コンピュータグラフィックス基礎

第7回 形状モデリング

三谷 純

3DCG表示

モデリング ⇒ レンダリング

- 対象物を計算機内で表現する
  - 形の定義
  - 表面の材質
  - 光源

今回のテーマ

- 対象物をディスプレイに表示する
  - 投影（座標変換）
  - 照光（反射・屈折の計算）

モデリング

- モデリングとは？
  - 画面表示したい物体の形，位置，大きさなどをコンピュータ内部のデータとして表現すること
  - 出来上がったデータ　→　形状モデル

- 目的に応じた適切なモデリングのために・・・
  - 多面体の表現方法
  - 曲線，曲面の表現方法
  - 自然物，複雑な形状の表現方法

「形」とは？

CGで扱うべきものの「形」は様々にある

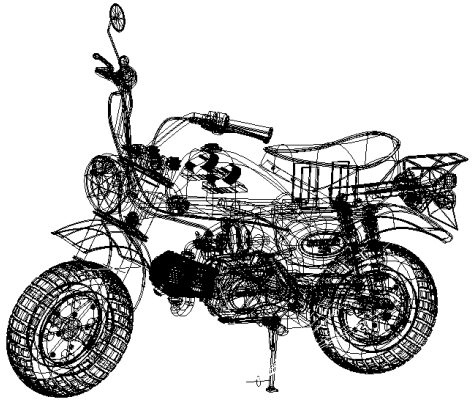
- 機械、建造物、家具（剛体）
- 人間の皮膚、布、服（柔物体）
- 海、滝、川（流体）

内部と外部を区別できる
↓
表面がある

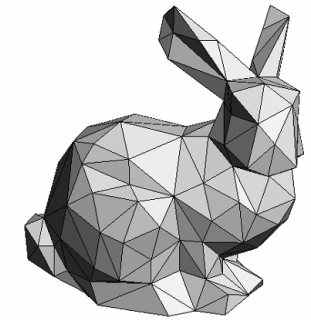
- 炎、煙（定まった形がないもの）
- 羽毛、髪、毛（ふさふさしたもの）

さまざまな形の表現方法 (1)

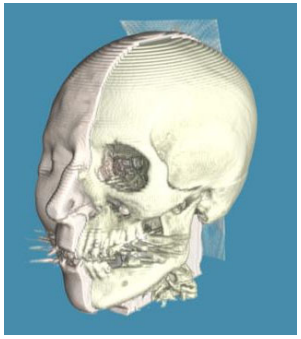
点群



ワイヤフレーム



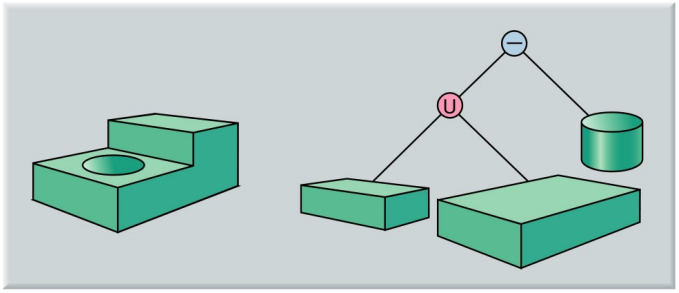
ポリゴンモデル
（三角形メッシュ）



ボクセル表現
（ボリューム表現）

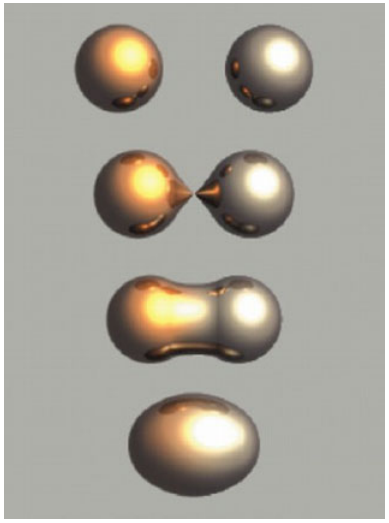
さまざまな形の表現方法 (2)

■図3.7——CSG による表現



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

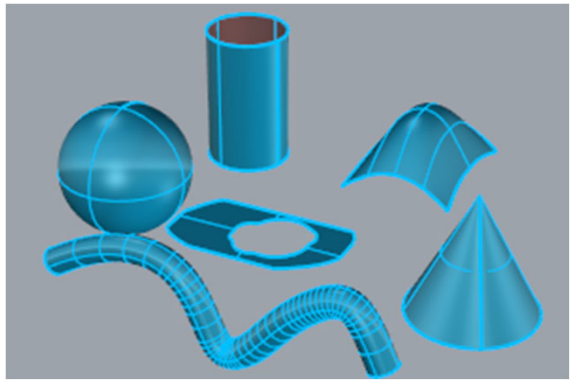
CSG表現（立体の演算）



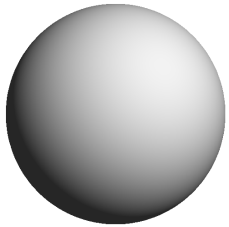
メタボール表現

パーティクル表現

さまざまな形の表現方法 (3)



パラメトリック曲面
（NURBS曲面）

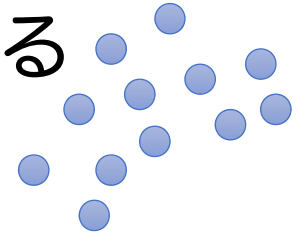


$$\mathbf{F}(x,y,z) = x^2 + y^2 + z^2 - r^2 = 0$$

陰関数表現

点群モデル

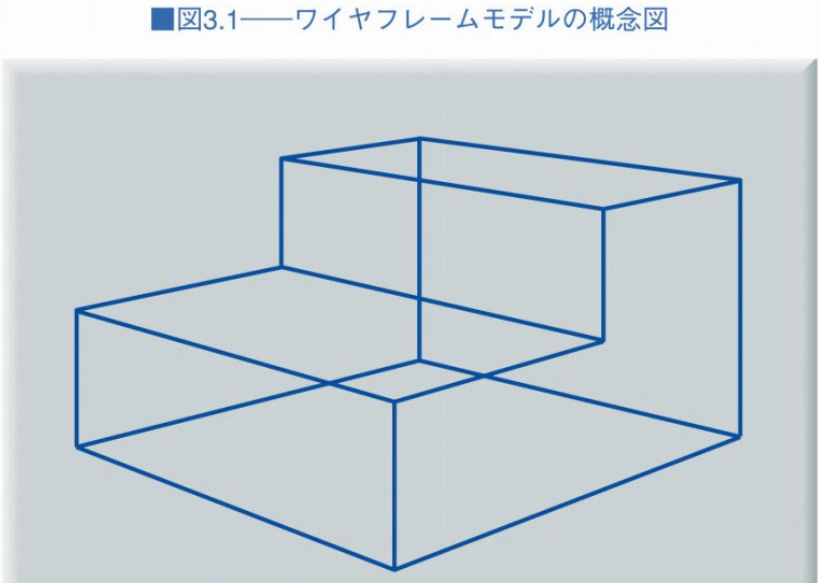
- 頂点座標だけを記録する
  - データ表現が単純
  - 立体を表現できない

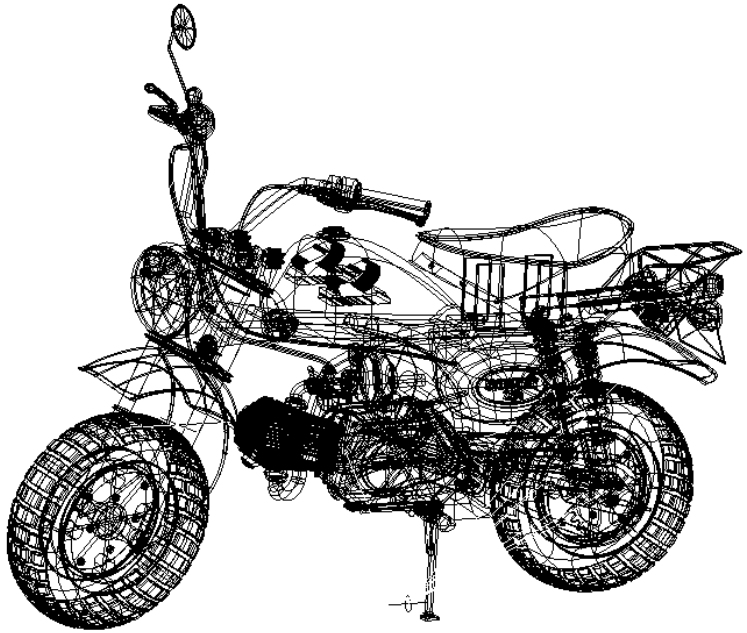


- 3次元形状測定器で得られる
  - point cloud, 点群
  - 後の処理が必要
    - 複数データの位置合わせ
    - 面の生成

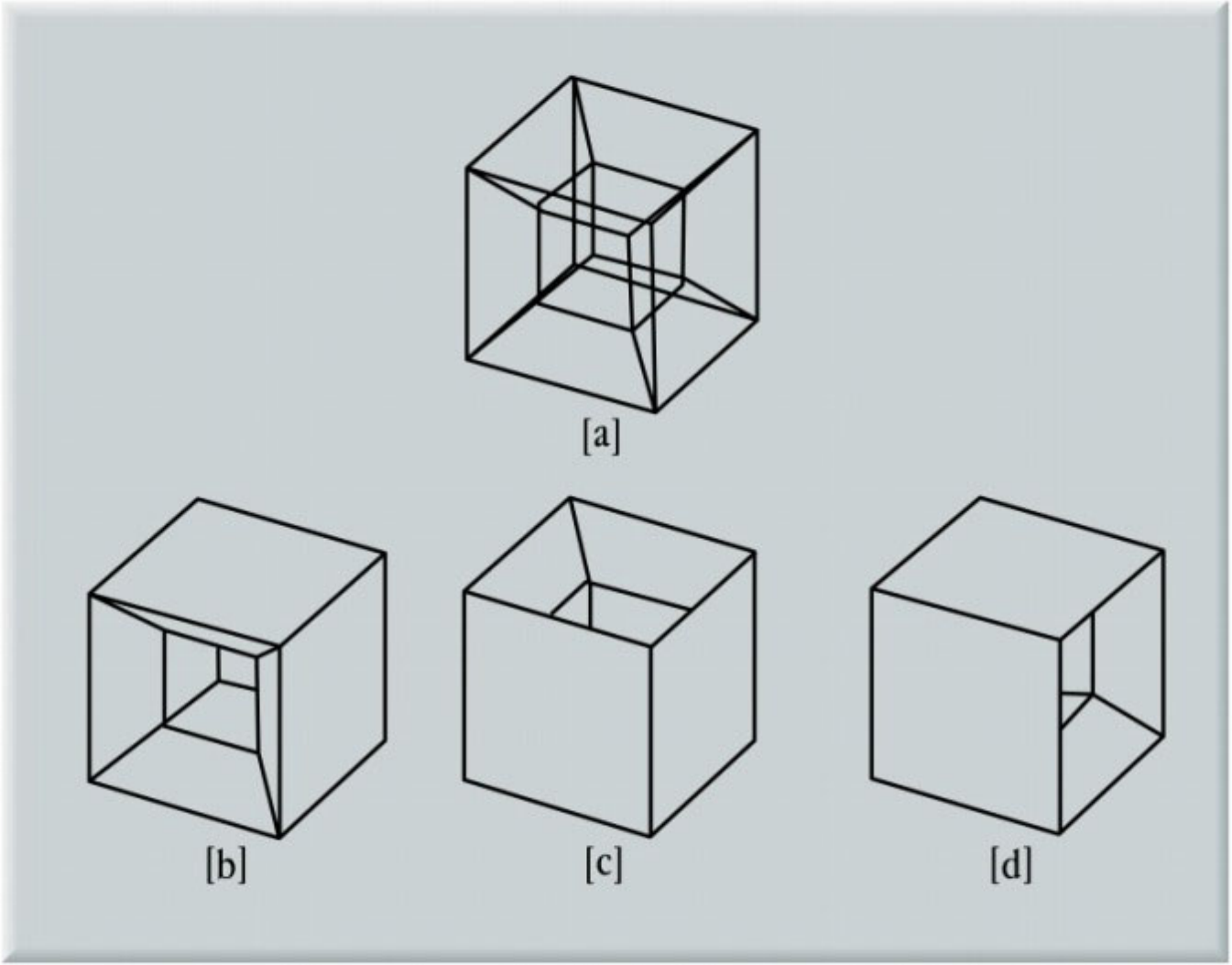
ワイヤーフレームモデル

- 頂点座標，稜線だけを記録する
  - データ表現が単純
  - 計算が容易
    - 計算機の性能が低かった時代によく使われた
  - 欠点
    - 裏側も見えてしまう（複雑な形を把握しにくい）
    - 形が一意に定まらない場合がある





