行について、部分集合は 2^m個のあるので、行列のn行全体では、$2^m \times 2^m \times \cdots \times 2^m = 2^{n\times n}$個の可能な組み合わせがある。攻撃者が、uの値から、xを推定することは、難しい。

攻撃者が 公開キーBから知りうる情報

攻撃者は、公開キーA, B と、$B = A^T s + e$ から、秘密キーsとエラーキーeについての情報を知りうるであろうか？

$B = A^T s + e$ から、$B^T = s^T A + e^T$ 。このBは公開されている。

先にLWE問題を、次のように定義した。

$$b_1 = < s, a_1 > + e_1$$
$$b_2 = < s, a_2 > + e_2$$
$$\vdots$$
$$\vdots$$

が与えられた時、s を求めよという問題である。

これは先の$B^T = s^T A + e^T$を、要素ごとに展開したものだ。

LWE問題の例

LWE問題とは、$< s, u > + e_k = b_k$ という式が、次のように与えられた時、sを求めよという問題である。

$$
\begin{aligned}
14s_1 + 15s_2 + 5s_3 + 2s_4 &+ e_1 = 8 \pmod{17} \\
13s_1 + 14s_2 + 14s_3 + 6s_4 &+ e_2 = 16 \pmod{17} \\
6s_1 + 10s_2 + 13s_3 + 1s_4 &+ e_3 = 3 \pmod{17} \\
10s_1 + 4s_2 + 12s_3 + 16s_4 &+ e_4 = 12 \pmod{17} \\
9s_1 + 5s_2 + 9s_3 + 6s_4 &+ e_5 = 9 \pmod{17} \\
3s_1 + 6s_2 + 4s_3 + 5s_4 &+ e_6 = 16 \pmod{17} \\
&\vdots \\
6s_1 + 7s_2 + 16s_3 + 2s_4 &+ e_m = \pmod{17}
\end{aligned}
$$

LWE問題のラティス問題への還元

LWE暗号の公開キー(A, B)を$(A, B^T = s^T A + e^T)$と表すことができる。もし、$B^T = s^T A + e^T$についての LWE問題が解かれるなら、LWE暗号の秘密キーは破られることになる。

LWE暗号を破ることの困難さは、まだ、きちんと説明されていないのだが、LWE問題を解くことの困難さに依拠している。

LWE問題の困難さの証明は、すでにその困難さが知られている他の問題にLWE問題を還元することで行われる。

その問題とは、いくつかの「ラティス問題」である。

そのことを、これから見ていこう。

ラティス問題と複雑性

ラティス問題

ラティスに関して、以前に、次の二つの問題を紹介した。

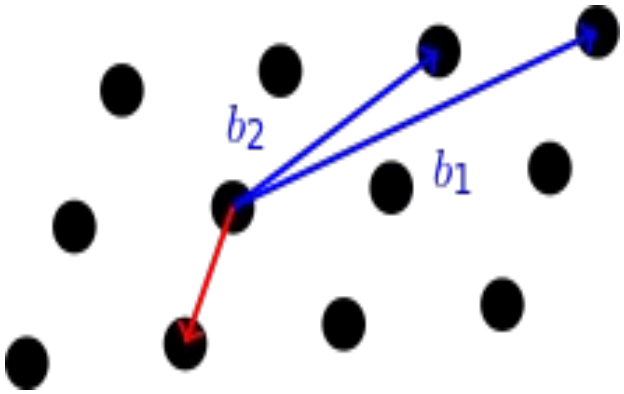
- Shortest vector problem (SVP)
- Closest vector problem (CVP)

ここでは、さらに、次の問題を紹介する。

- Shortest independent vectors problem (SIVP)
- bounded distance decoding problem (BDD)
- GapSVP_{γ}

これらの問題は、次元が高くなると一般には解くのが難しい。ここではその複雑性を概観する。その難しさが、ラティス暗号の基礎になっている。

Shortest vector problem (SVP)

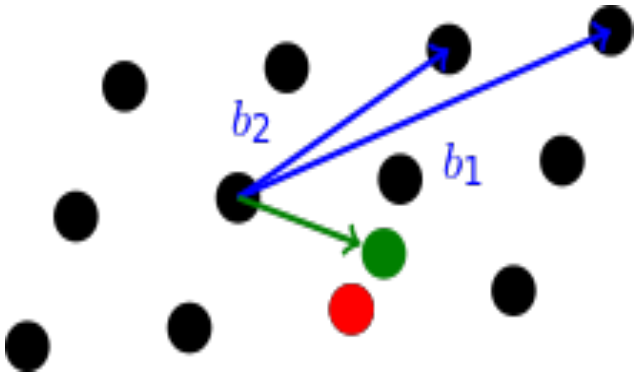


左の図のように基底b_1, b_2で張られるラティスがあるとする。

このラティス上の点で、互いに一番近い二点を求めよ。

（答は、赤い線で結ばれた二点である）

Closest vector problem (CVP)



左の図のように基底b_1, b_2で張られるラティスがあるとする。

この平面上に、ラティスに属さない点を一つとる。（緑の点）

ラティス上の点で、この緑の点に一番近い点を求めよ。

（答は、赤い点である）

Shortest independent vectors problem (SIVP)

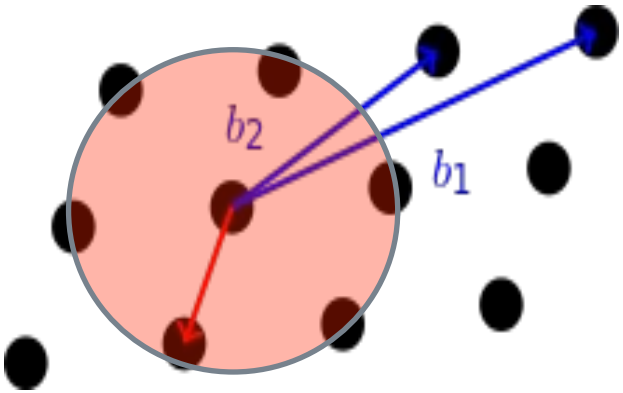
$B = \{b_1, b_2. \ldots, b_n\}$を基底とするラティス Lが与えられた時、

