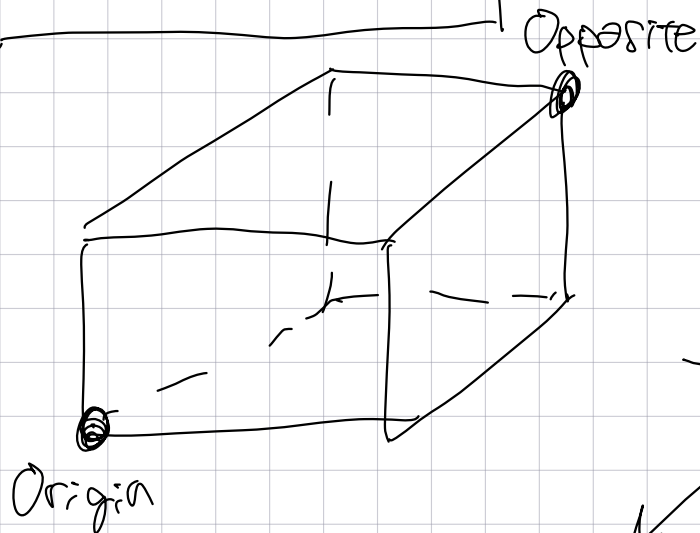
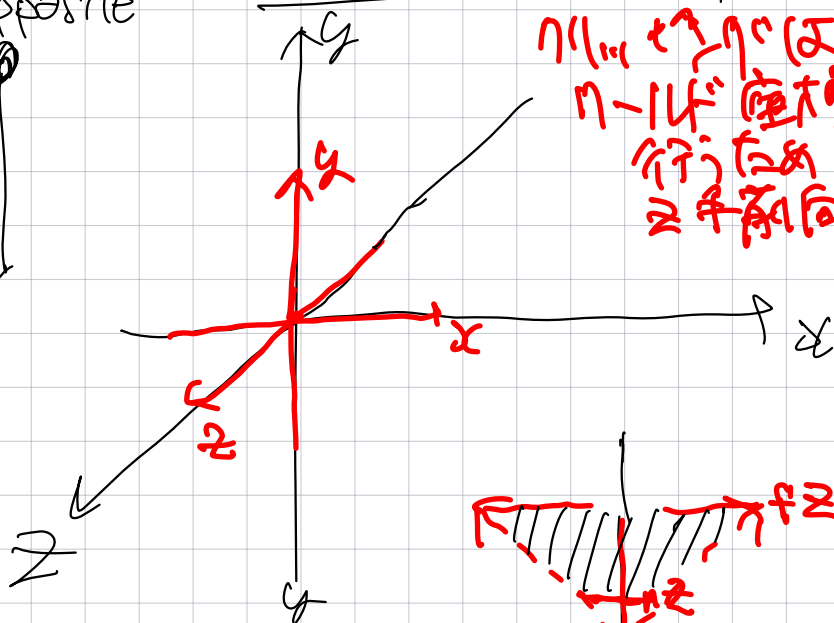