■図3.2——立体を一意に表現できない例

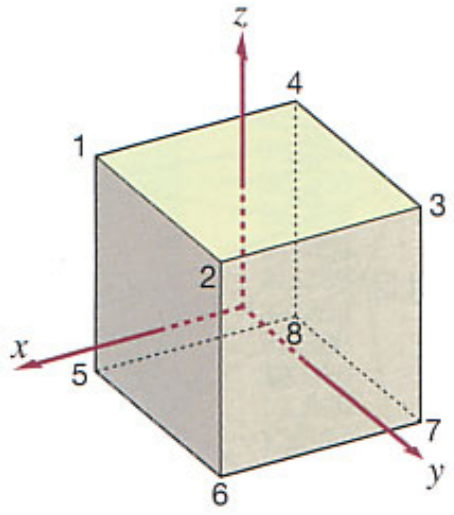


（鳥谷・千代倉『3次元CADの基礎と応用』共立出版）

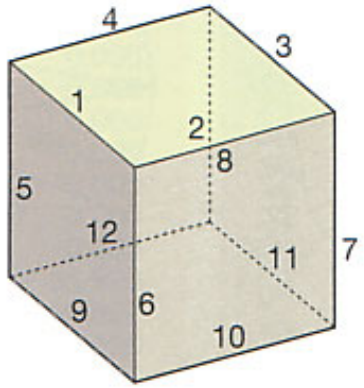
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

ワイヤーフレームモデルのデータ構造

- 頂点リストと稜線リスト



(a) 頂点番号



(b) 稜線番号

図2.19 ■立方体の番号付け

表2.1 ■立方体のモデル
(a)は頂点の座標値，(b)は稜線を構成する頂点の番号．

(a)頂点リスト

頂点番号	x	y	z
1	1	−1	1
2	1	1	1
3	−1	1	1
4	−1	−1	1
5	1	−1	−1
6	1	1	−1
7	−1	1	−1
8	−1	−1	−1

注：中心が原点で一辺の長さ2の立方体

(b)稜線リスト

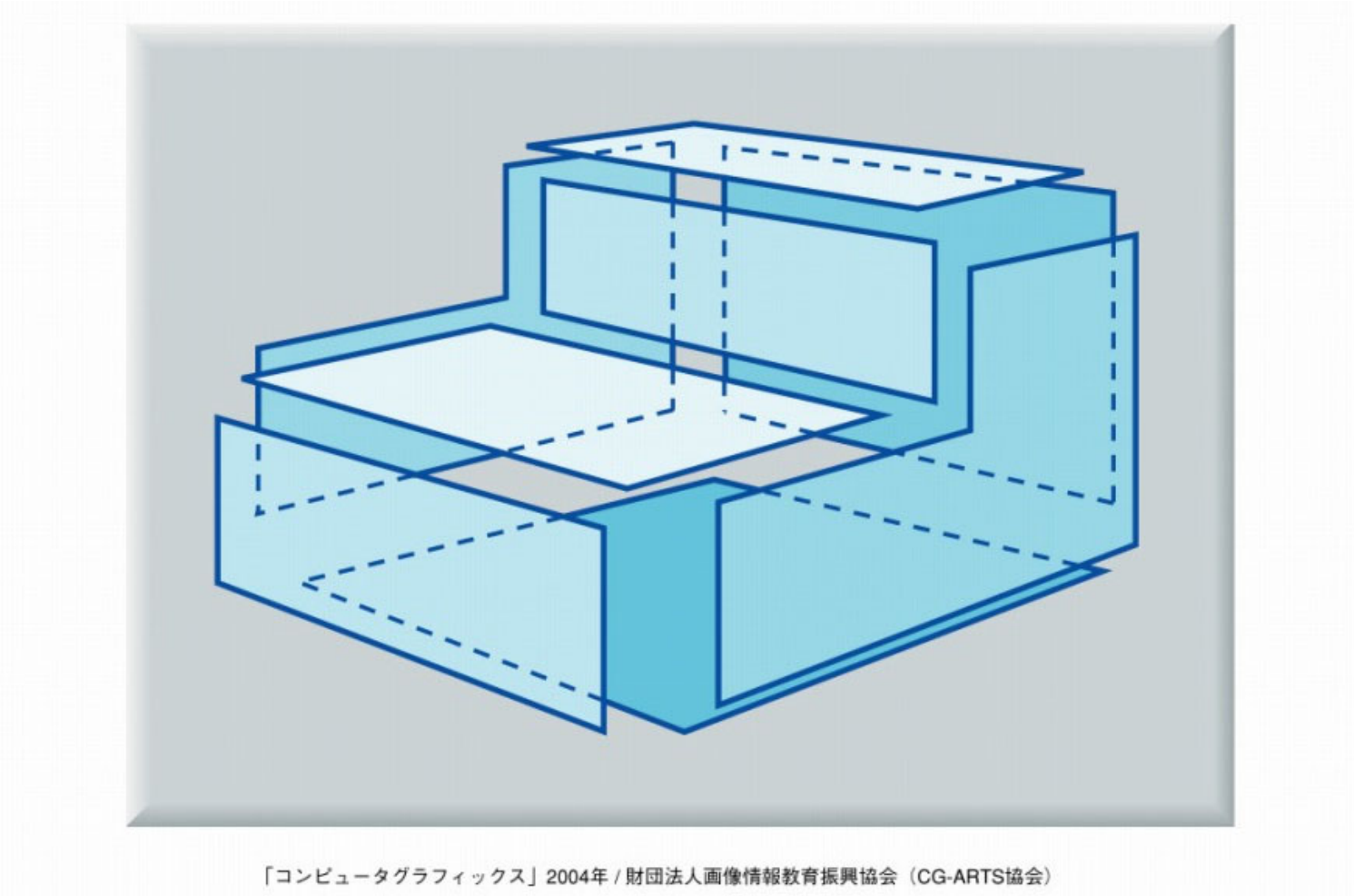
稜線番号	始点	終点
1	1	2
2	2	3
3	3	4
4	4	1
5	1	5
6	2	6
7	3	7
8	4	8
9	5	6
10	6	7
11	7	8
12	8	5

ワイヤフレームモデルの表現

```
class Vertex {
public:
  Vector3d position;
};

class Edge {
public:
  Vertex *sp;
  Vertex *ep;
};

class Model {
public:
  std::vector<Vertex*> vertices;
  std::vector<Edge*> edges;
};
```

サーフェスモデル

- ワイヤーフレーム＋面情報



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

サーフェスモデルのデータ構造

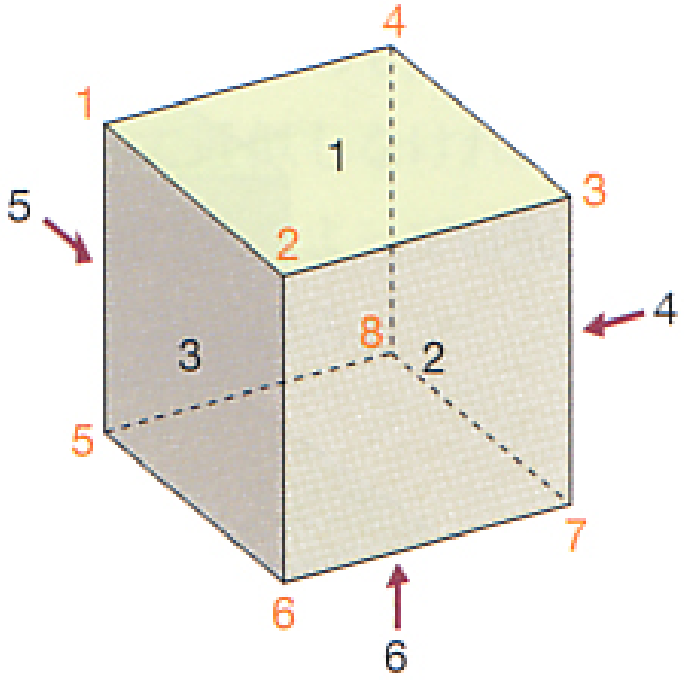


図2.22 ■サーフェスモデル生成のための面の番号付け

表2.2 ■面番号(左図の面の頂点番号リスト)

面番号	頂点番号			
1	1	2	3	4
2	2	6	7	3
3	1	5	6	2
4	3	7	8	4
5	4	8	5	1
6	5	8	7	6

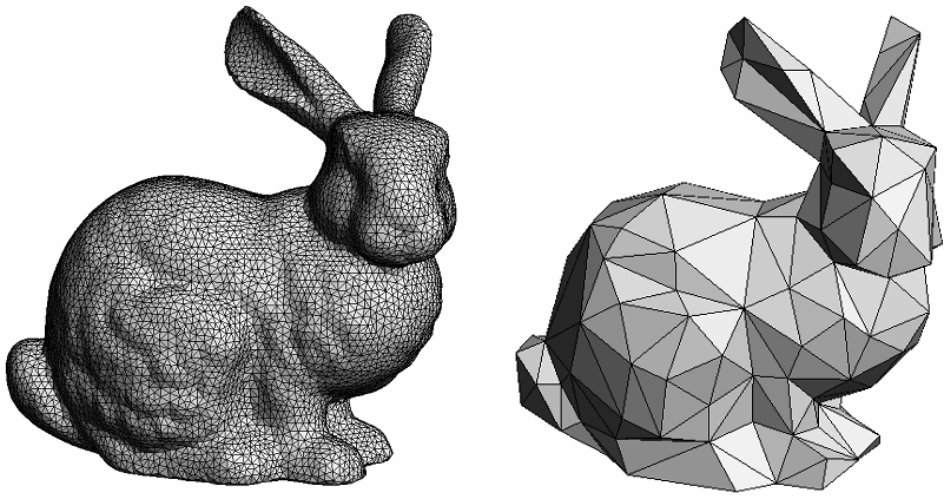
注：頂点番号の並びは表面から見て反時計まわりで統一

サーフェイスモデルの表現

```
class Vertex {
public:
  Vector3d position;
};

class Face {
public:
  std::vector<Vertex*> vertices;
};

class Model {
public:
  std::vector<Vertex*> vertices;
  std::vector<Face*> faces;
};
```

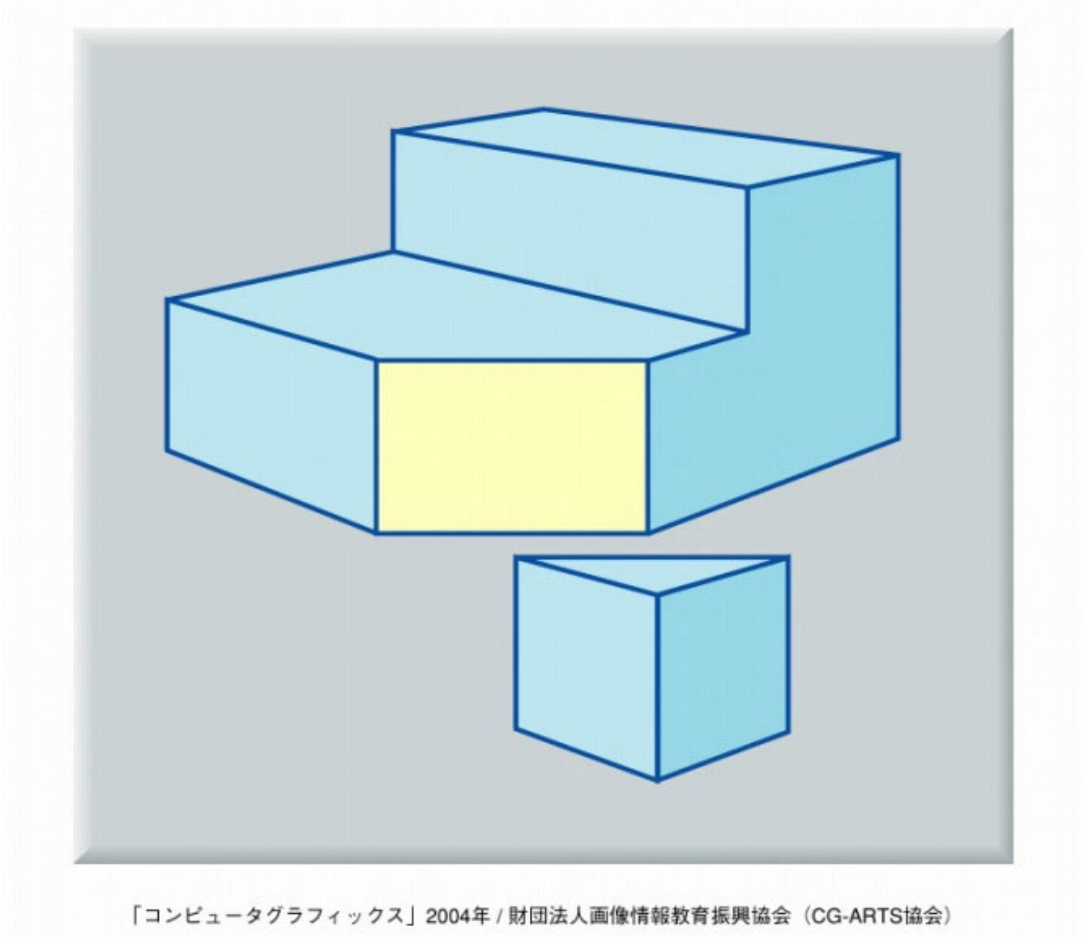
ポリゴンモデル（三角形メッシュ）

- 三角形の集合で形を表現する
  - 自然物など数式で表現しにくい形状を扱いやすい
  - データ構造がシンプル
  - 面の細かさで精度を調整できる（データ量と精度のトレードオフ）
  - 様々な形状処理手法が開発されている