$$\max\|v_i\| \leq max\|b_i\|$$

を満たす、独立な $v_1, v_2, \ldots, v_n$ を見つける問題。

最小半径 $\lambda_1(L)$

最小半径 $\lambda_1(L)$ で、ラティスL の、ゼロでない最小のベクトルの長さを表す。



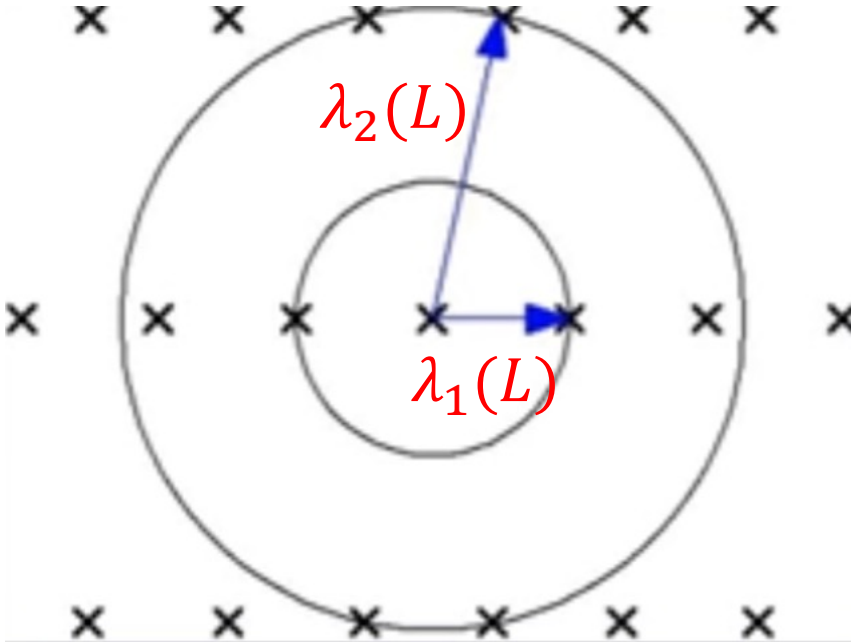
$\lambda_1(L)$

基底b_1, b_2で張られるラティス L があるとする。

このラティスL 上の点で、互いに一番近い二点を求めよ。**SVP**（答は、赤い線で結ばれた二点である）

最小半径 $\lambda_k(L)$の系列

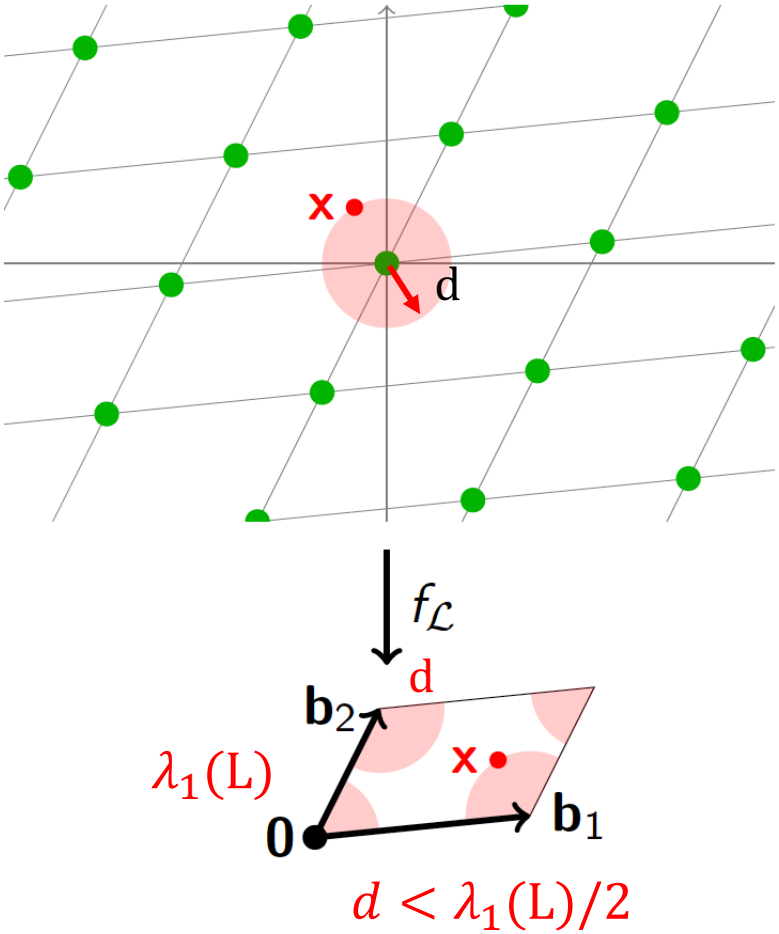
もっと一般的に、$\lambda_k(L)$ で、ラティスL の、ゼロでないk個の独立したベクトルを含む、球の最小の半径を示す。



Bounded Distance Decoding problem (BDD)

ある距離のパラメーター d に対して、ラティス L で最大距離でdの範囲に点x があるとする。この時、x に最も近いラティス・ベクトルを求めよという問題。

$d < \lambda_1(\mathrm{L})/2$ であれば、正しい答えは、ユニークに定まる。



$GapSVP_\gamma$

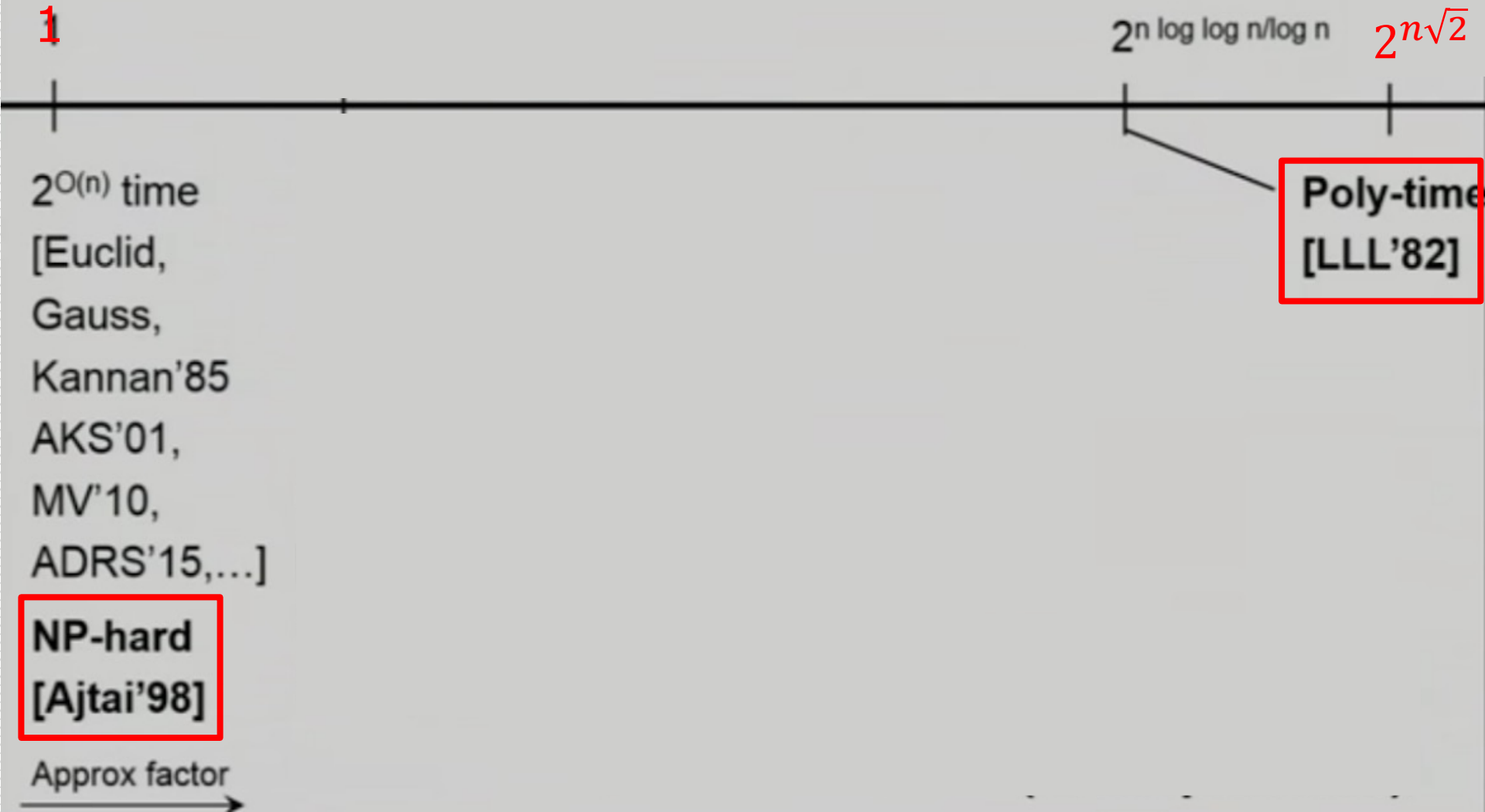
ラティス Lが与えられた時、$\lambda_1(L)$を、乗数因数 γ で近似せよという問題。

この問題は、γが小さい時には NP-困難であることが知られている。

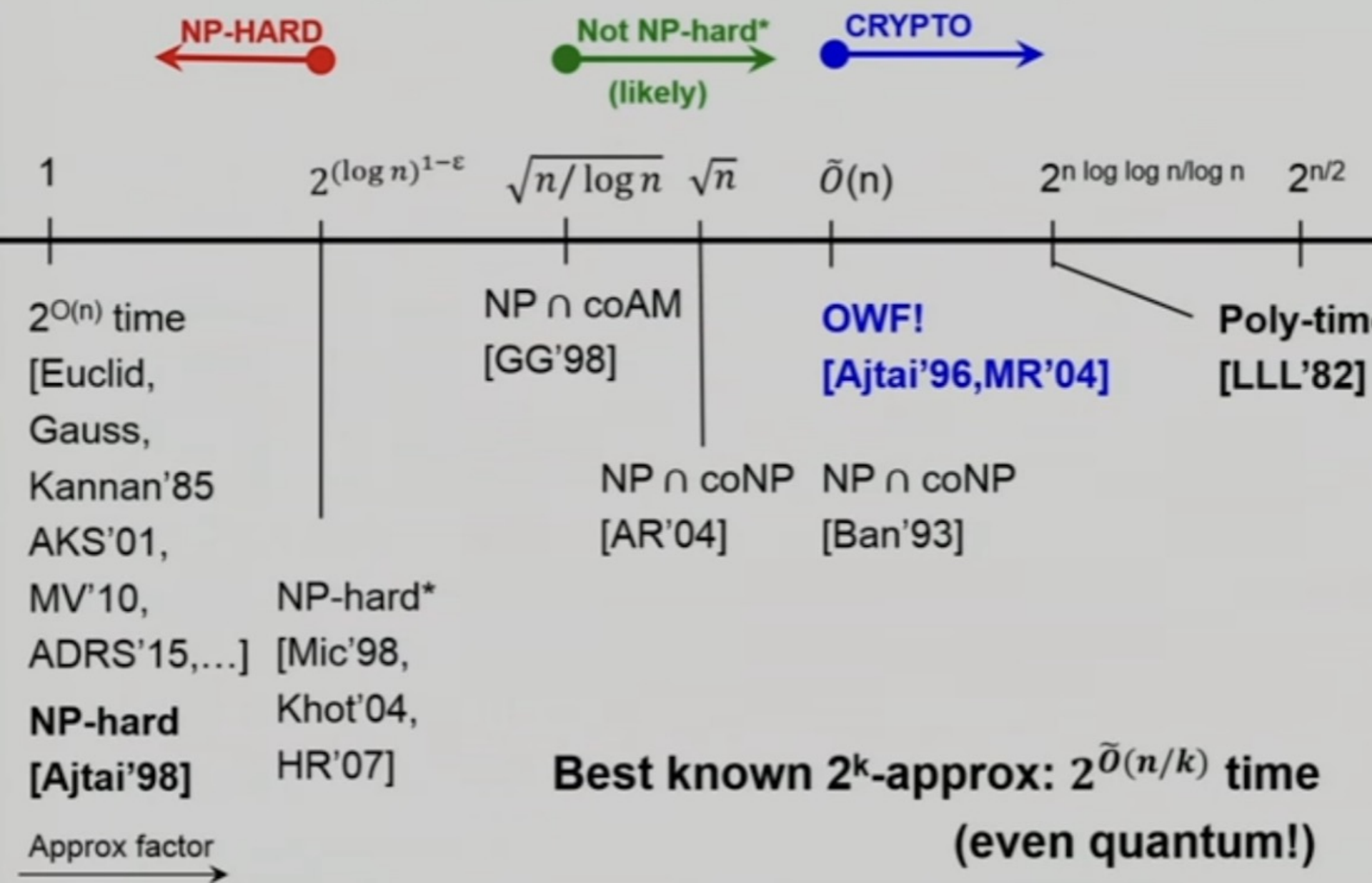
γが大きい時には、たとえば、$\gamma = 2^{O(n)}$ の時には、多項式時間で解ける。