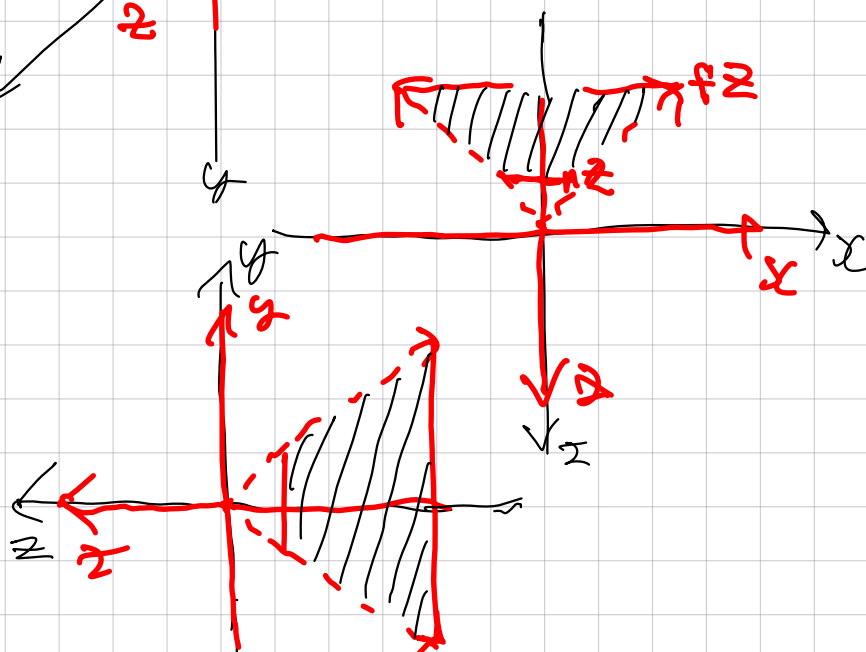
# RANGE\_CUBE



# CLIPPING



711... 711... は  
711... 座標で  
行っている  
2次元向き

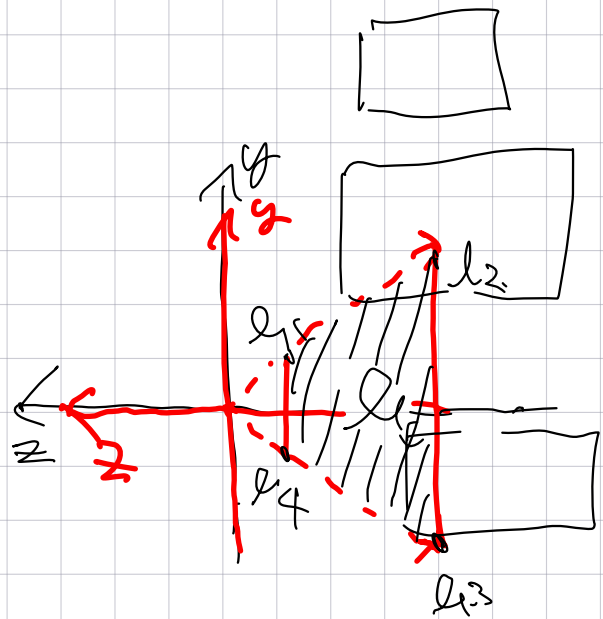
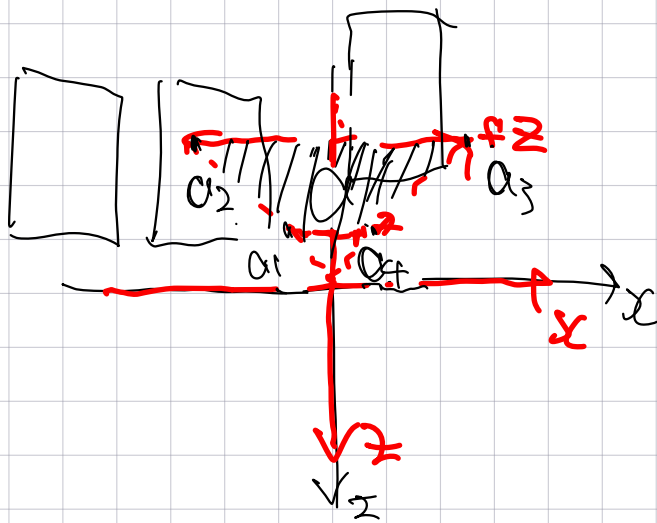


＜処理順＞

1. xz軸での交差判定
2. yz軸での交差判定
3. 1及び2が真の場合、交差判定は真となり、  
一方のみの場合、偽となる



# <交差判定>



・  $x \geq 車道$  の  $l_1, l_2$  の  $l_1$  領域を  $a$ 、 $y \geq 車道$  の  $l_1$  を  $l$  とする  
 $(x, y_1), (x_2, y_2)$

2点を通る直線の方程式は

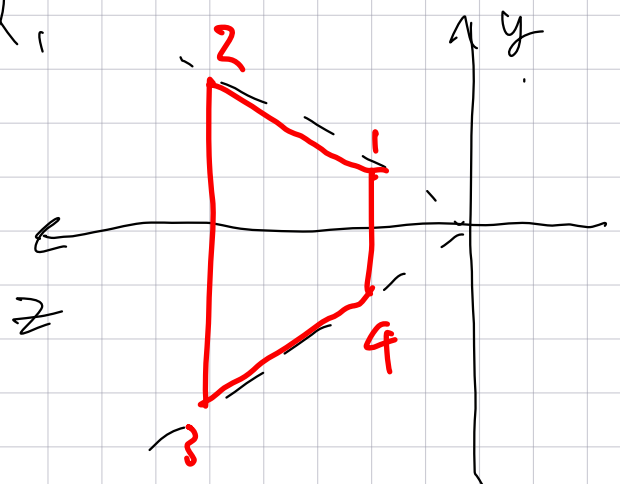
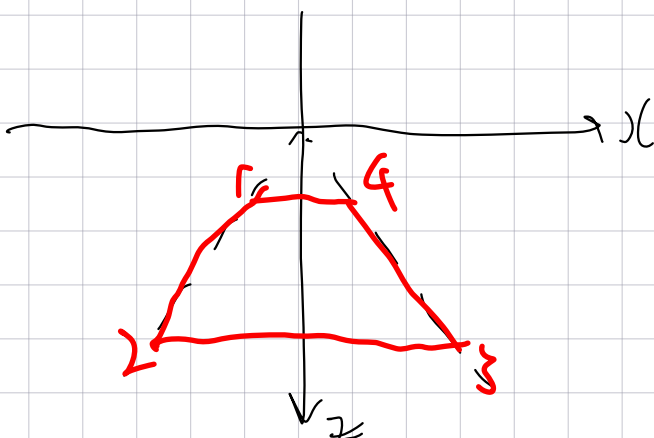
$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) \Leftrightarrow y = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) + y_1$$