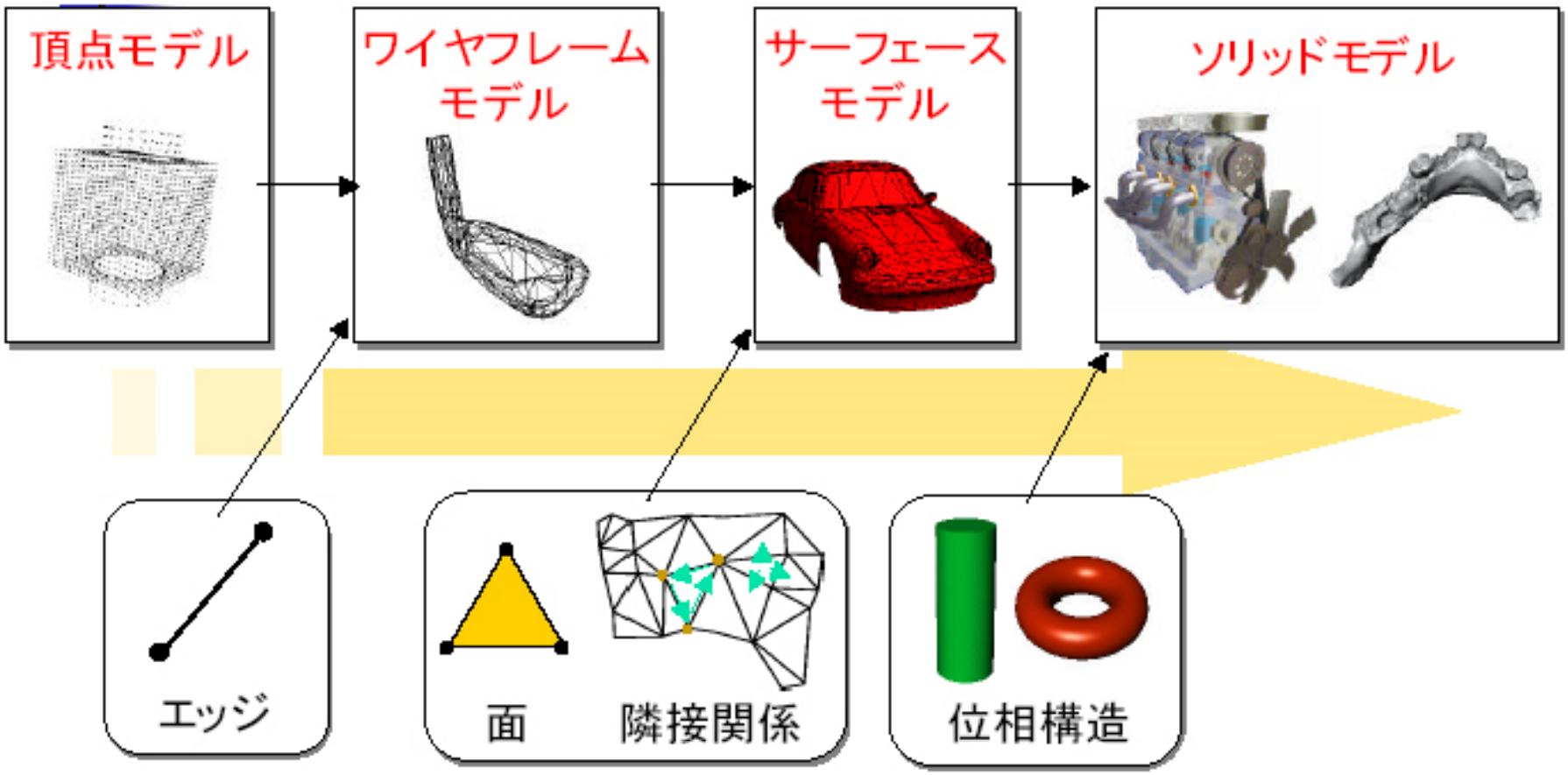
ソリッドモデル

- サーフェスモデル＋物体の内外を区別する情報
(穴の無いサーフェスモデルで表現することも)



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

形状の表現法の進化



ソリッドモデルの形状表現

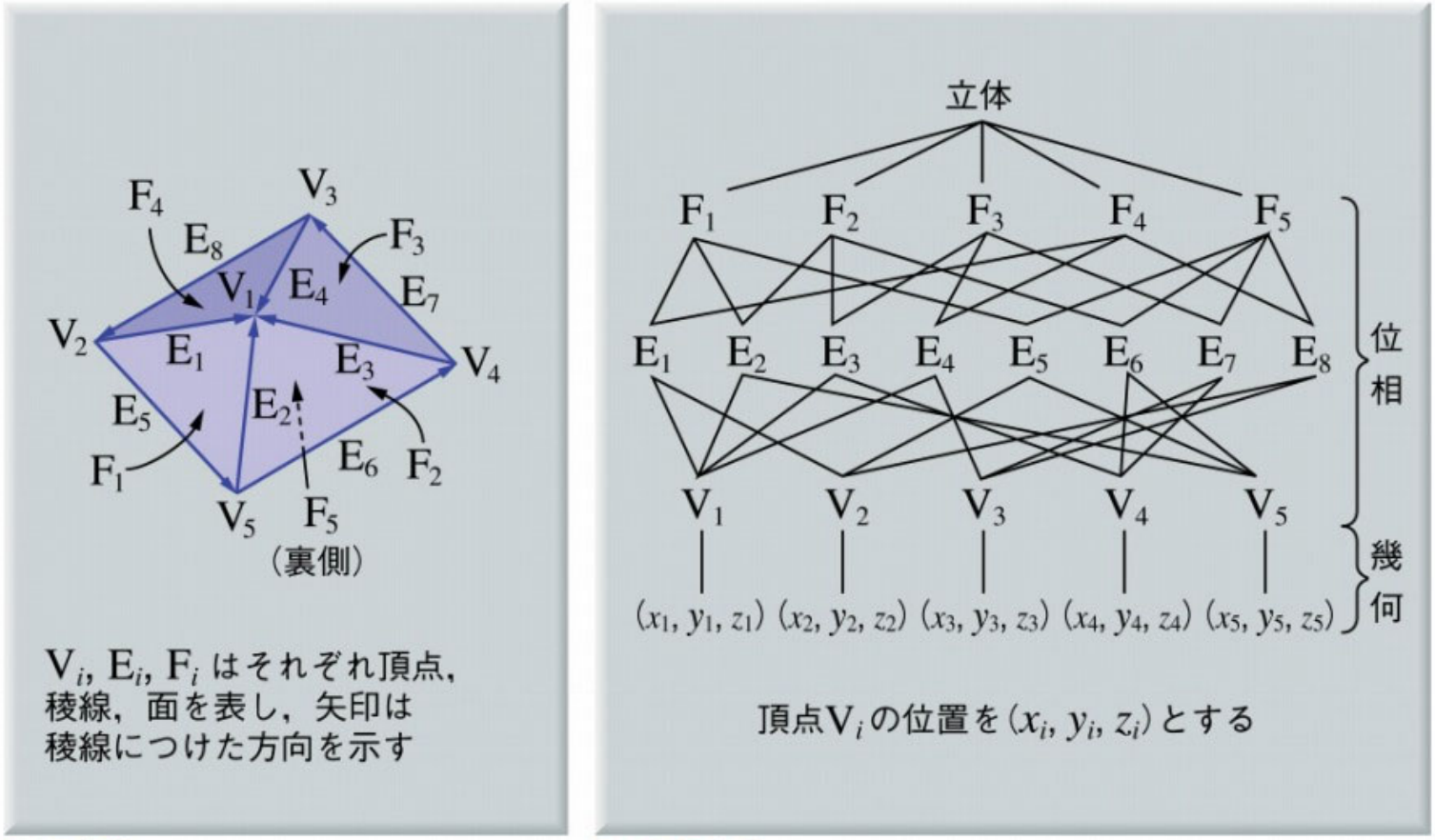
- 建築物や機械部品などを設計するCADの分野で使用される
- 数式で表現される形状を扱うことが多い

  - 境界表現
  - CSG表現
  - スイープ表現
  - 局所変形

境界表現

- 頂点座標と稜線・面の接続関係で立体を表現

■図3.5——四角錐の境界表現



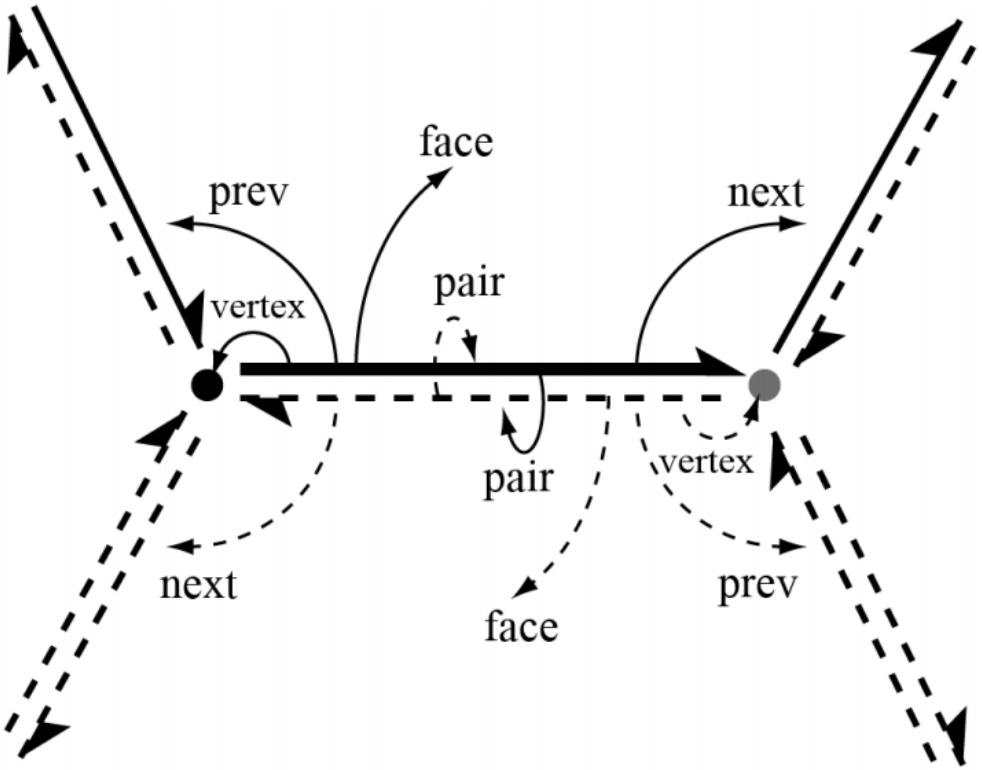
[a] 四角錐　　[b] 境界表現

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

境界表現

- 幾何情報
  - 形状を定めるための情報
  - つまりは頂点の座標値
- 位相情報
  - 立体表面がどのように構成されているかを定める情報
    例
    - どの頂点とどの頂点が稜線で結ばれているか
    - ある面のカドを構成している頂点はどれであるか
    - ある面の周囲を構成している稜線はどれであるか
    - ある面と隣接する面はどれであるか
  - 位相情報をどのようなデータ構造で保持するかには、様々な方法がある（例：ハーフエッジ構造）