SVP: アルゴリズムと複雑性クラス

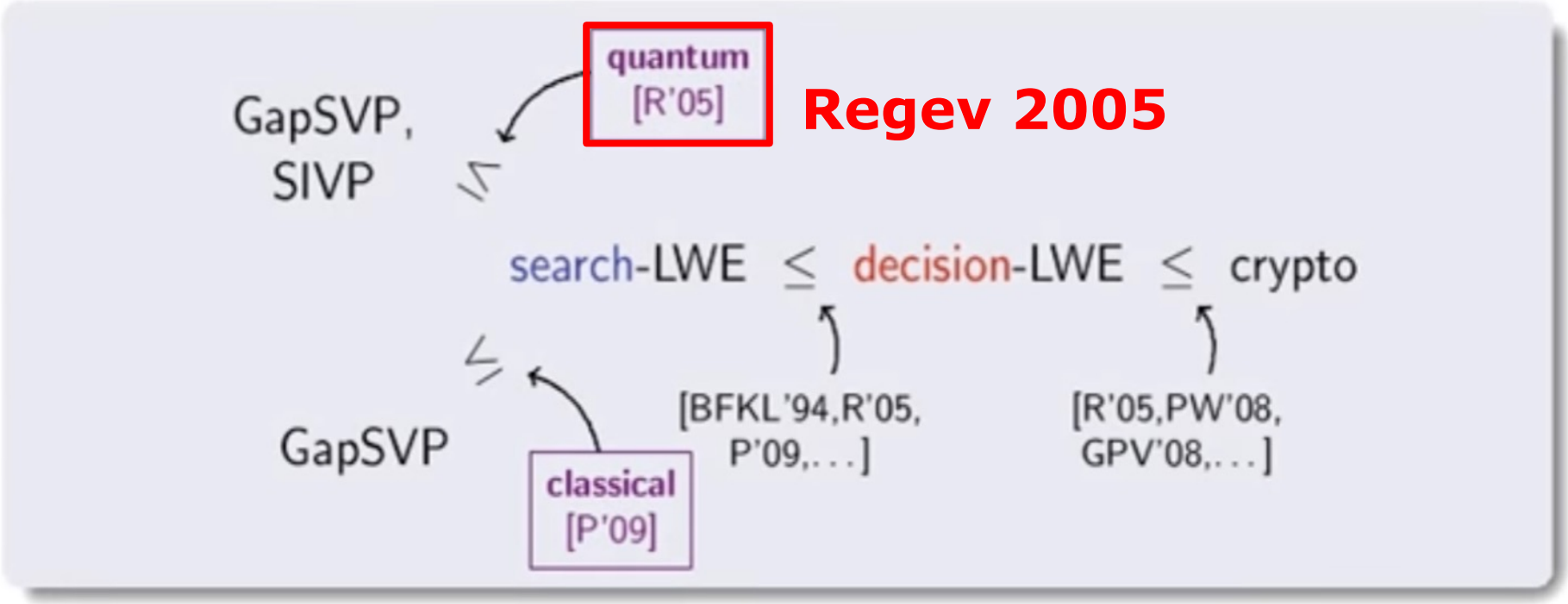
Vinod Vaikuntanathan, "The Mathematics of Lattices"
Simons Cryptography Bootcamp
https://www.youtube.com/watch?v=LlPXfy6bKIY



SVP: アルゴリズムと複雑性クラス



Overview of LWE Hardness



Chris Peikert, Learning with Errors
Winter School on Lattice-Based Cryptography and Applications
https://web.eecs.umich.edu/~cpeikert/pubs/slides-barilan5.pdf

Ajtai の仕事

Ajtai の仕事

ラティス問題を暗号に応用する動きでAjitaiの果たした役割は、大きかった。

- ラティス問題 SVPの複雑性の解明
- $\mathbb{Z}^n$ラティスのランダムなクラス
- SIS問題を利用した一方向関数の定義とHashへの応用
- Worst-Case/Average-Case Equivalence
- SVCに基づく公開キー暗号システムの提案

Miklos Ajtai

ラティス問題 SVPの複雑性の解明

「我々は、L2ノルムを持つラティスにおける最短ベクトル問題がランダム還元に対してNP-困難であることを示す。

さらに、ランダム還元では、 (L2ノルムに関して) $1 + 2^{-n^{\epsilon}}$ の係数以下で最短の非ゼロベクトルよりも長いベクトルを見つけることも NP-困難である。この絶対定数εが $\varepsilon > 0$ であることも示している。

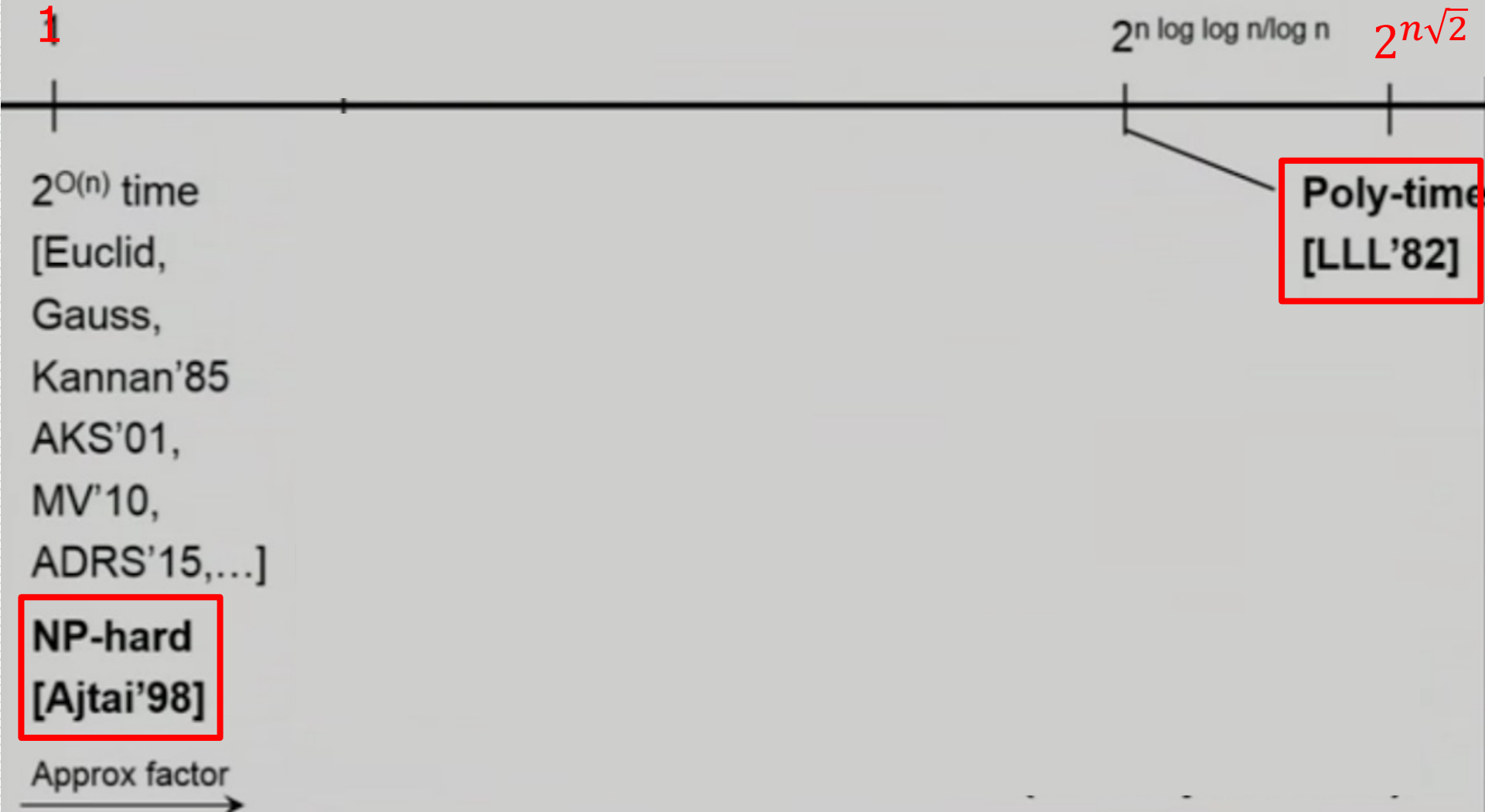
対応するランダム還元のもとでの決定問題は、NP完全である。」

Ajitai, The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions,

https://dl.acm.org/doi/pdf/10.1145/276698.276705

SVP: アルゴリズムと複雑性クラス

Vinod Vaikuntanathan, "The Mathematics of Lattices"
Simons Cryptography Bootcamp
https://www.youtube.com/watch?v=LlPXfy6bKIY