$x \geq 車道$ 、 $y \geq 車道$  の  $車道$  は  $車道$  である  $x, y \in a$ 、 $l$  とし上の式を代入して

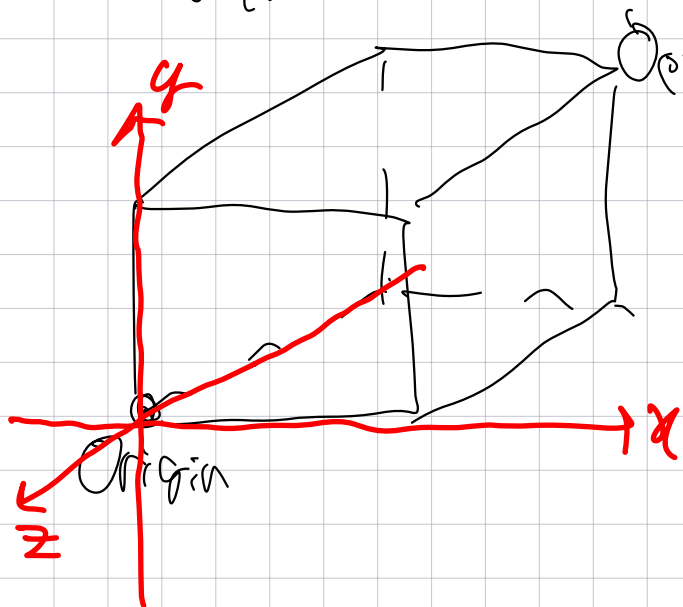
$$z = \frac{z_2 - z_1}{x_2 - x_1} (x - x_1) + z_1$$

二点の座標を使用(24)の式で  $a$ 、 $l$  の領域を決定(24)の式で判定する。

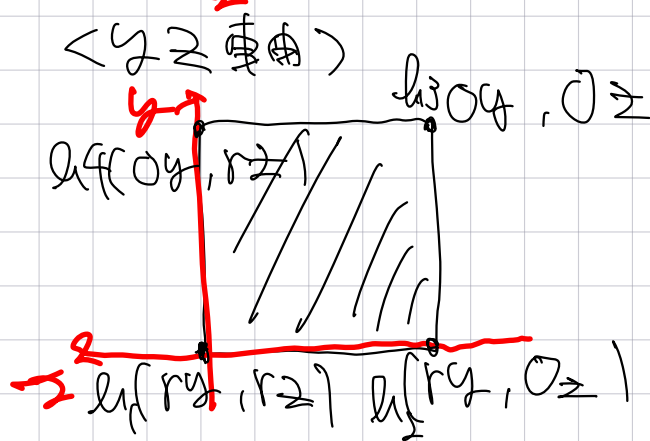
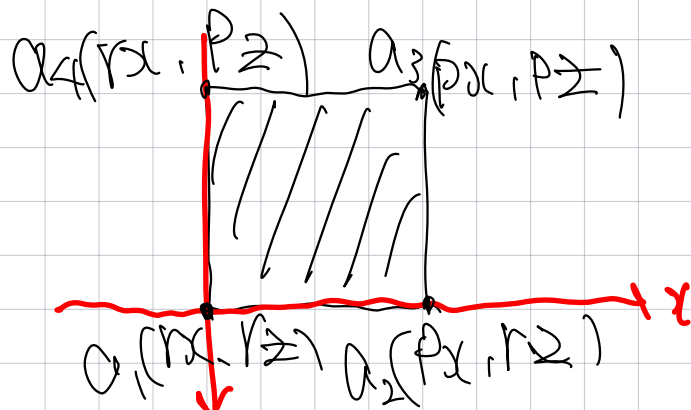
$$x = \frac{x_2 - x_1}{z_2 - z_1} (z - z_1) + x_1$$



# CRANGE\_CUBE $\alpha z$ $z$ 軸での領域



Origin  $\rightarrow r$ , Opposite  $\rightarrow p$   
 Opposite  $\rightarrow (z$  軸)



$$z = \frac{z_2 - z_1}{\alpha_2 - \alpha_1} (\alpha - \alpha_1) + z_1, \quad \alpha \text{ 不変 } \alpha \in [\alpha_1, \alpha_2] \text{ 領域で変}$$

## <Range cube の回転による値の変化>

- ・ Origin, Opposite などなく長方形の8頂点の座標を格納する必要がある。
- ・ 8頂点の移動、回転後、新たに Origin, Opposite を決める  
(オブジェクトのアーキテクチャ処理はゴニゴニ?)

・ その後、領域の重なりによる判定を行う。

## <ポリゴンの表裏判定でリル・ヒンク>

### <表裏判定>

ポリゴンの法線ベクトル (3点の点のベクトルはすべて等しいことが前提) で、カメラから面へ向かうベクトルとの内積を行い、その値が正の場合にのみ面は表を向いているとする

$$\vec{a} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \cdot \vec{b} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = a_1 b_1 + a_2 b_2 + a_3 b_3 \quad \text{で内積が求まる}$$

① 3x1 行列と 3x1 行列の<sup>内積の</sup>値を求め正負の結果を西2列に保存する  
GPU 関数が必要

## <ポリゴンのリル・ヒンク>