ハーフエッジ構造

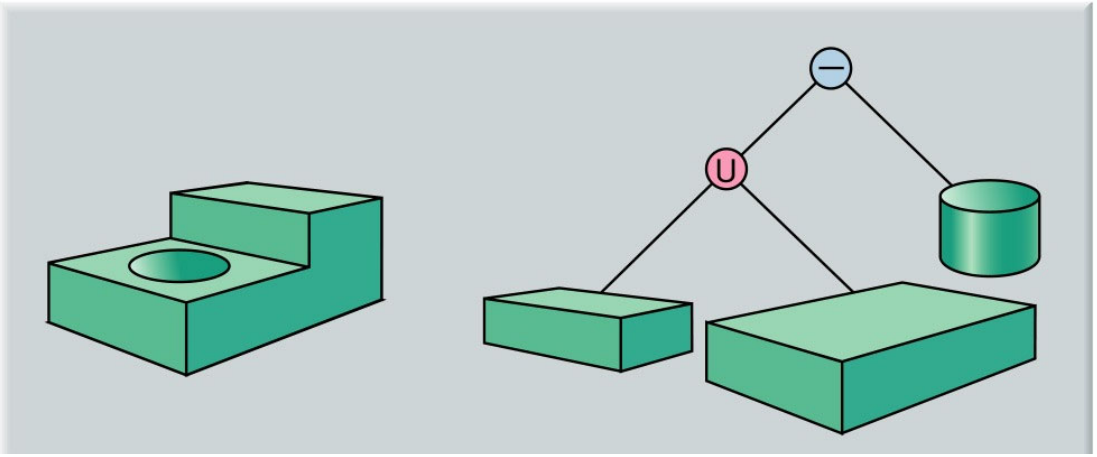


近傍の面、頂点、稜線に高速にアクセスできる

CSG（Constructive Solid Geometry）表現

- 立体をプリミティブ（基本立体）と，その組み合わせで表現
- 基本立体の種類，大きさ，位置情報，結合状態をツリー構造で表す

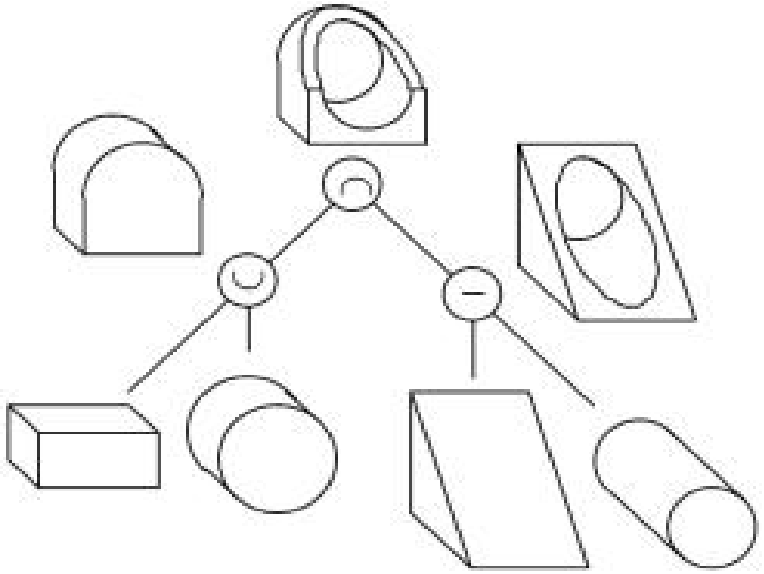
■図3.7——CSG による表現

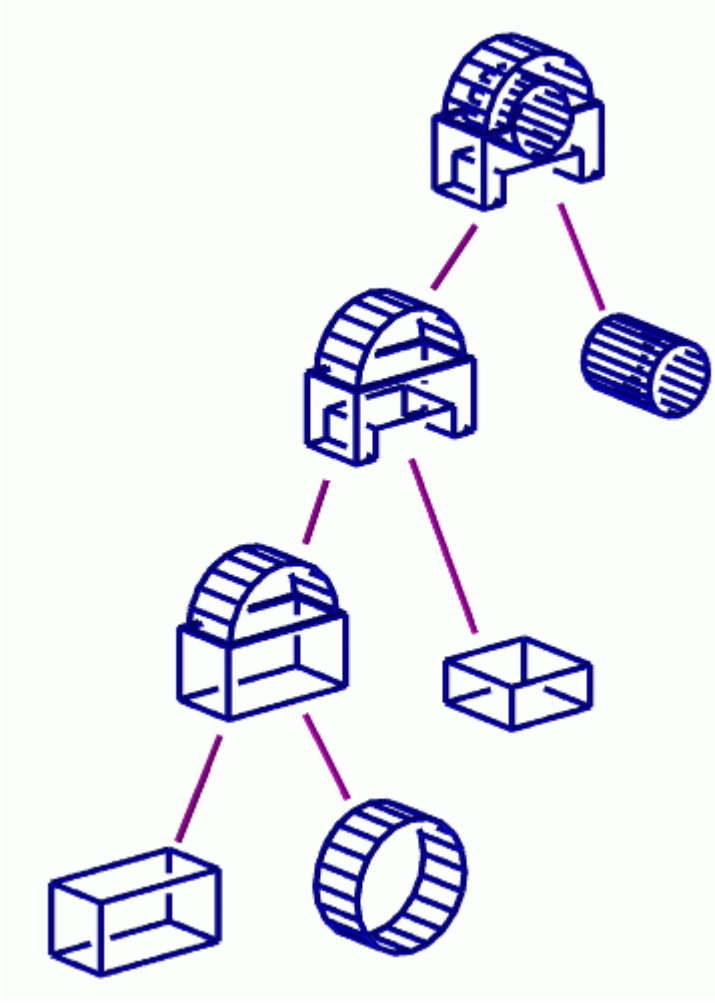


基本立体：　立方体，円柱，多角柱，円錐，球

集合演算：　和集合，積集合，差集合，補集合

CSGの例



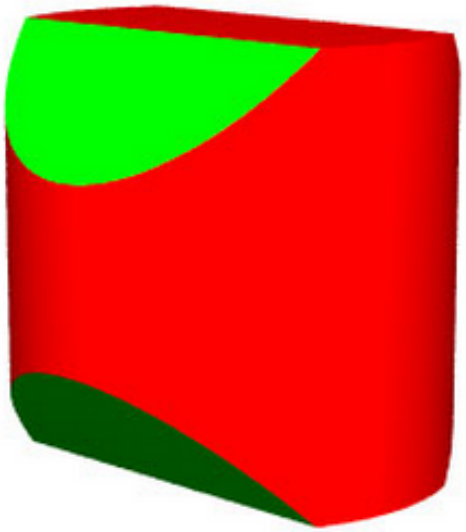


CSGの例：POVーRayのシーンファイルの記述

シーンファイル

```
#include "colors.inc"
#include "shapes.inc"

camera{
  location <0,2,-10>
  look_at <0,0,0>
  angle 30
}

light_source{<-10,10,-10> color White}
light_source{<10,10,-10> color White}
light_source{<0,2,-10> color White}

intersection{
  object{
    Disk_Y
    pigment{ color Red}
  }
  object{
    Cube
    rotate 45*x
    rotate 45*y
    pigment{ color Green}
  }
}

background{color White}
```

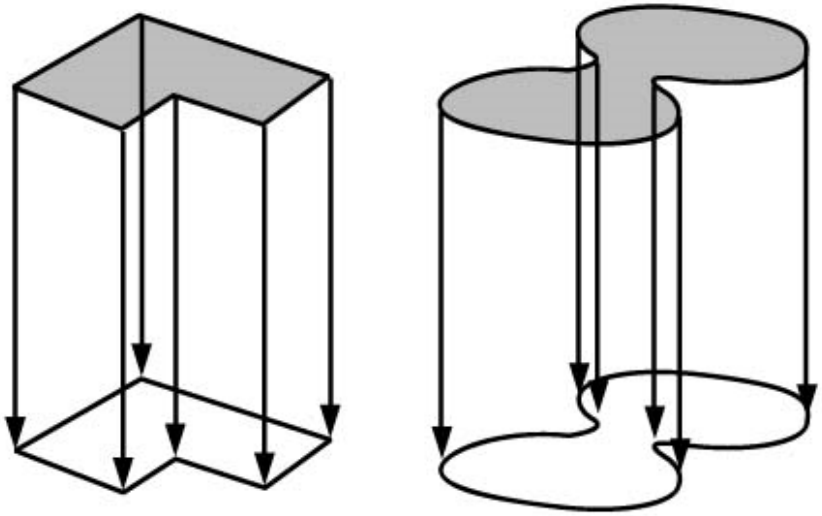
画像例



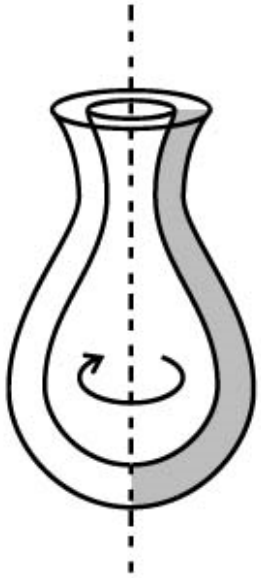
http://www.sato-lab.jp/ml2006/8th.html

スイープ表現

- 平面図形を一定方向に移動したときの軌跡で立体を表現
- 局所変形との組み合わせで，様々な形状を表現可能
- 平行移動スイープ，回転移動スイープ



〈a〉平行移動スイープ



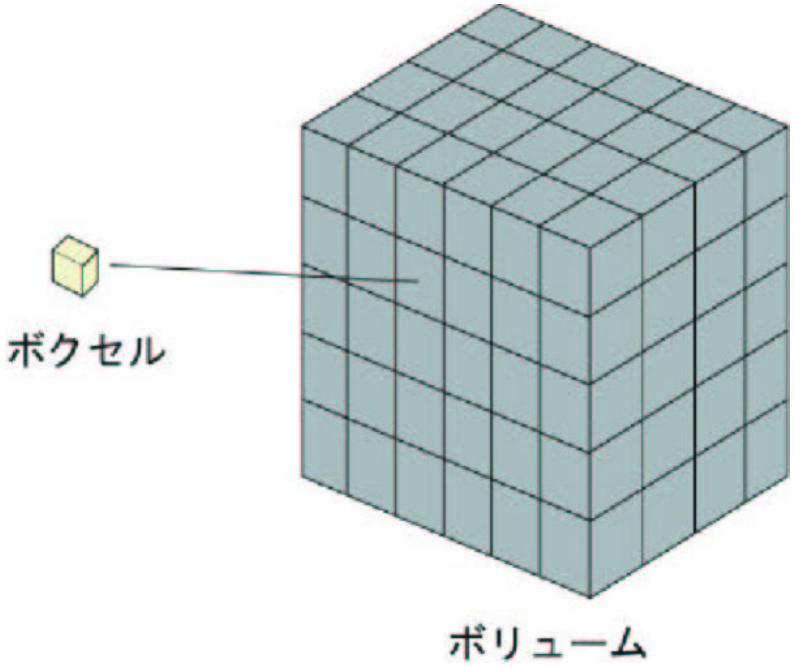
〈b〉回転移動スイープ

他の表現方法

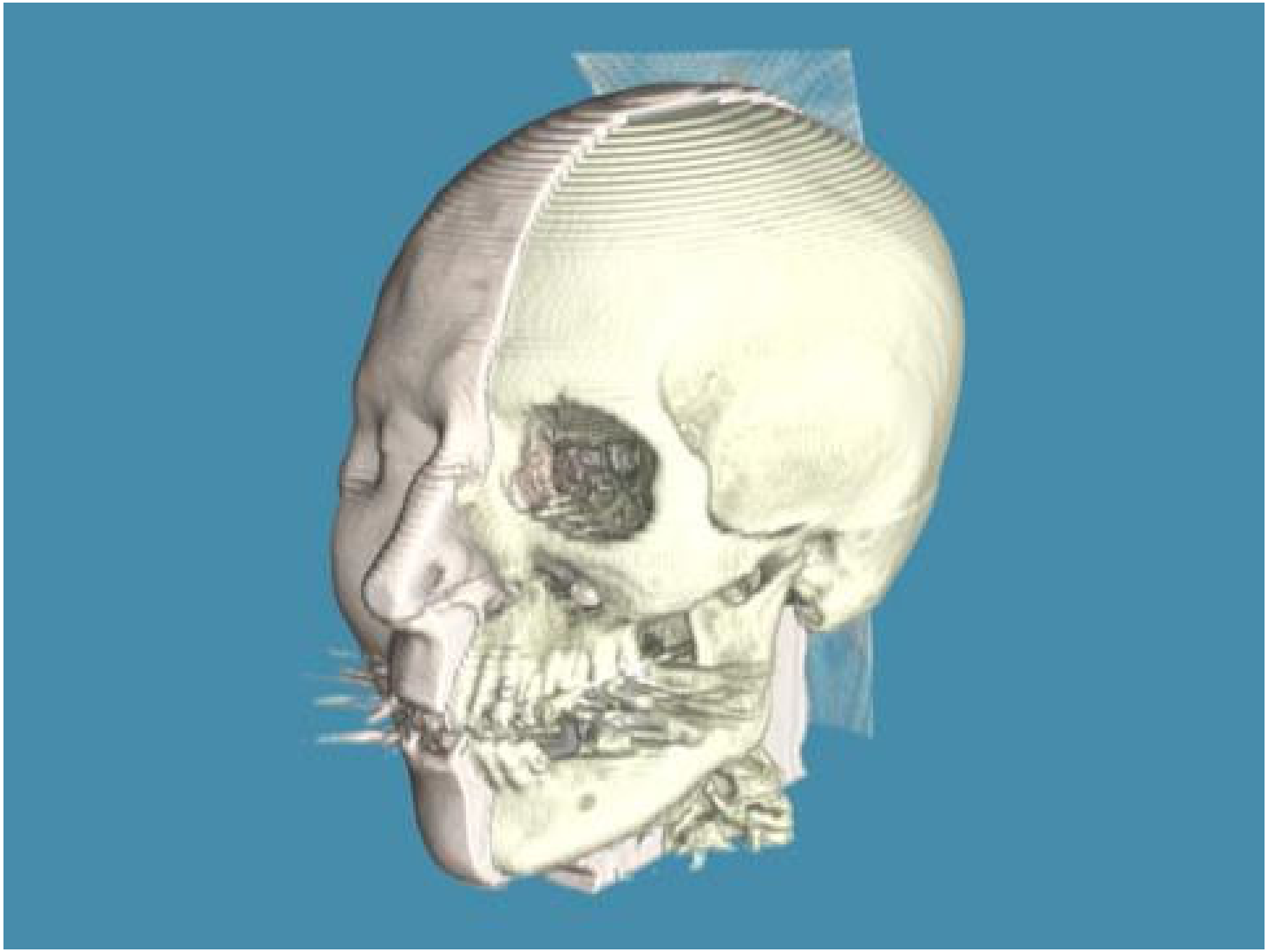
- ボリューム表現
- 八分木表現
- フラクタル図形
- メタボール
- パーティクル

ボリューム表現

- 立体を3次元の格子状の小立方体（ボクセル）の集合で表す
- 長所
  - データ構造が単純，集合演算が容易
  - 人工的な物体より，自然界の不規則な形状表現に適する
- 短所
  - データ量が膨大，操作に手間がかかる