$\mathbb{Z}^n$のラティスのランダムなクラス

「その要素とともに、短いベクトルを一緒に生成する ことができる $\mathbb{Z}^n$のラティスのランダムなクラスを与える。

もし、少なくとも1/2の確率でランダムなラティスで短いベクトルを見つける確率的多項式時間アルゴリズムが存在するとすれば、$\mathbb{Z}^n$の全てのラティスに対して 次の三つのラティス問題を、指数関数的に1 に近い確率で解く確率的多項式時間アルゴリズムが存在する。

三つの問題とは、… 」

M. Ajtai, Generating Hard Instances of Lattice Problems, Proceedings 28th Annual ACM Symposium on Theory of Computing 1996

https://dl.acm.org/doi/pdf/10.1145/237814.237838

SIS問題を利用した一方向関数の定義とHashへの応用

「近年、Ajtaiは、ラティスにおけるいくつかのよく知られている近似問題の難しさと同等のセキュリティを持つ一方向関数の構築を与えた。

我々は、本質的に同じ構築を使ってして、衝突のないハッシュ関数を得ることができることを示す。」

Oded Goldreich, Shafi Goldwasser, and Shai Halevi, Collision-Free Hashing from Lattice Problems,
Cryptology ePrint Archive, Paper 1996
https://eprint.iacr.org/1996/009

SVCに基づく公開キー暗号システムの提案

「次のラティス問題を考える。「長さが最大で $n^c \| v \|$ である他のベクトルが、vに平行であるという意味で最短ベクトルvは一意であるとして、n次元ラティスL内の非ゼロの最短ベクトルを見つけよ」

我々は、このラティス問題の最悪のケースが、多項式時間で解決されない限り、安全な確率的公開鍵暗号システムを提示する。」

M. Ajtai, C. Dwork, A Public Key Cryptosystem with Worst-Case/Average-CasEequivalence. Proc. 29-th Annual ACM Symp. on the Theory of Computing, 1997

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.45.7341&rep=rep1&type=pdf

SIS問題とAjtai関数

SIS問題
Small Integer Solution

SIS問題は、次のような問題である。

$a_i \in \mathbb{Z}_q^n$ であるn個の要素からなるランダムなベクトル$\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_m}$がm個与えられているとする。

この時、次の式が成り立つような $z_i \in \mathbb{Z}$ を求めよ

$$z_1\boldsymbol{a_1} + z_2\boldsymbol{a_2} + \cdots + z_m\boldsymbol{a_m} = \boldsymbol{0}$$

ただし、$z_i \in \{-1, 0, +1\}$とする。

$$z_1 \begin{pmatrix} \vdots \\ \boldsymbol{a_1} \\ \vdots \end{pmatrix} + z_1 \begin{pmatrix} \vdots \\ \boldsymbol{a_2} \\ \vdots \end{pmatrix} + \cdots + z_m \begin{pmatrix} \vdots \\ \boldsymbol{a_m} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \boldsymbol{0} \\ \vdots \end{pmatrix}$$

最後の、$z_i \in \{-1, 0, +1\}$という条件がなかったとすれば、先の式を満たす 整数z_i を求めるのは易しい。

SIS問題
行列での表現

先のベクトル $\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_m} \in \mathbb{Z}_q^n$ を並べて、次のように行列Aを作る。

$$A = \underbrace{\left[\begin{pmatrix}\vdots\\ \boldsymbol{a_1}\\ \vdots\end{pmatrix}\begin{pmatrix}\vdots\\ \boldsymbol{a_2}\\ \vdots\end{pmatrix}\ldots\begin{pmatrix}\vdots\\ \boldsymbol{a_m}\\ \vdots\end{pmatrix}\right]}_{m} \Big\} n$$

$\mathrm{A} \in \mathbb{Z}_q^{n \times m}$

この時、SIS問題は、次のように表現できる。

ランダムな行列 $\mathrm{A} \in \mathbb{Z}_q^{n \times m}$ に対して、$z \in \{-1, 0, +1\}^m$で、

$$Az = 0$$

を満たすzを求めよ。

ランダムな行列Aで定義される Ajtaiの関数 f_A

先に構成したランダムな行列 $A \in \mathbb{Z}_q^{n \times m}$ に対して、$x \in \{0,1\}^m$ 上で定義され、$y \in \mathbb{Z}_q^n$に値を取る 関数f_Aを次のように定義する。

$$f_A(x) = Ax = y$$

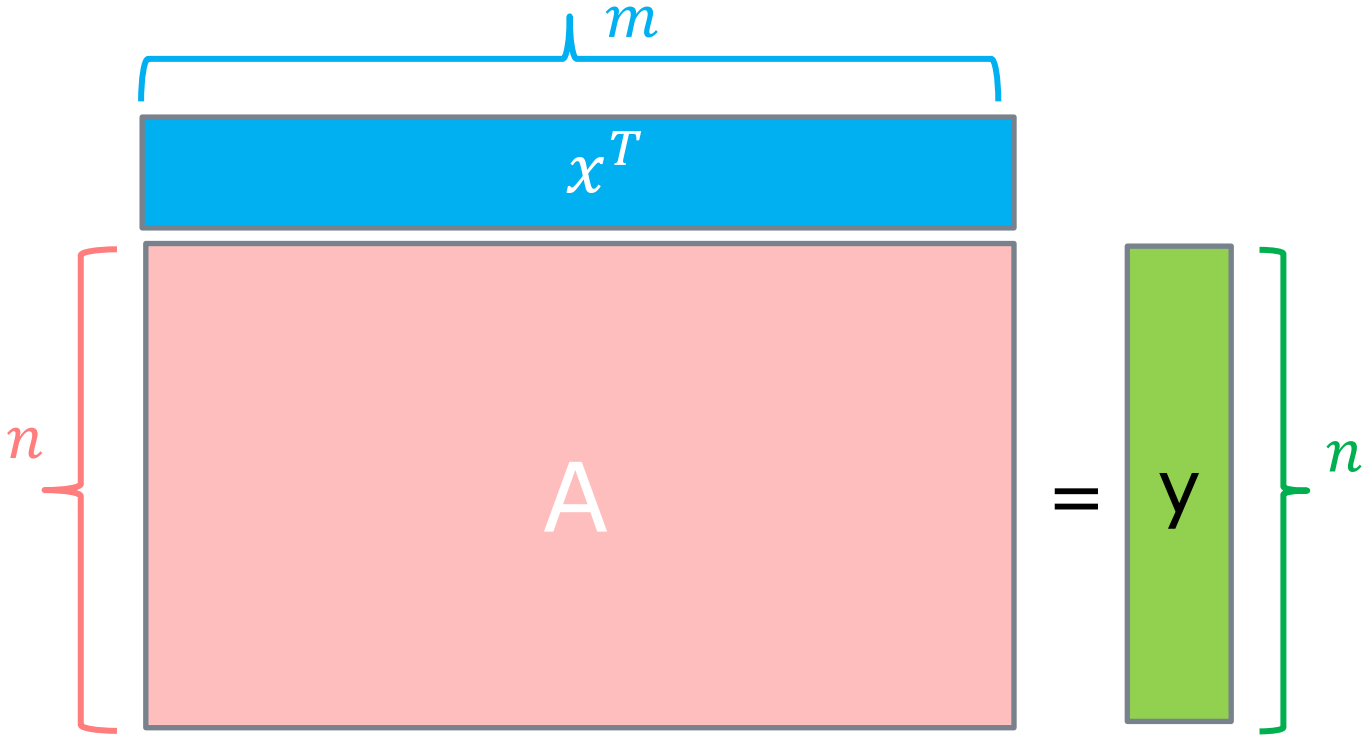
$$f_A(x) = \left[\begin{pmatrix} \vdots \\ \boldsymbol{a_1} \\ \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ \boldsymbol{a_2} \\ \vdots \end{pmatrix} \dots \begin{pmatrix} \vdots \\ \boldsymbol{a_m} \\ \vdots \end{pmatrix} \right] \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = y$$

ただし、$x_1, x_2, \dots, x_m \in \{0,1\}$である。

$$f_A\colon x \in \{0,1\}^m \to y \in \mathbb{Z}_q^n$$

Ajtaiの関数 f_A

次のように表した方が、ベクトルx,yと行列Aのサイズの関係がわかりやすいかもしれない。



$$f_A(x) = Ax = y$$

関数 f_A ： Ajtaiの「一方向関数」

- f_A: $x \in \{0,1\}^m \rightarrow \mathrm{y} \in \mathbb{Z}_q^n$
- 関数 f_Aの入力xと出力y のビット数を考える。
  - x $\in \{0,1\}^m$のビット数 $= m\ bit$
  - y $\in \mathbb{Z}_q^n$ のビット数 $\approx$ n個の$\mathbb{Z}_q$のビット数$= n \log_2 q\ bit$

  $\mathbb{Z}_q = (0,1,\ldots,q-1)$ だから、$q < 2^x, \log_2 q < x$ で、$\mathbb{Z}_q$の要素は高々$\log_2 q$ ビットで表現されることがわかる。1byteで表現される整数は$\mathbb{Z}_{256}$で、$\log_2 256 = 8$ ビット。
- 関数 f_Aは、m ビットの入力を、$n \log q$ ビットの出力に変える。
- $m > n \log q$ なら、関数 f_Aは、入力情報xを出力情報yに圧縮する。
- $m > n \log q$ なら、関数 f_Aは、一方向関数である。

