

# 面ベース形状モデルにおける演算操作と曲面の導入

4 N - 2

東京電機大学 ○市村 高志 齊藤 剛

## 1 はじめに

形状モデルにおける立体間のブール演算操作には、高速性、安定性、そして堅牢性を保障する必要がある。

ブール演算操作を用いて形状を生成する場合、設計者は、立体を面一致、稜線一致、頂点一致などの位置関係で配置し、形状生成を行なうことが多い。このような配置は、通常「特異な場合」として扱われるが、これらは設計者の意図であり、形状モデルとして、このような拘束の下での演算が確実に行なえることが必要である。このような状況では、計算誤差のため正しく形状生成が行なえない場合があり、その解決法について種々の研究が行なわれている<sup>2)</sup>。

本稿では、面ベース形状モデルでのブール演算操作の実装、特に前述の拘束を持つ配置について述べ、これらの演算が規則的な変換処理によって簡潔に実現できることを示す。また、製品形状には欠かせない、フィレット面の面ベース形状モデルへの導入とその応用としての2次曲面の表現についても述べる。

## 2 面ベース形状モデルのデータ構造

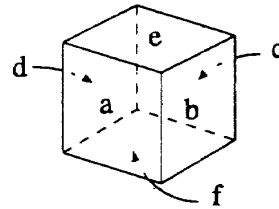
本節では、面ベース形状モデルのデータ構造の概要を述べる。詳細は、文献1)等を参照されたい。

本モデルにおける立体表現の基本は、面隣接表である。面隣接表は、面の隣接関係を表す隣接面ループと、面の幾何情報からなる。幾何情報は、各面の単位法線ベクトルと面上の一点である。

隣接面ループは、立体を構成するすべての面に識別可能な名前を付け、各面に隣接する面群を左回りのリストで表したものである。頂点や稜線の情報は、必要に応じて面隣接表から求めることができる。図1は、6面体を表す面隣接表の面ループの例である。

## 3 面ベース形状モデルでの演算操作

本節では、面ベース形状モデルでの立体間のブール演算操作の手順と例を述べる。



face	face loop
a	b-e-d-f-
b	c-e-a-f-
c	d-e-b-f-
d	a-e-c-f-
e	b-c-d-a-
f	b-a-d-c-

図1. 6面体と隣接面ループ

### 3.1 演算処理の概要

本モデルにおけるブール演算操作は、演算により生成される立体の各面に対して、位相情報、すなわち、隣接面ループを再構成することにより実現できる。

本モデルの特長として、「頂点表記 (a, b, c) は面 a, b, c の交点である」のように、面の位相関係を直接的に表すことがある。この特長を利用し、ある面に関して「演算により干渉してできる頂点」と「演算後も存続する頂点」を検出することにより、その面の新しい隣接面ループを求める。

### 3.2 演算処理の手順

図2に示す立体Aは面 {0, 1, 2, 3, 4, 5} から、立体Bは面 {a, b, c, d, e, f} からなる6面体である。これらの立体の和を取る演算操作を例に取り上げ、頂点の検出法、ならびに位相情報を再構成する手順を述べる。

始めに、一致する面の検出を行なう。図2において、面4と面eは一致している。この面一致が設計者の意図によるものであるならば、元の立体の位相情報が異なる限り、一致する面として取り扱う。設計者の意図ではなく偶然一致してしまう面については、操作指定時に自動的に検出し、同様に扱う。

次に、各面の干渉稜線の頂点、存続する頂点のリストを求める。例えば、面4については次のようになる。

1. 稜線 (4, 0) は立体Bの面によって干渉を受けない。従って、頂点 {(4, 3, 0), (4, 0, 1)} が得られる。
2. 稜線 (4, 1) からは (4, 0, 1) と、面 a との交点 (4, 1, a) が得られ、{(4, 0, 1), (4, 1, a)} となる。面 e とは一致している為、交点は検出しない。

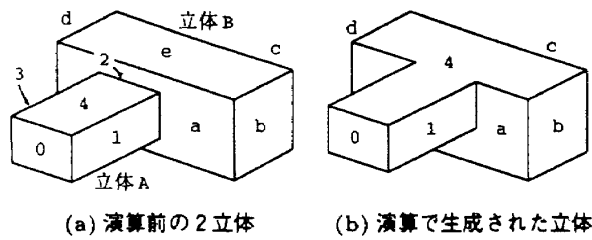


図2. 立体間の演算(和)の例

3. 稜線(4,2)は面eに包含される. 従って, 頂点は得られない.
4. 稜線(4,3)からは, 2.と同様の理由で $\{(4,a,3), (4,3,0)\}$ が得られる.