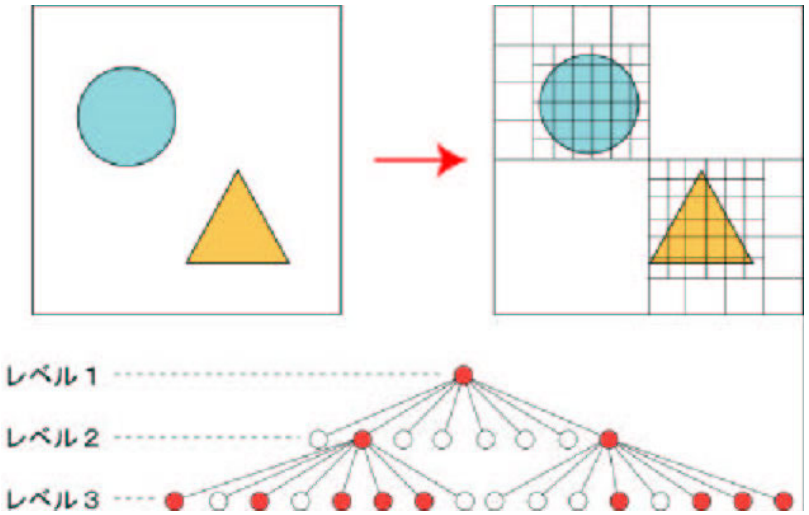
ボリュームレンダリング



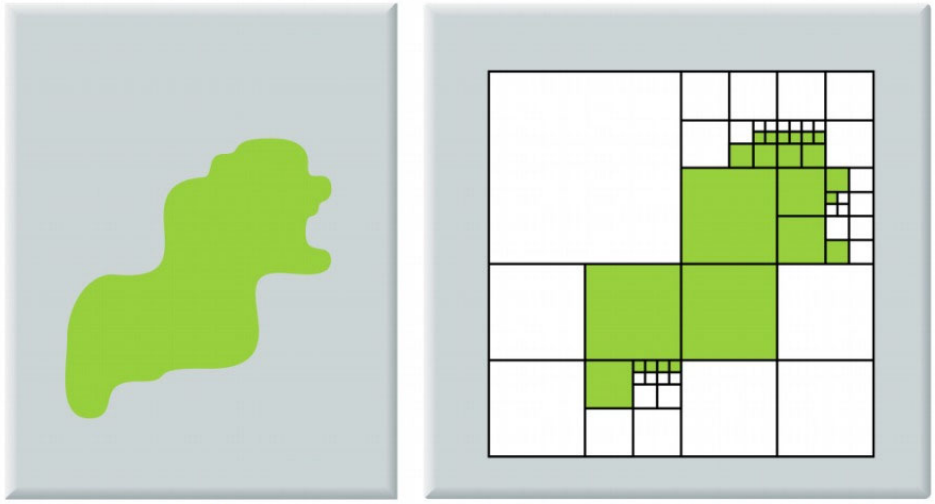
ボリュームレンダリング

八分木表現

- ボクセルを階層的に，木構造で生成
- 物体が存在するボクセルのみ細かく分割
- 空間量（メモリ）も少なくて済み，高速



■図3.49——2次元図形の四分木による表現

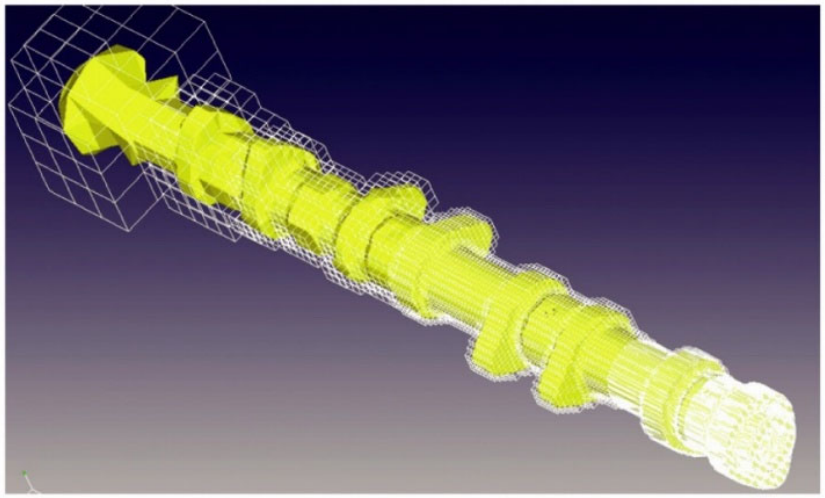


[a] 2次元図形の例

[b] 四分木による表現（領域分割5段目までの例）

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

■図3.50——カムシャフトのCADデータの八分木表現

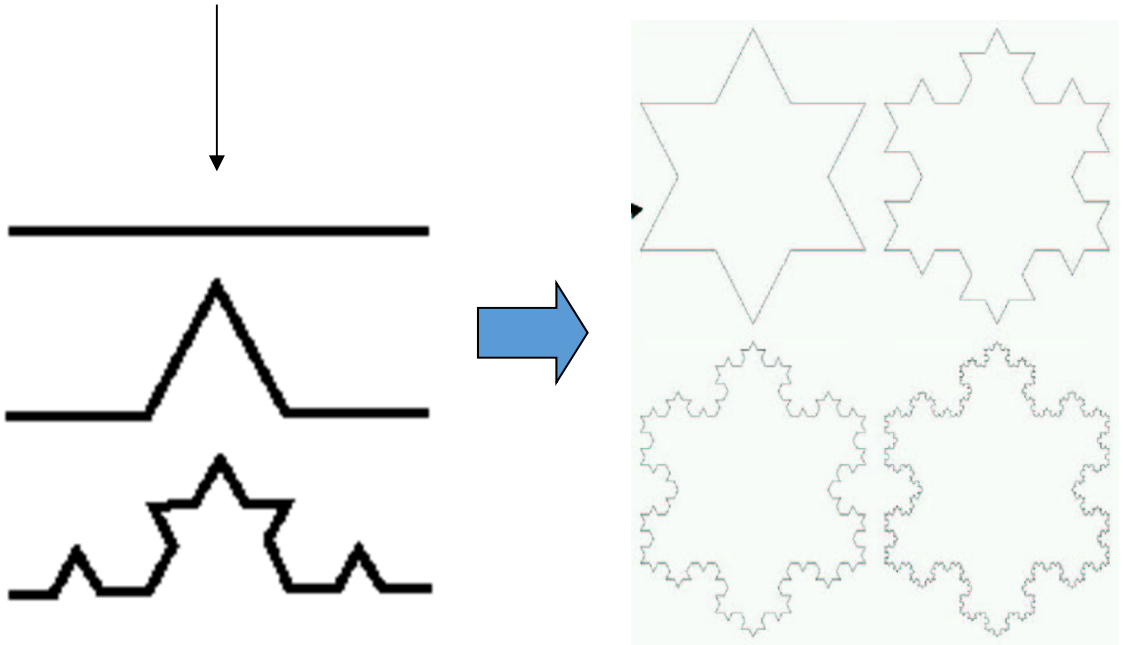


（入力CADデータ：©独立行政法人産業技術総合研究所ものつくり先端技術センター，
八分木化：©独立行政法人理化学研究所ものつくり情報技術統合化研究プログラムによるボリュームCAD）
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

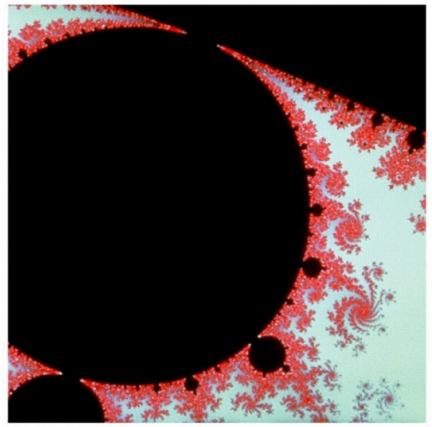
3次元画像の八分木表現

フラクタル図形

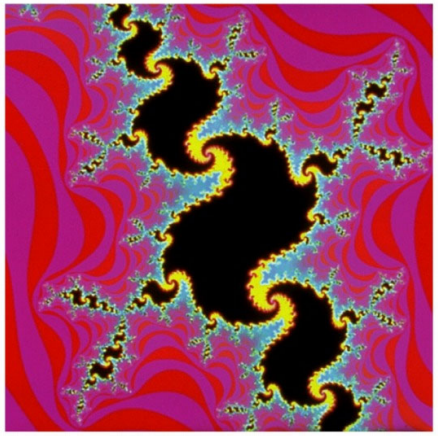
- 全体形状がその形状の各部分にも現れるような形状.
- 自己相似形状, 再帰構造
  - 例）コッホ曲線，ジュリア集合，マンデブロ集合など