電子署名とHash関数の性質

情報を圧縮する一方向関数をHash関数と呼ぶ。Hash関数は、文書の真正性を保証する電子署名に用いられる。

電子署名のHash関数Hは、もちろん $x \neq y \rightarrow H(x) \neq H(y)$ を満たさねばならないが、それに加えて、次のような性質が求められる。

- y=H(x)が与えられた時、H(x′)=y となるような、x′を見つけることは困難である。署名から、元の文書を復元することはできないということ。
- xが与えられた時、H(x)=H(x′)となるx′を見つけることは困難である。また、H(x)=H(x′)となるx,x′を見つけることは困難である。同じ署名を持つ文書を見つけることは困難である。

最後の性質をHash関数が「衝突に対して耐性を持つ」という。

関数 f_Aの衝突耐性とSIS問題

f_Aの衝突耐性をみてみよう。

f_Aの二つの異なる入力$x \in \{0,1\}$と x' $\in \{0,1\}$について、Hash値の衝突が起きたとしよう。

$$f_A(x) = f_a(x') \Leftrightarrow Ax = Ax'$$

この時、$Ax - Ax' = A(x - x') = 0$

$x - x' = z$とすると、$Az = 0$

$x \in \{0,1\}$、x' $\in \{0,1\}$だから、

$$x - x' = z \in \{-1,0,+1\}$$

$Hash\ f_A$で衝突が起きたとすれば、SIS問題は解を持つ。

逆にSIS問題を解くことが困難であれば、f_Aは衝突耐性を持つ。

Dual ラティス

ラティス問題への重要なアプローチ
「Dualなラティス」で考える

ラティス問題の複雑性を考える時、重要なアプローチがある。
それは、あるラティスの双対形 Dual ラティスで問題を考える事だ。

Regevの有名な論文も、この手法の繰り返しに、ひとつの特徴がある。

少し単純化して言うと、あるラティスの双対形は、元のラティスより目の細かいラティスになる。いくつかのラティス問題にとっては、この性質は、役に立つ。

ここでは、Dualなラティスについて、基本的なことをまとめた。

ラティスLとそのDual L^*

$b_1, b_2, \ldots, b_n \in \mathbb{R}^m$をラティスLの基底とする。ラティス$L(b_1, b_2, \ldots, b_n)$の要素は、基底の整数倍の和で表される。

$$L(b_1, b_2, \ldots, b_n) = \{\sum x_i b_i \mid x_i \in \mathbb{Z}\}$$

基底 $b_1, b_2, \ldots, b_n$から作られる行列をBとすると。

$$L(B) = L(b_1, b_2, \ldots, b_n) = \{Bx \mid x \in \mathbb{Z}^n\}$$

と表してもよい。

この時、ラティスLのdual L^* を次のように定義する。

$$L^* = \{y \in \mathbb{R}^n \mid \forall x \in L, < x, y > \in \mathbb{Z}\}$$

ラティスLのdual L^*は、ラティスLのすべての点xに対して、xとyの内積$< x, y >$が整数値を取る、すべてのベクトル$y \in \mathbb{R}^n$の集合である。

ラティスLとそのDual L^* の例

n個の整数からなるベクトル$\mathbb{Z}^n$の集合を考える。

$\mathbb{R}^n$の正規直交基底を $e_1, e_2, \ldots, e_n$ とすれば、

$$\mathbb{Z}^n = L(e_1, e_2, \ldots, e_n) = \{\sum n_i e_i \mid n_i \in \mathbb{Z}\}$$

と表すことができるので、$\mathbb{Z}^n$はラティスである。

ラティス$\mathbb{Z}^n$の$dual$ を考えよう。

ラティス$\mathbb{Z}^n$の点xは、$x = \sum n_i e_i$で表される。 $n_i \in \mathbb{Z}$。

$(\mathbb{Z}^n)^*$の点yは、$y \in \mathbb{R}^n$だから y= $\sum r_i e_i$で表される。$r_i \in \mathbb{R}$ 。

$< x, y > \in \mathbb{Z}$ から

$$< x, y > = < \sum n_i e_i , \sum r_j e_j > = \sum n_i r_i e_i = n_i r_i \in \mathbb{Z}$$

$n_i \in \mathbb{Z}$であるので、$r_i \in \mathbb{Z}$, がわかる。よって、$y \in \mathbb{Z}^n$

$$(\mathbb{Z}^n)^* = \mathbb{Z}^n$$

ラティスLとそのDual L^* の例

n個の偶数からなるベクトル$2\mathbb{Z}^n$の集合を考える。
$\mathbb{R}^n$の正規直交基底を $e_1, e_2, \ldots, e_n$ とすれば、

$$2\mathbb{Z}^n = L(2e_1, 2e_2, \ldots, 2e_n) = \{\sum n_i(2e_i) \mid n_i \in \mathbb{Z}\}$$

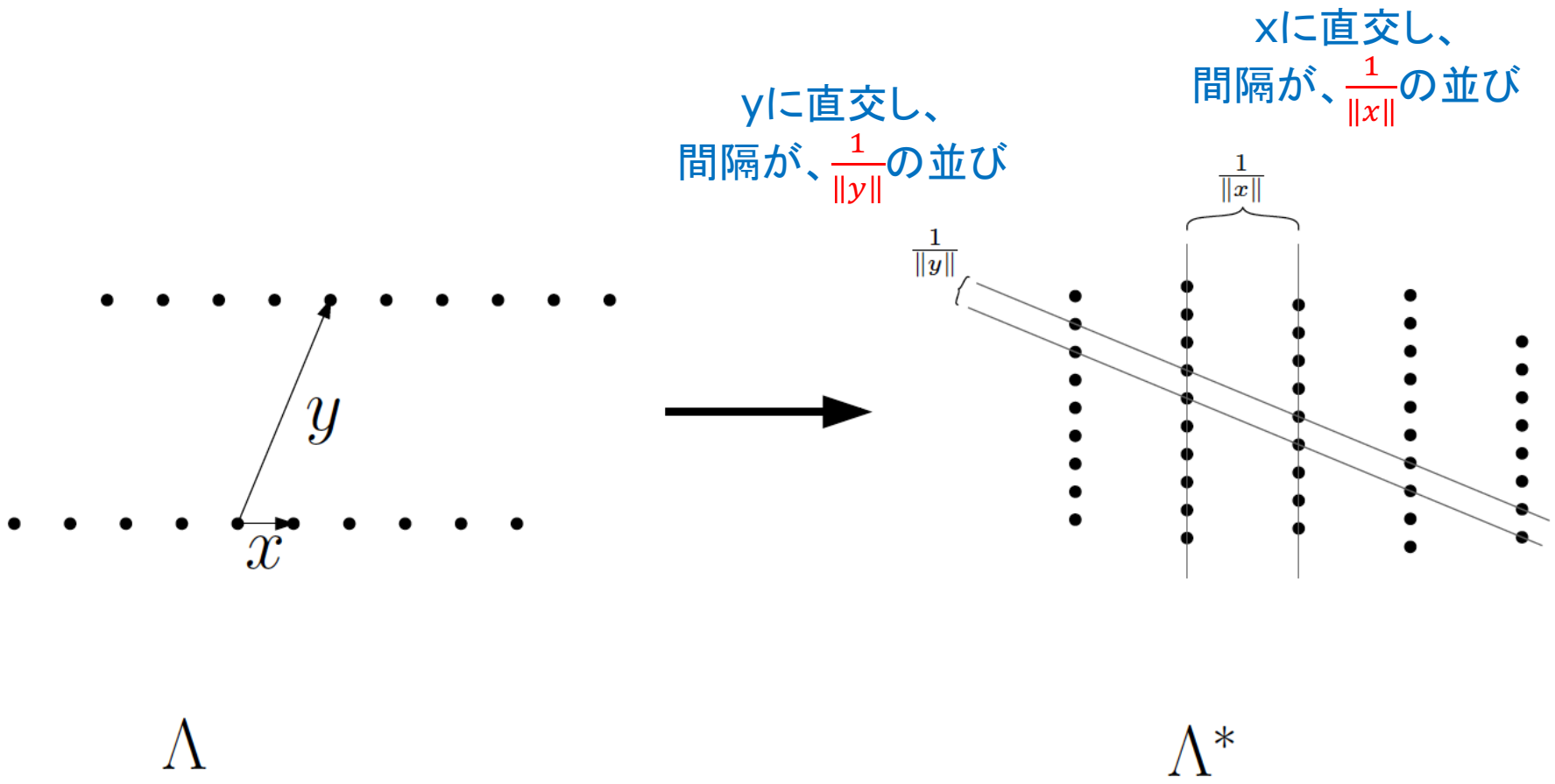
$2\mathbb{Z}^n$はラティスでその点xは、$x = 2\sum n_i e_i$で表される。$n_i \in \mathbb{Z}$。
$(2\mathbb{Z}^n)^*$の点yは、$y \in \mathbb{R}^n$だから y= $\sum r_i e_i$で表される。$r_i \in \mathbb{R}$ 。
$< x, y > \in \mathbb{Z}$ から

$$< x, y > = < 2\sum n_i e_i , \sum r_j e_j > = 2\sum n_i r_i e_i = 2n_i r_i \in \mathbb{Z}$$