以上が, 立体Aから得られる面4の頂点である.

最後に位相情報の再構成を以下の方法で行なう. 検出した頂点リストより, 面4の隣接面ループは,  $\{0-1-2-3-\}$ から $\{0-1-a-\dots,-a-3-\}$ に置き換えられる. ここで, 立体Bの面eが面4と一致しているため, 面eからも面4のループが得られる. 面eの頂点を検出し, ループを求めると,  $\{a-3-\dots,-1-a-b-c-d-\}$ となる. この面4, および面eの各ループから面4の完全なループ $\{0-1-a-b-c-d-a-3-\}$ が得られる.

面4の隣接面ループを求め終わると, 他の面の隣接面ループのeを全て4に置き換え, 面4に対する位相情報の再構成は終了する. この様にして各面の位相情報を再構成すれば, 図2(b)の立体が生成される.

積演算では, 存続, 消滅する位相情報が逆転する. 差演算では, 一方の立体の面の単位法線ベクトル, 隣接面ループを反転させたものとの積を取る. 例えば図3(a)H型の立体Aと立体Bの上面, 前面, 後面の各面を一致させて差を取ると, 同図(b)となる.

#### 4 フィレット面の導入

本節では, 面ベース形状モデルへのフィレット面の導入について述べる.

頂点や凸稜線に丸めを施したフィレット面を取り扱うために, 本モデルでは平面をフィレットを表す曲面式に対応させる. 図4に示す例は, 図3(b)の立体の斜めにカットされた面を曲面式に対応させた例である. 平面に曲面式を対応させることで, フィレット面を他の面と同様に扱うことができる. さらに, これを応用し, このフィレット面を組み合わせることにより, 一般の2次曲面が表現できる. その例を図5に示す. この立体には8つのフィレット面を用いている.

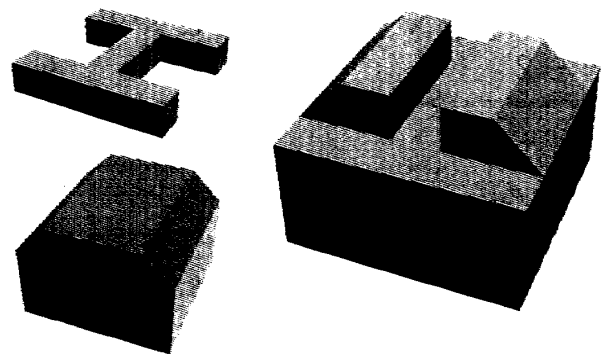


図3. 立体間の演算(差)の例

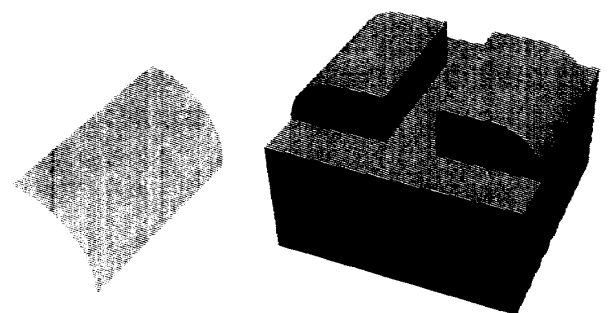


図4. フィレット面の生成

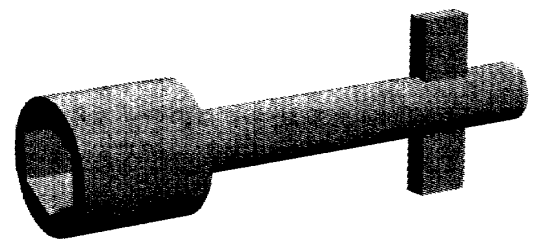


図5. 2次曲面を含む形状の例

#### 5 まとめ

本報告では, 面ベース形状モデルにおける立体のブール演算操作についてその処理手順を述べ, 拘束をもつ演算を指定できること, 演算が規則的な変換処理を用いて実現できることを示した. また, フィレット面の導入について述べ, フィレット面が平面と同一の手法で取り扱えることを示した.

今後の課題として, 形状情報からの立体生成経歴を抽出する方法について検討することがあげられる.

#### 参考文献

- 1) 市村, 斉藤: 面ベース形状モデルにおける形状特徴の生成, 情報処理学会第56回全国大会, 1998.3
- 2) 斉藤, 穂坂: 立体モデル間の干渉と演算, 第16回設計シンポジウム, 1998.7