■図3.52——マンデルブロ集合

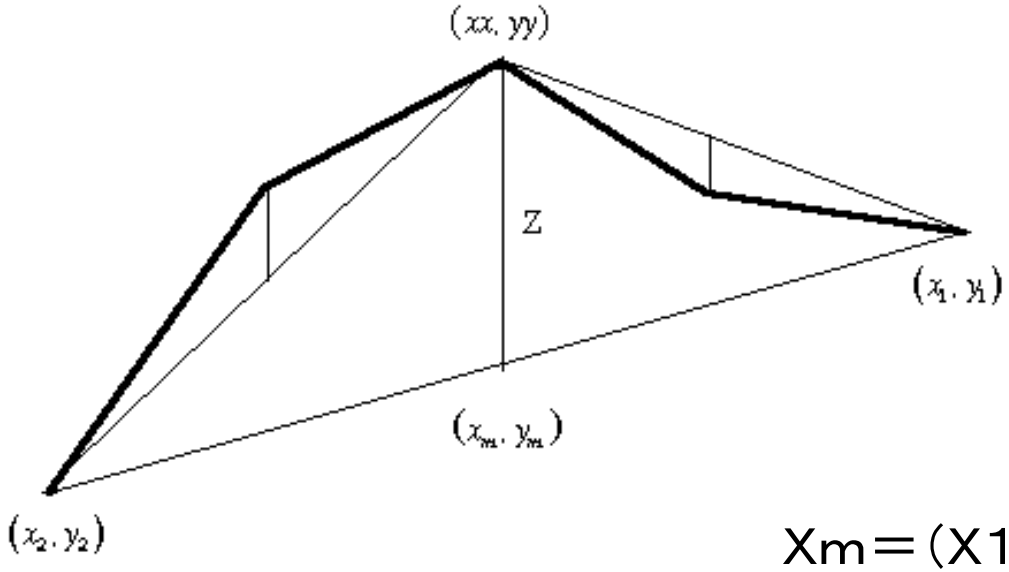


■図3.53——ジュリア集合



中点変位法

- 中点に起伏量Ｚを加える操作を繰り返す
- 起伏量zは，正規分布に従う乱数によって決定



Xm＝(X1＋X2)／2 、Ym＝(Y1＋Y2)／2
XX ＝ Xm ＋ Z、YY ＝ Ym ＋ Z

中点変位法による画像生成

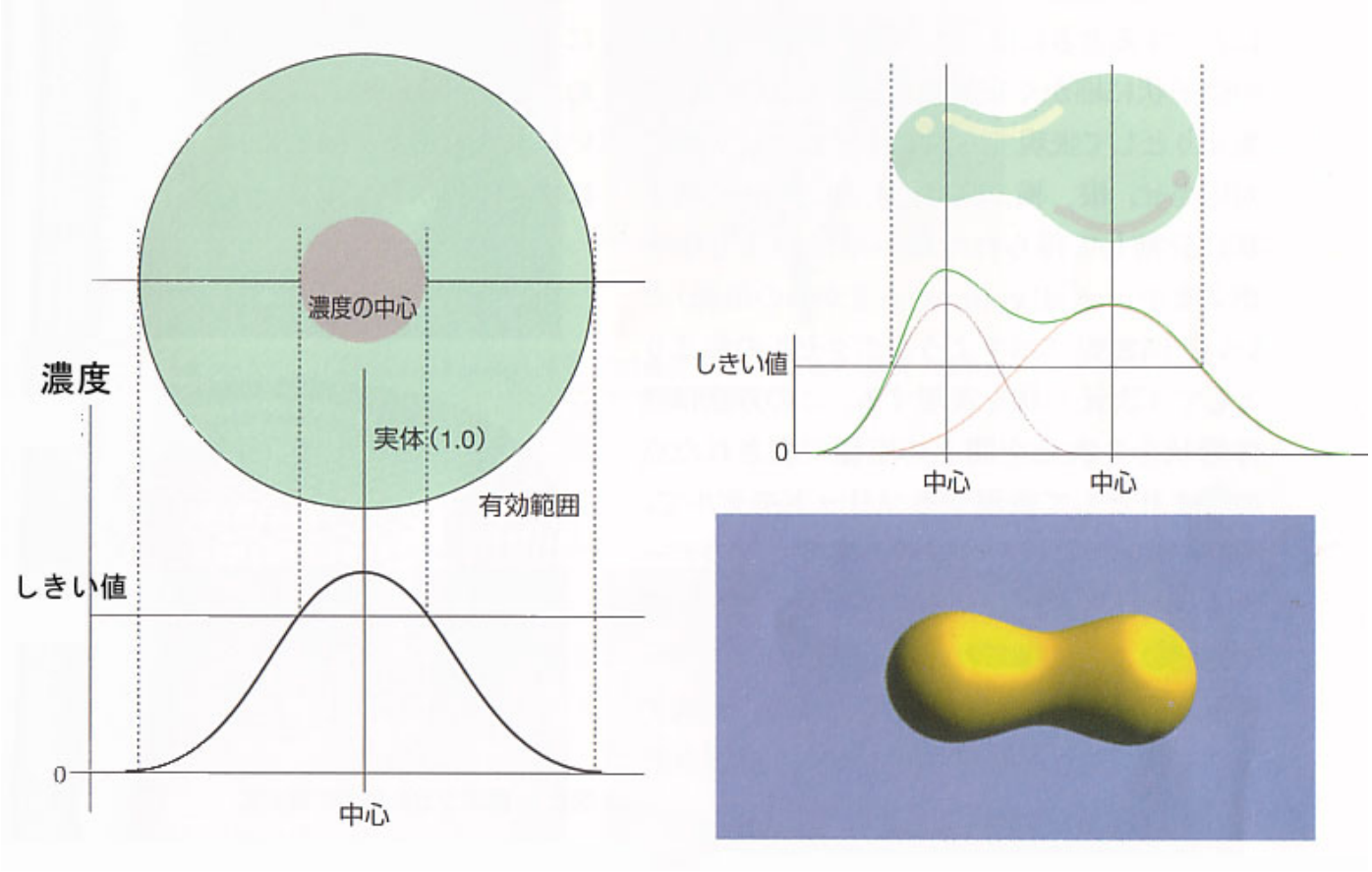
■図3.57——中点変位法による山岳形状の生成

（提供：北陸先端科学技術大学院大学宮田研究室）

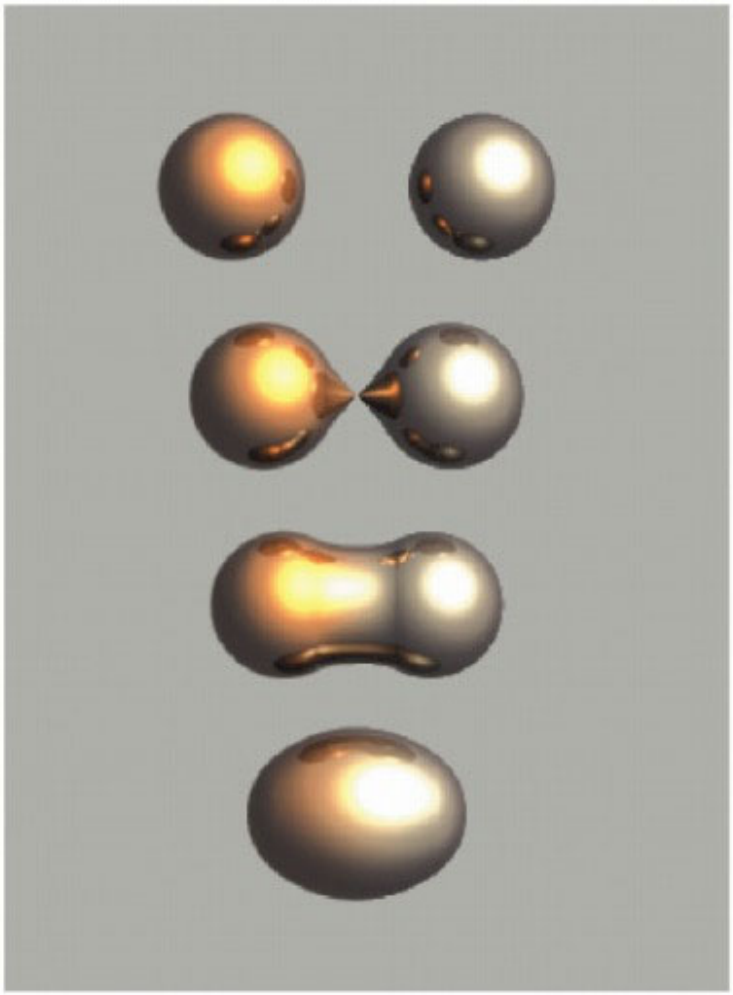
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

メタボール

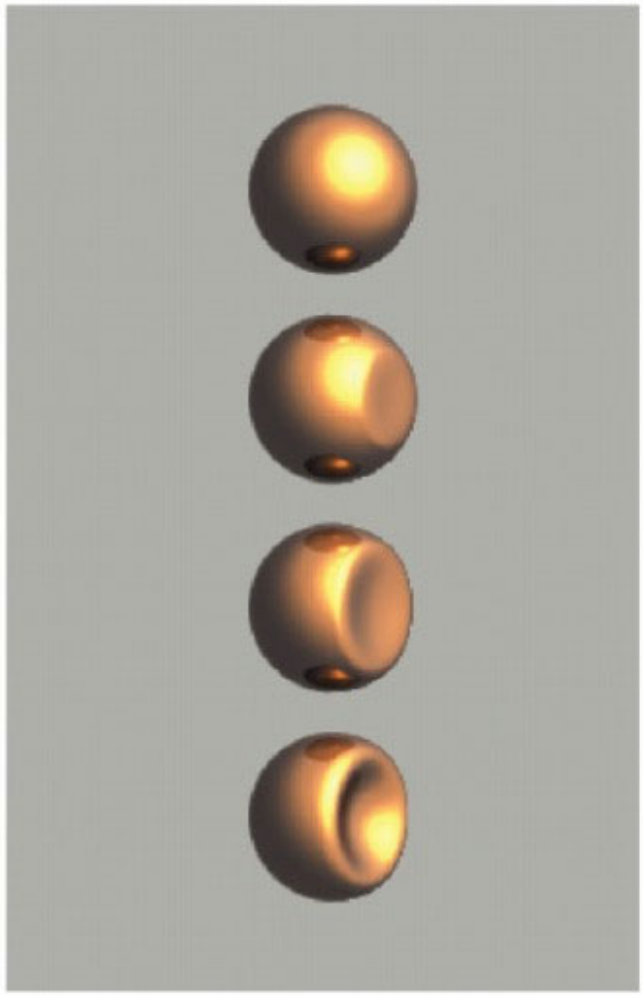
- 立体を球の集まりで表現
- 距離とともに減衰する濃度（関数）を定義し，その重ね合わせで形状を表現



■図3.61——メタボールによる形状生成の例



[a] メタボールの融合



[b] 負のメタボールによる変形

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

パーティクル

- 形状が不定で，明確な表面が存在しない物体
  - 樹木，炎，滝，雲　などの自然物
- 一定の規則に従って生成した多数の粒子で表現
  - 粒子（パーティクル）の生成，移動，消滅，衝突　の物理的規則が必要

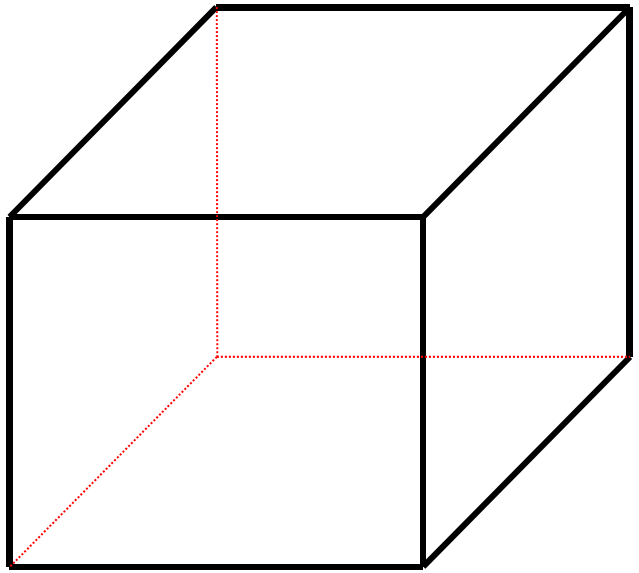
パーティクルで表現した
炎と煙

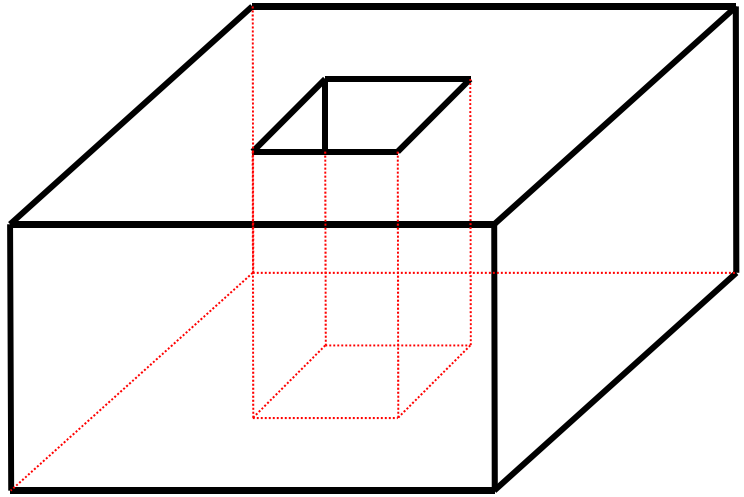
オイラー・ポアンカレの公式

- オイラー・ポアンカレの公式とは
  - ソリッドモデルに不変な構成要素の関係式

    v - e + f - h = 2 (m - g)

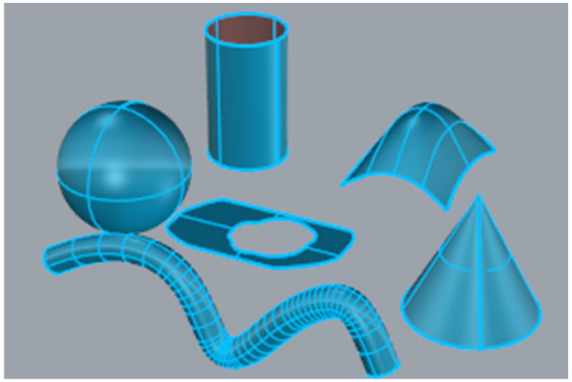
    頂点数 - 稜線数 + 面数 - 面内ループ数 = 2 （物体数 - 貫通穴数）　　どちらでも成り立つか？



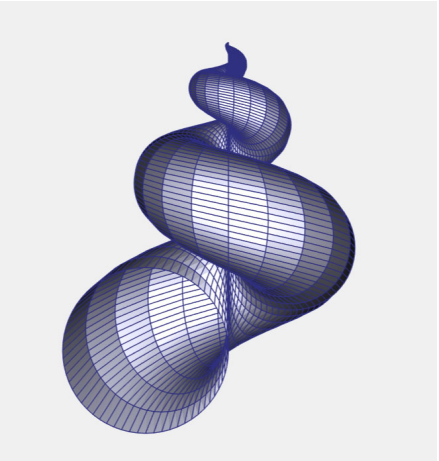


パラメトリック表現

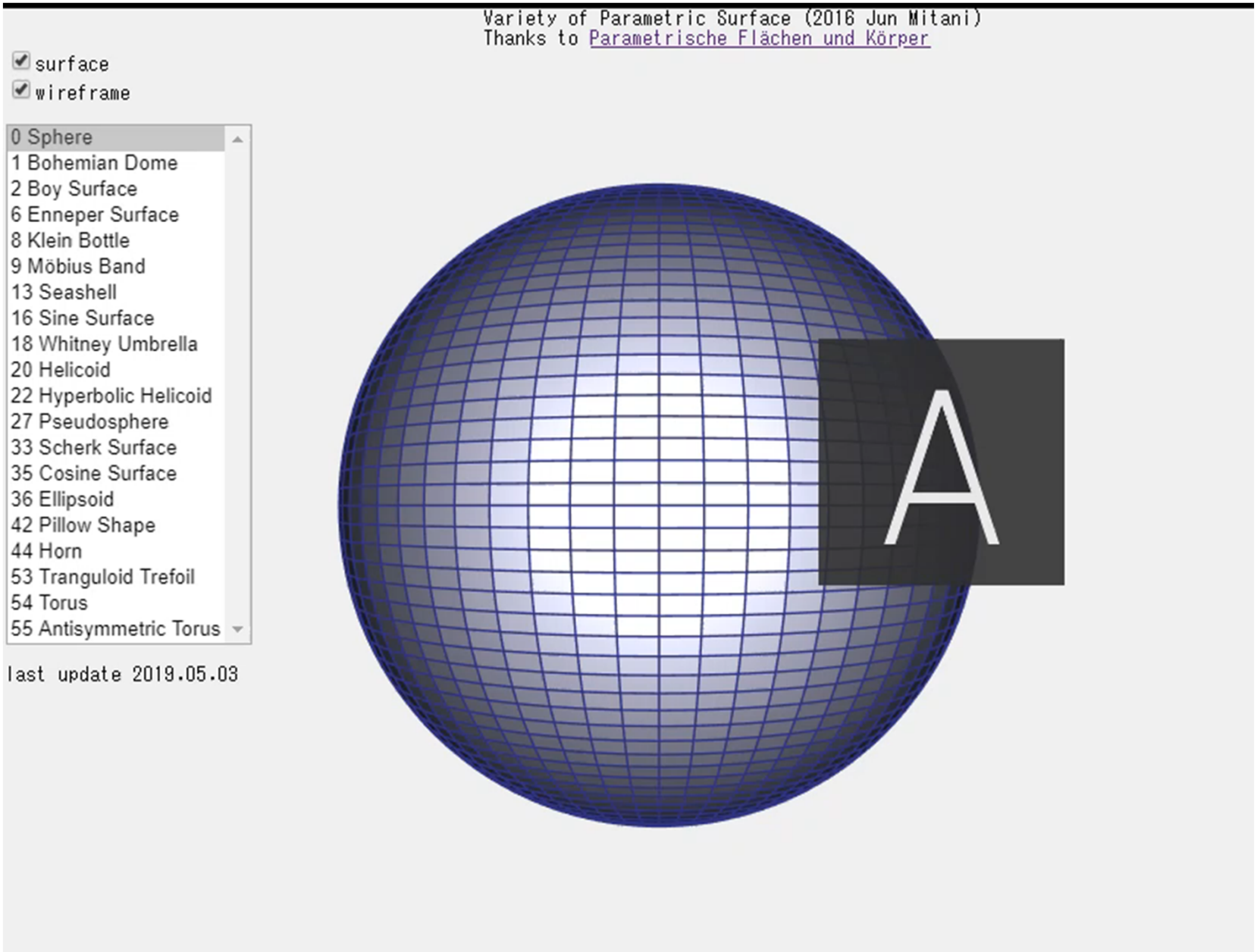
- 曲面上の座標x,y,zそれぞれを2変数(u,v)の関数で表す。



- 正確な形状定義
- 滑らかな曲面
- 数学的な解析ができる
- データ量が少ない
- 自由形状を表現する場合には複数の曲面を組み合わせる



$$\mathbf{f} : u, v \mapsto \begin{pmatrix} a(1-\frac{v}{2})\cos(nv\pi)(1+\cos(u\pi)) + c\cos(nv\pi) \\ a(1-\frac{v}{2})\sin(nv\pi)(1+\cos(u\pi)) + c\sin(nv\pi) \\ \frac{v}{2}b + a(1-\frac{v}{2})\sin(u\pi) \end{pmatrix}$$