$n_i \in \mathbb{Z}$であるので、$r_i/2 \in \mathbb{Z}$でなければならない。よって、$y \in \frac{1}{2}\mathbb{Z}^n$

$$(2\mathbb{Z}^n)^* = \frac{1}{2}\mathbb{Z}^n$$

ラティスLとそのDual L^* の例



ラティスLの基底とそのDual

ラティスLの基底を$\mathrm{B} = (b_1, b_2, \ldots, b_n) \in \mathbb{R}^{m \times n}$ とする。
この時、基底Bのdual $\mathrm{D} = (d_1, d_2, \ldots, d_n) \in \mathbb{R}^{m \times n}$を、次のように定義する。

- BとDは、同じ空間を張る。
- $B^T D = I$

二つ目の条件は、$< b_i, d_j > = \delta_{ij}$ に等しい。
Dは$(B^T)^{-1}$ で与えられる。
また、$B^T = D^{-1}$.で$DB^T = I$である。
これから、$DB^T B = B$。右から$(B^T B)^{-1}$ を掛けて、$D = B(B^T B)^{-1}$ である。これは後で$(L^*)^* = L$を示すときに使う。

L^*はラティスである

Dが基底Bのdual なら、$\left(L(B)\right)^* = L(D)$であることを示す。

まず、$L(D) \subset \left(L(B)\right)^*$ であることは、次のようにしてわかる。

$x \in L(D)$としよう。ある$a_i \in \mathbb{Z}$について$x = \sum a_i b_i$ と表せる。

$$< x, d_j > = \sum a_i < b_i, d_j > = a_i \in \mathbb{Z}$$

これから$x \in \left(L(B)\right)^*$がわかる。

次に、$\left(L(B)\right)^* \subset L(D)$ をしめそう。

$y \in \left(L(B)\right)^*$なるyを一つ取る。BとDは同じ空間を張るので、ある$a_i \in \mathbb{R}$について、$y = \sum a_i d_i$ と表せる。

$$< y, b_j > = \sum a_i < d_i, b_j > = a_i \in Z$$

これから、$y \in L(D)$がわかる。

L^*はラティスである

$L(D) \subset \left(L(B)\right)^*$ で、$\left(L(B)\right)^* \subset L(D)$ から、$\left(L(B)\right)^* = L(D)$。

L^*は、基底Bのdual な基底Dを基底とするラティスである。

ラティスLについて、$(L^*)^* = L$。

BをLの基底とする。この時、$D = B(B^TB)^{-1}$はL^*の基底である。

$$B(B^TB)^{-1} \cdot \left((B(B^TB)^{-1})^{\mathrm{T}} \cdot B(B^TB)^{-1}\right)^{-1} = \mathrm{B}$$

Bは、$(L^*)^*$ の基底である。

$\det(L^*) = 1/\det(L)$

$$\det(L^*) = |\det(B^T)^{-1}| = \left|\frac{1}{\det B^T}\right| = \left|\frac{1}{\det B}\right| = \left|\frac{1}{\det L}\right|$$

Regevの登場

「ポスト量子暗号」

「量子暗号のあと」を指すものでも、「量子のあと」をイメージしたものでもない「ポスト量子暗号」という言葉は、僕には、少し奇妙なものに思われる。

量子暗号と呼ばれるものは現実的な技術としては存在しない。量子コンピューターは世界中でまだラボの中にしかなく、量子通信の利用者（もちろん実験的）は、多く見積もっても一万人を超えない。量子の時代は、まだ、始まっていない。

では、なぜ、「ポスト量子暗号」なのか？

“Post Shor” の時代

ただ、特に暗号の世界で、「量子」が強く意識されるのには理由がある。それは、「量子コンピュータを使えば、現在の暗号は簡単に破れる」という「Shorのアルゴリズム」の発見されたからだ。

その衝撃的な発見から40年が経とうとしているのだが、幸か不幸か、量子コンピュータはいまだShorのアルゴリズムを実行する能力を持たず、多くの人の日常的な意識の中では、暗号技術が危ないという危機意識はほとんど共有されていない。

むしろ、「暗号通貨」や「暗号資産」といった形での暗号を利用した技術への関心は高く、その利用も、かつてなく広がっている。

その意味では、確かに、現在は “Post Shor” の時代である。

もう一つの “Post Shor”

暗号資産の未来を謳歌しようとする層に比べて、過去のトラウマを抱えている暗号の世界の専門家の意識は、すこし複雑である。彼らは、しばらくの間だが、トラウマを癒し克服する確信を見つけられないでいた。

現代暗号の理論的基礎が、計算複雑性の理論であることは理解していても、現実に利用されている暗号は、「素因数分解問題」「離散対数問題」にしろ、いずれも経験的に見つけ出されたものだ。そのいずれもが、理論的には、「Shorのアルゴリズム」の攻撃の射程内にあることを、彼らは、よく知っていた。

かといって、その理論上の脆弱性を現実的な問題として提起するのにも躊躇があった。解決策が見えなかったからだ。

“Post Shor” で “Pre Quantum” の中間時代
「ポスト量子暗号」時代

こうした状況を大きく変えたのは、Regev だった。

Regev に先行して、Ajitaiは、暗号理論の基礎に、ラティス問題を置くことを提案していた。

Regev は、この道を理論的に飛躍的に発展させ、さらに、その理論に基づく、ほとんどのマシン・デバイスにも実装可能で、かつ「量子耐性」を持つと目される LWE暗号を提案した。

Regevの登場によって、“Post Shor” で “Pre Quantum” の中間時代である「ポスト量子暗号」時代が始まったと、僕は考えている。

Oded Regev

「ポスト量子暗号」といわれる暗号技術の新しい時代を、理論・技術の両面で切り開いた中心人物。

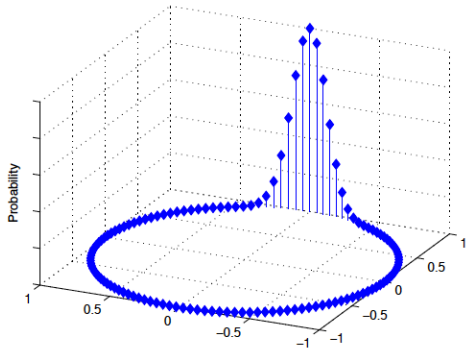
ラティス問題の複雑性の分析に新しいアプローチを導入し、その後の暗号の基礎理論の研究を方向づけた。

「ポスト量子暗号」の代表的な暗号技術である LWE暗号の提唱者。

On Lattices, Learning with Errors, Random Linear Codes, and Cryptography

Oded Regev *

May 2, 2009

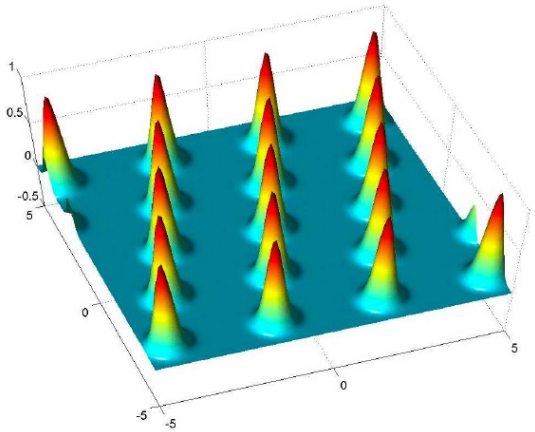


Abstract

Our main result is a reduction from worst-case lattice problems such as GAPSVP and SIVP to a certain learning problem. This learning problem is a natural extension of the 'learning from parity with error' problem to higher moduli. It can also be viewed as the problem of decoding from a random linear code. This, we believe, gives a strong indication that these problems are hard. Our reduction, however, is quantum. Hence, an efficient solution to the learning problem implies a *quantum* algorithm for GAPSVP and SIVP. A main open question is whether this reduction can be made classical (i.e., non-quantum).

We also present a (classical) public-key cryptosystem whose security is based on the hardness of the learning problem. By the main result, its security is also based on the worst-case quantum hardness of GAPSVP and SIVP. The new cryptosystem is much more efficient than previous lattice-based cryptosystems: the public key is of size $\tilde{O}(n^2)$ and encrypting a message increases its size by a factor of $\tilde{O}(n)$ (in previous cryptosystems these values are $\tilde{O}(n^4)$ and $\tilde{O}(n^2)$, respectively). In fact, under the assumption that all parties share a random bit string of length $\tilde{O}(n^2)$, the size of the public key can be reduced to $\tilde{O}(n)$.