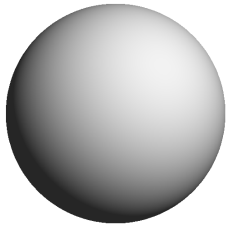


http://mitani.cs.tsukuba.ac.jp/ja/software/js/parametric_surface/index.html

陰関数表現

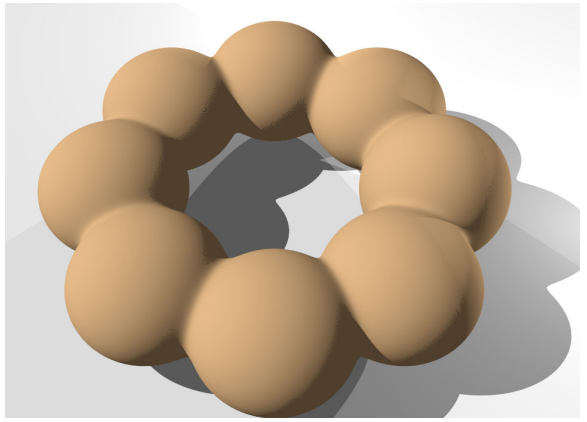
$$\mathbf{F}(x, y, z) = 0$$

の形式の数式で表現する



$$\mathbf{F}(x, y, z) = x^2 + y^2 + z^2 - r^2 = 0$$

・正確な形状定義
・滑らかな曲面
・データ量が少ない
・立体の内側（＜０）と外側（＞０）を定義できる

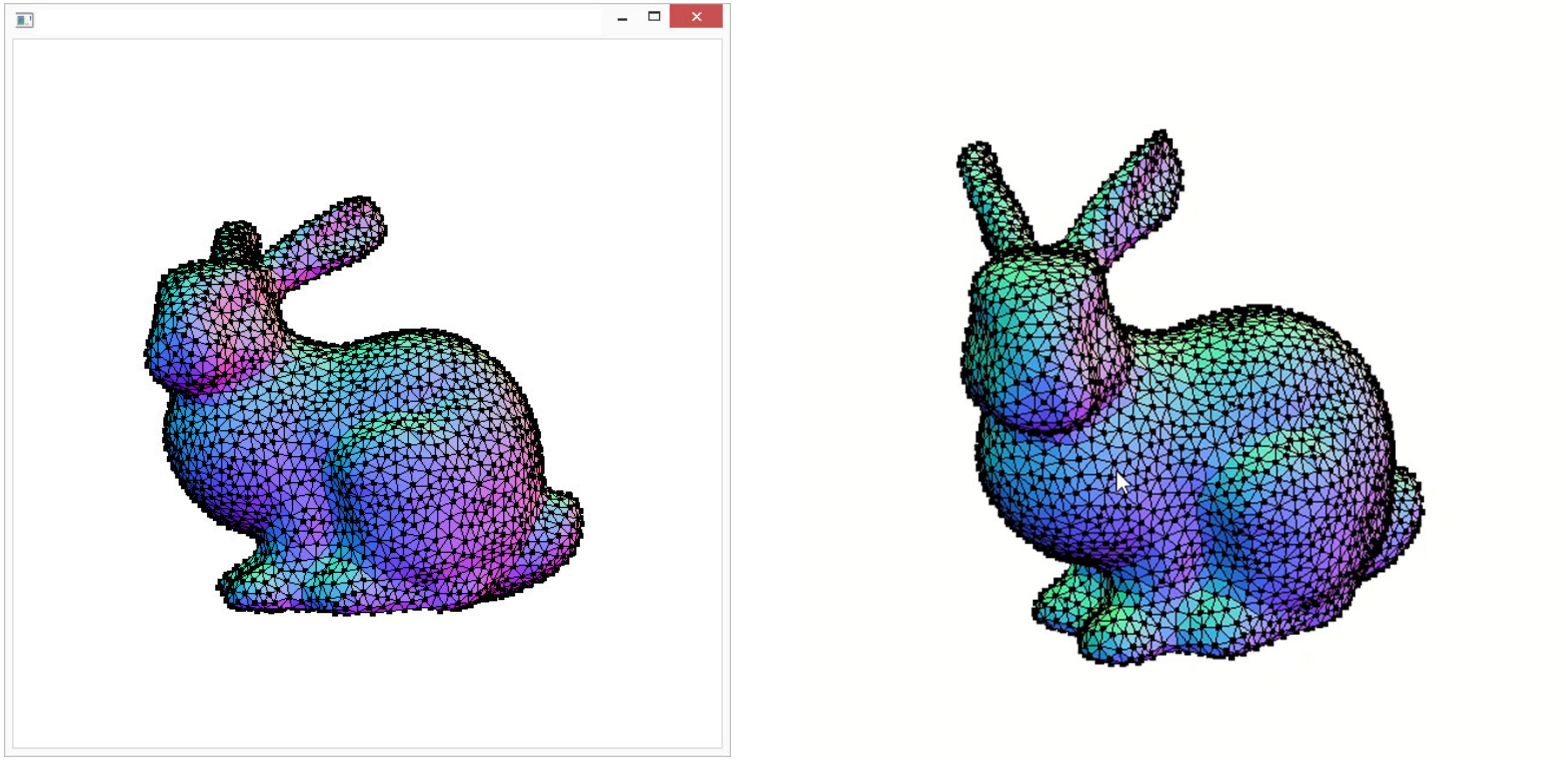


$$\sum_{k=0}^{7} \exp\left\{-0.7\left(\left(x - 6.75\cos\frac{k\pi}{4}\right)^2 + \left(y - 6.75\sin\frac{k\pi}{4}\right)^2 + 1.2z^2\right)\right\} - 0.001 = 0$$

課題説明

OBJViewの動作確認

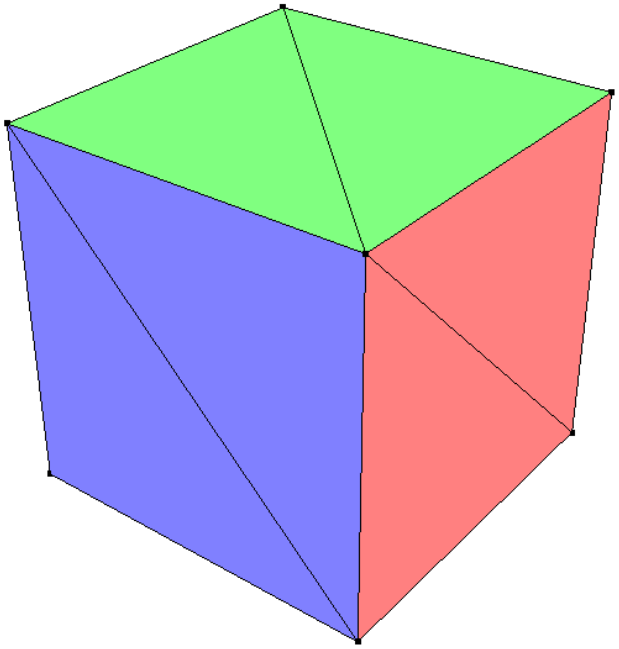
- OBJファイルを表示するアプリケーション
  - Windows, Mac で動作するバイナリコードとソースコード付き



立方体モデルデータの作成

- OBJ形式で立体の形を記述してみる（テキストファイル）
- 記述した形が意図したものになっているか確認する

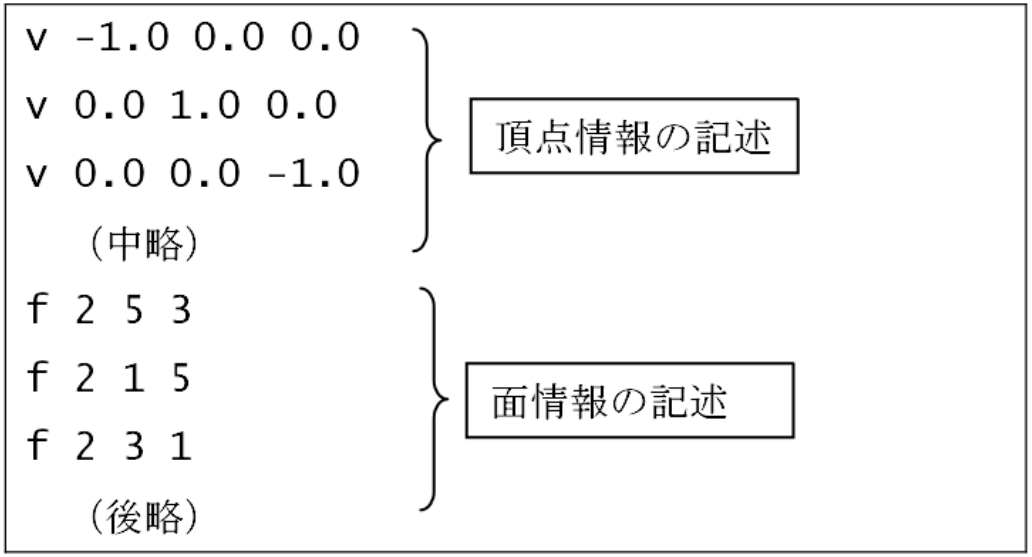
```
v 0.0 0.0 100.0
…
…
…
f 1 2 3
f 1 3 4
…
…
```



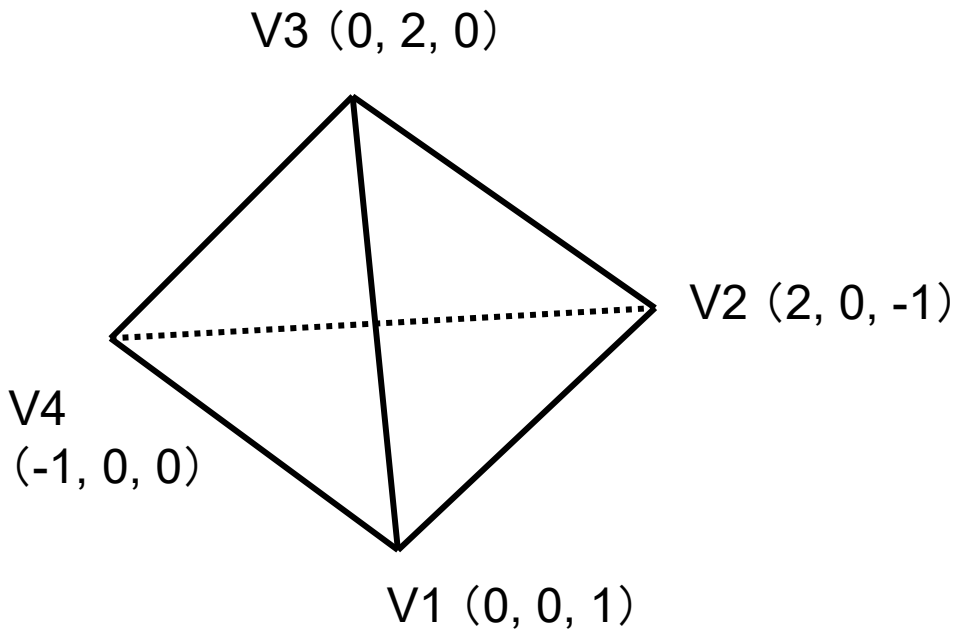
面の向きが裏になっている場合はグレーで表示される

OBJ形式とは

- Wavefront社の策定したフォーマット
  - 数あるフォーマットの1つに過ぎない。これ以外にも様々なフォーマットがある。一長一短。
- キーワード「v」の後に頂点の座標を記す
- キーワード「f」の後に面を構成する頂点番号を記す
- 1行1エントリ



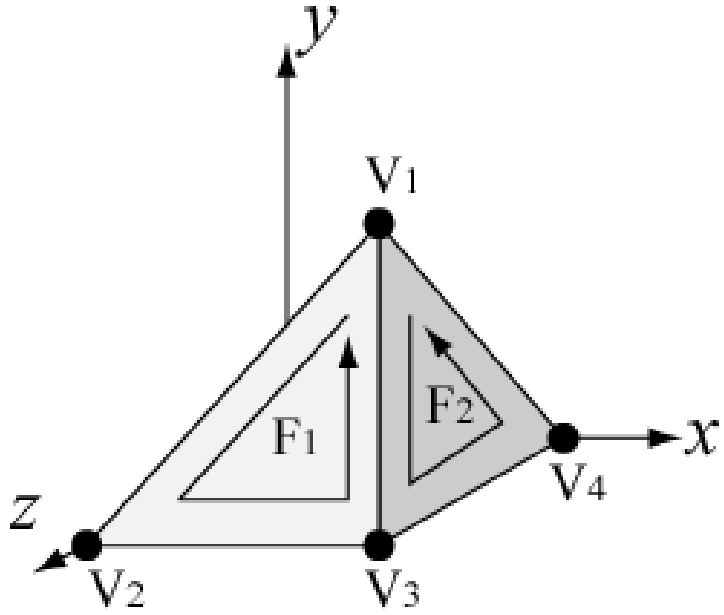
例




```
v 0 0 1
v 2 0 -1
v 0 2 0
v -1 0 0
f 3 1 2
f 3 4 1
f 3 2 4
f 1 4 2
```