1 Introduction



Main theorem. For an integer $n \geq 1$ and a real number $\varepsilon \geq 0$, consider the 'learning from error' problem, defined as follows: the goal is to find an unknown $\mathbf{s} \in \mathbb{Z}_2^n$ given a list of 'errors'

$$\langle \mathbf{s}, \mathbf{a}_1 \rangle \approx_\varepsilon b_1 \pmod 2$$
$$\langle \mathbf{s}, \mathbf{a}_2 \rangle \approx_\varepsilon b_2 \pmod 2$$
$$\vdots$$

Regevの証明概要

LWE問題のラティス問題への還元

ここでは、RegevのLWE問題のラティス問題への還元の証明を見ていこう。

具体的には、LWE問題のSVP問題への還元を見ていく。

基本的なアプローチとしては、ラティスLとそのdual ラティスL^*の双方を考えることと、DGS – Digital Gaussian Sampling という手法をとることに特徴がある。

ラティス Lの世界 ⟺ Dual ラティス L^*の世界

フーリエ変換

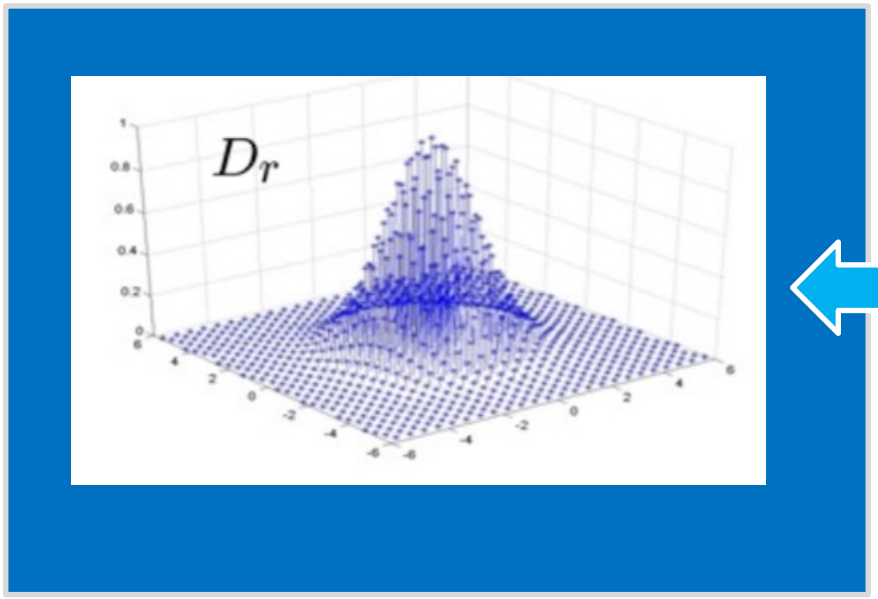
$\forall x \in L,$

$D_r(x) = e^{-\|x/r\|^2}$

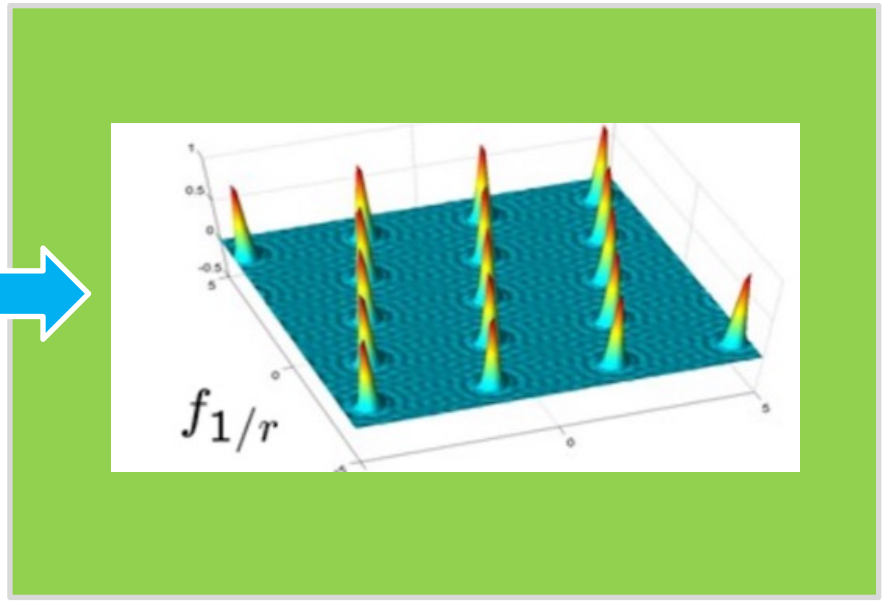
⟺

$\forall x \in L^*,$

$f_{1/r}(x) \approx e^{-\|r \cdot dist(x,L^*)\|^2}$

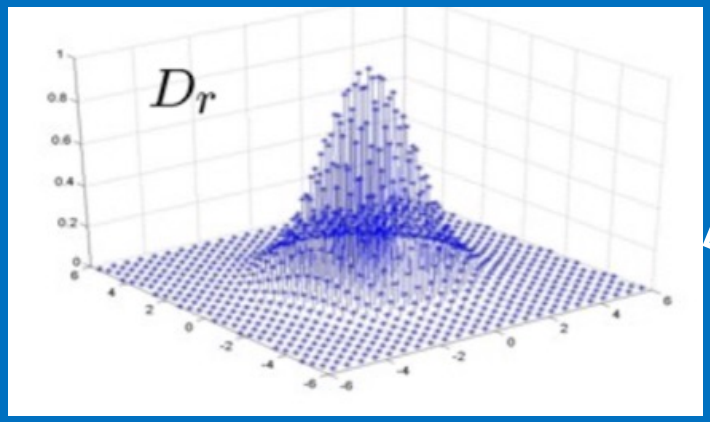


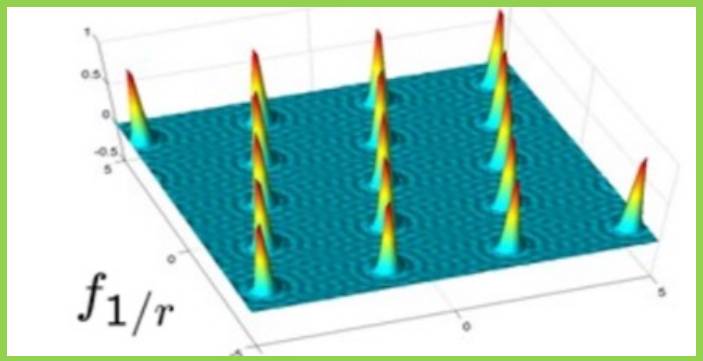
⟺

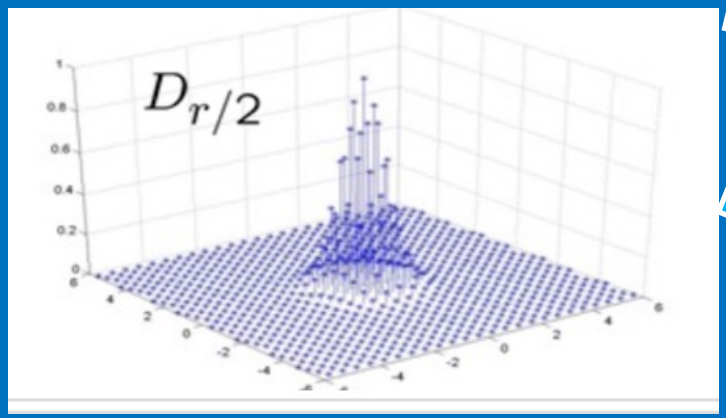


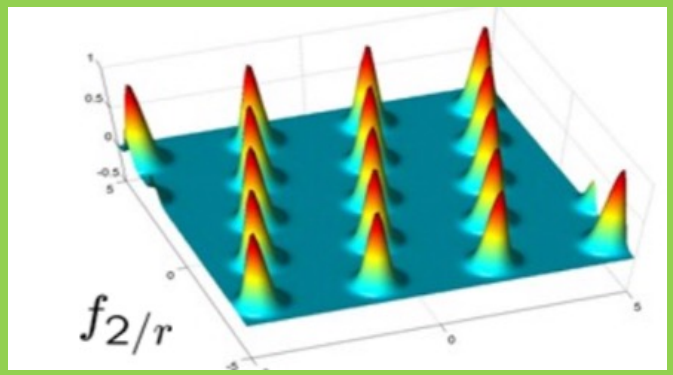
$L = (L^*)^*$

Regevの証明の流れ

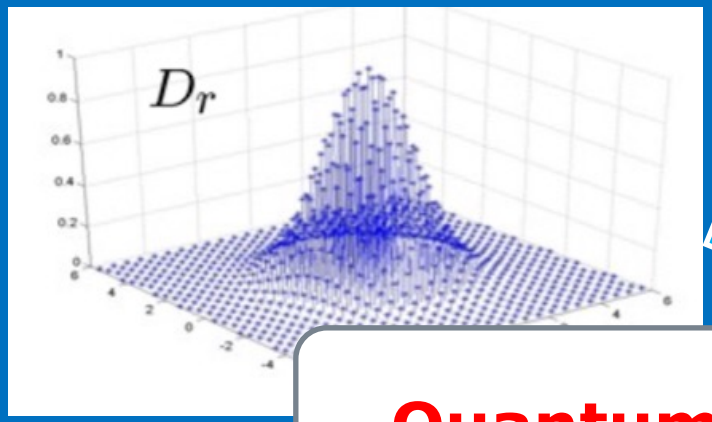




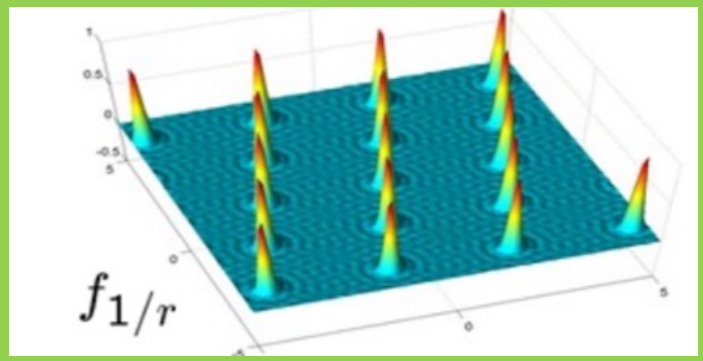




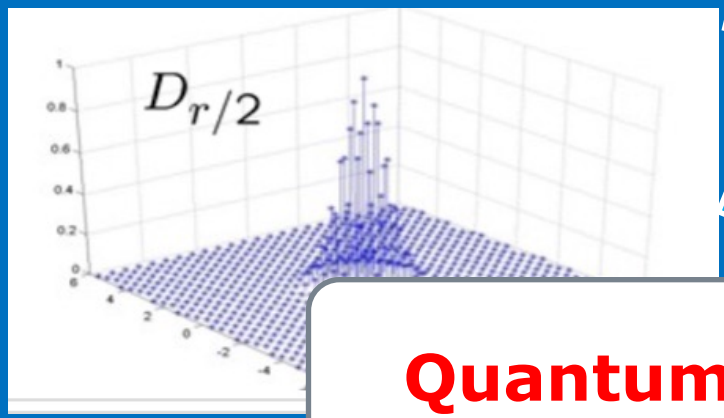
Regevの証明の流れ



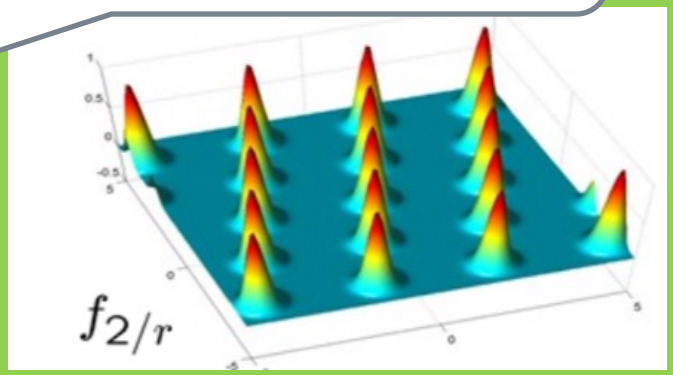
Classic with LWE Oracle



Quantum !



Classic with LWE Oracle



Quantum !

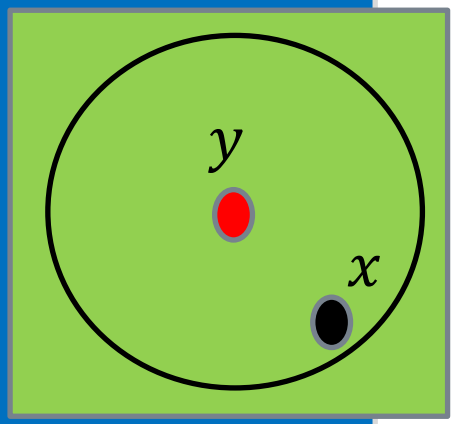
ラティス Lの世界

Dual ラティス L^*の世界

D_rでのサンプリング

$D_r/2$のサンプリングデータとオラクルLWEから、L^*で$CVP_{p/r}$が証明できる

Quantum !



第一レジスターに与えられたxについて、一番近い格子点yを第二レジスターに返す量子回路を考える。

|x,0> -> |x, y>

この回路は、

|x,y> ->|x,0>

の計算が可能である。

ラティス Lの世界

D_rでのサンプリング

Dual ラティス L^*の世界

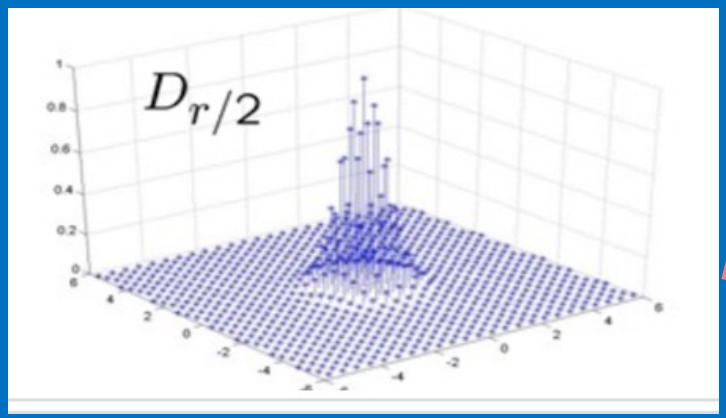
CVPのアルゴリズムは使えるので、この|x>と$D_r/2$からのサンプリングで、次の状態を作れる。

$$\sum_{x\in\mathbb{R}^n} f_{2/r}(x)|x>$$

この式をフーリエ変換する。

ラティス Lの世界

D_rでのサンプリング



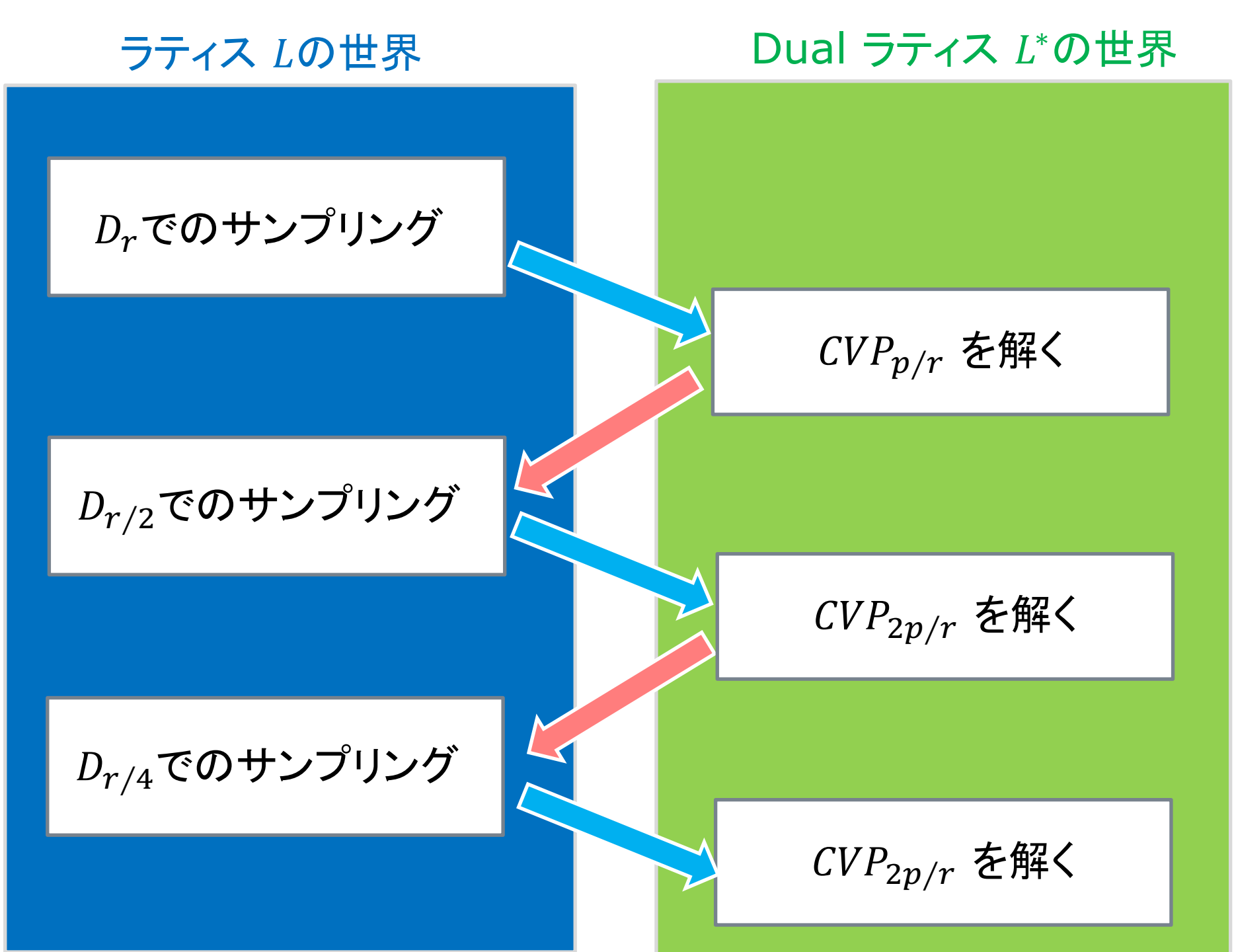
Dual ラティス L^*の世界

CVPのアルゴリズムは使えるので、この|x>と$D_r/2$からのサンプリングで、次の状態を作れる。

$$\sum_{x\in\mathbb{R}^n} f_{2/r}(x)|x>$$

この式をフーリエ変換する。

それは、$D_{r/2}$ にほかならない。


ラティス Lの世界
Dual ラティス L^*の世界
D_rでのサンプリング
$D_{r/2}$でのサンプリング
$D_{r/4}$でのサンプリング
$CVP_{p/r}$ を解く
$CVP_{2p/r}$ を解く
$CVP_{2p/r}$ を解く

LWEのSVCへの還元

LWEを解くアルゴリズムが存在すると仮定する。
次のステップで、それはSVCに還元できる。

- $r = 2^n$ とする。
- D_rからサンプリングする。
- 以下を繰り返す
  - D_rからのサンプルで、$D_{r/2}$ を計算する。
  - $\mathrm{r} \leftarrow r/2$
- もしも、rが小さかったら、Short Vectorとして出力する。


© NASA, ESA/CSA/Judy Sch