頂点番号は1から始まる

座標系と面の向き



反時計回りに見える向きが表面

OBJViewでは、裏面（内側）は灰色で表示される

パラメトリック曲面の作成実験

格子状に配置された頂点座標を2次元配列で保持

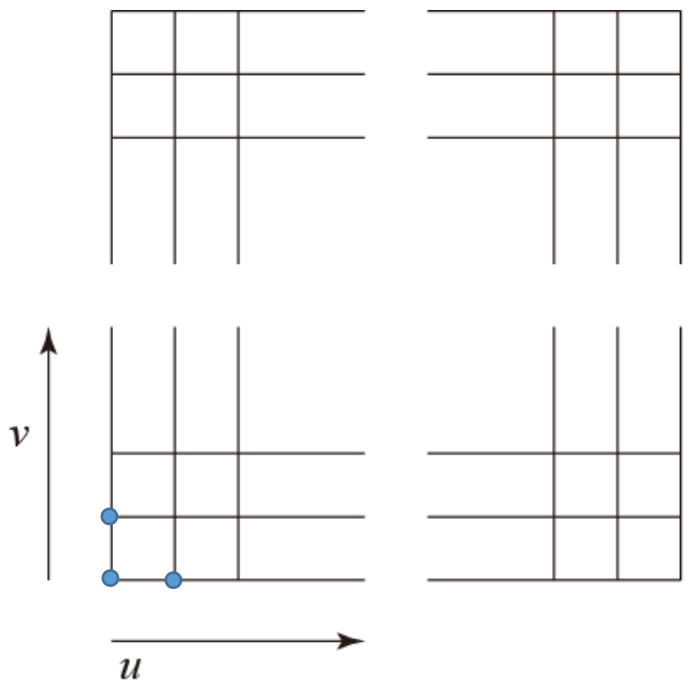
```
double x[NUM_U+1][NUM_V+1]; // x 座標
double y[NUM_U+1][NUM_V+1]; // y 座標
double z[NUM_U+1][NUM_V+1]; // z 座標
```

頂点位置の計算

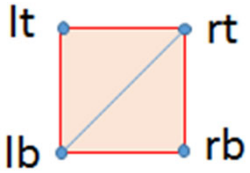
```
for(int i = 0; i < NUM_U+1; i++) {
  for(int j = 0; j < NUM_V+1; j++) {
    // u と v の値を 0.0 ～ 1.0 に正規化する
    double u = 1.0 / NUM_U * i;
    double v = 1.0 / NUM_V * j;

    // 座標値の設定
    x[i][j] = uとvで定義される
    y[i][j] = uとvで定義される
    z[i][j] = uとvで定義される
  }
}
```

頂点インデックスの計算

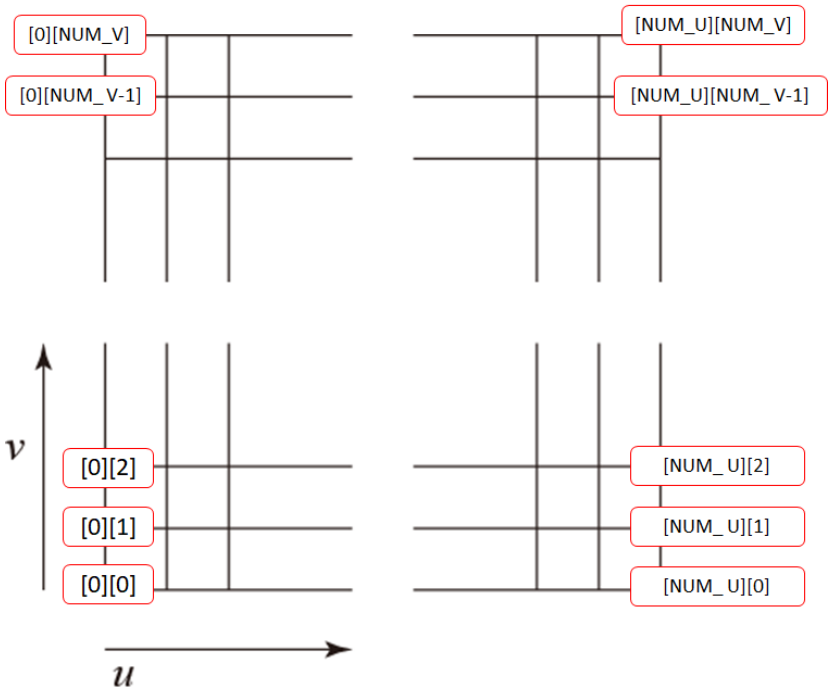
```
for(int i = 0; i < NUM_U; i++) {
  for(int j = 0; j < NUM_V; j++) {
    int lb_index = 左下の頂点番号の計算式
    int lt_index =  左上の頂点番号の計算式
    int rb_index = 右下の頂点番号の計算式
    int rt_index =  右上の頂点番号の計算式

     // 三角形を構成する頂点番号を出力
     fprintf(fout, "f %d %d %d¥n", lb_index, rt_index, lt_index);
     fprintf(fout, "f %d %d %d¥n", lb_index, rb_index, rt_index);
  }
}
```



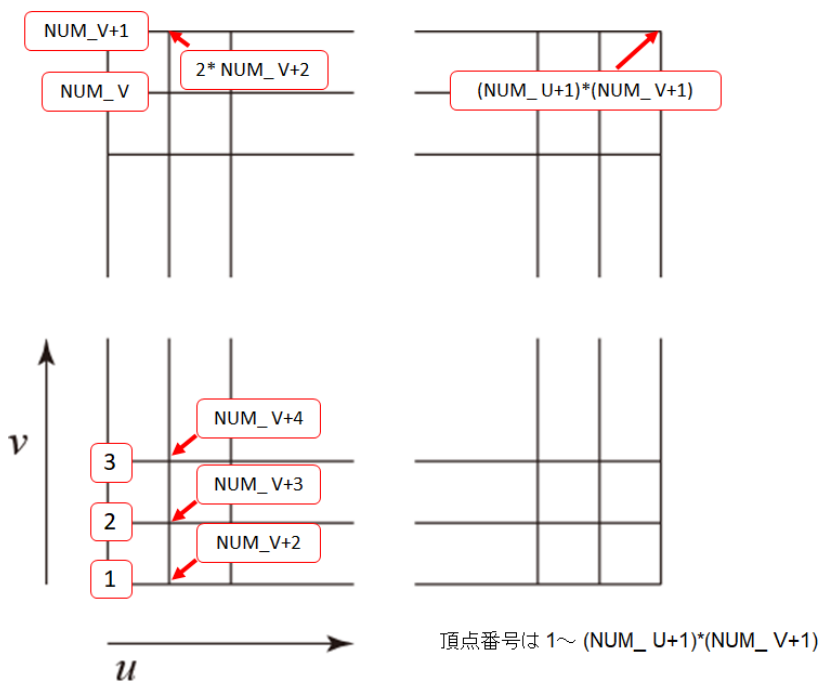
1つの四角形領域は2つの三角形で表現。
それぞれ、{lb, rb, rt}および{lb, rt, lt}
に位置する頂点を参照する。



サンプルコードのイメージ（2次元配列のインデックス）



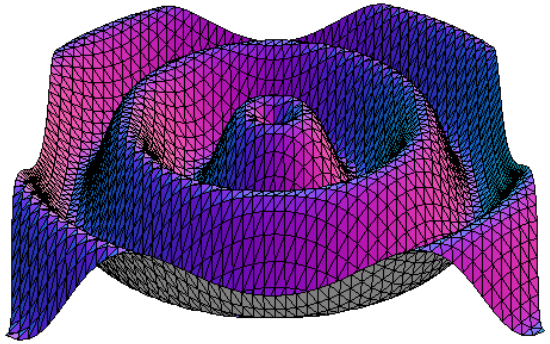
サンプルコードのイメージ（頂点番号）



頂点番号は 1～ (NUM_U+1)*(NUM_V+1)

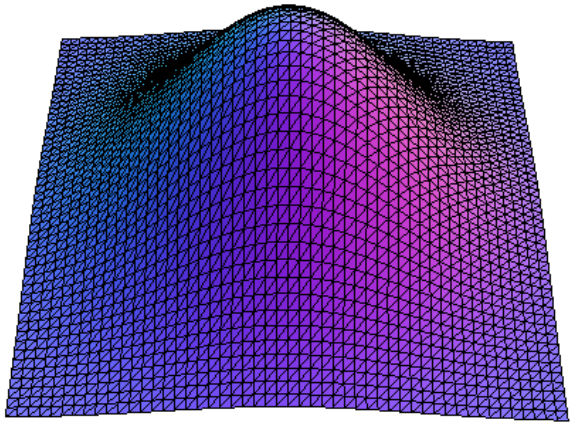
数式で表現されるパラメトリック曲面

波紋



$$\begin{cases} x = u \\ y = v \\ z = \dfrac{1}{10}\sin(8\sqrt{(u-1/2)^2+(v-1/2)^2}\,\pi) \end{cases} \quad (0 \le u \le 1, 0 \le v \le 1)$$

ガウス関数

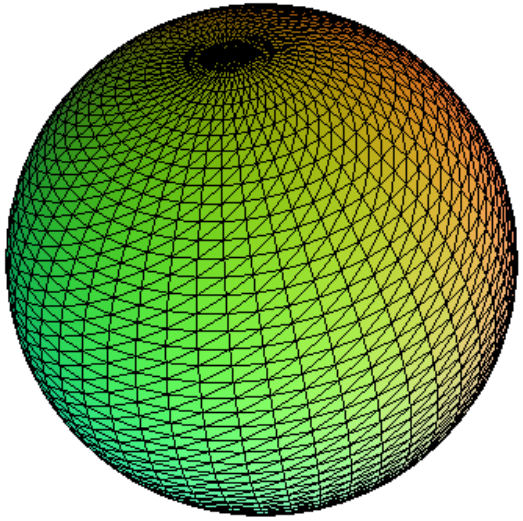


$$\begin{cases} x = u \\ y = v \\ z = \dfrac{1}{2}\exp\left\{-\dfrac{(u-1/2)^2+(v-1/2)^2}{0.1}\right\} \end{cases} \quad (0 \le u \le 1, 0 \le v \le 1)$$

※ 頂点位置を計算する個所だけ変更すればよい

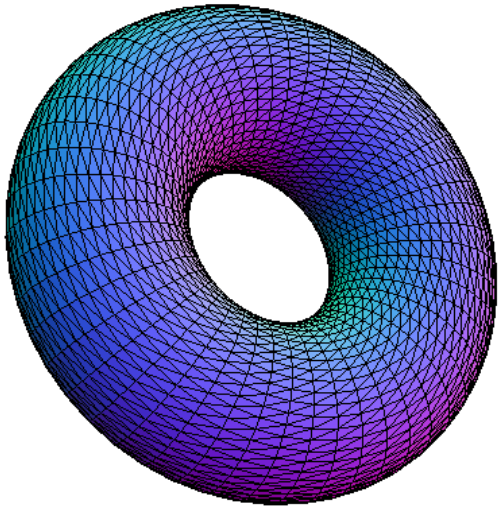
数式で表現されるパラメトリック曲面

球



$$\begin{cases} x = \cos(u)\cos(v) \\ y = \sin(u)\cos(v) \\ z = \sin(v) \end{cases} \quad (0 \le u \le 2\pi, -\frac{\pi}{2} \le v \le \frac{\pi}{2})$$

トーラス

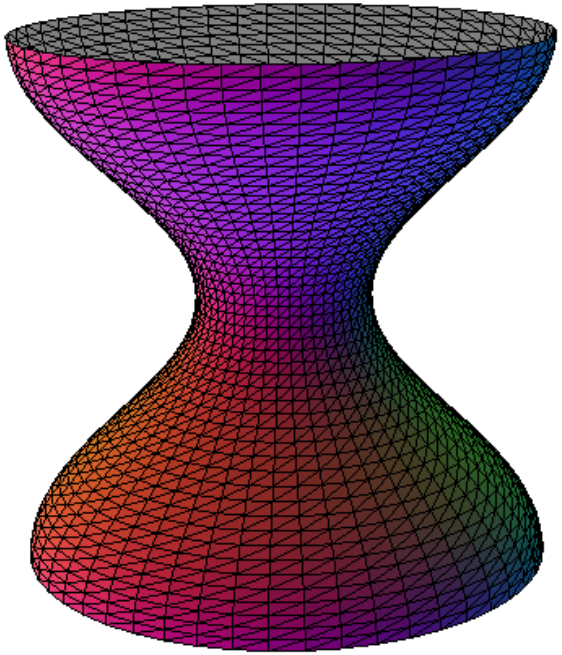


数式を自分で考えてみよう

※ 頂点位置を計算する個所だけ変更すればよい

自由形状

ベジェ曲線の作図と組み合わせて、様々な曲面を作ってみよう



曲線を回転してできる回転対称な図形

先輩たちの作例