Consistencyの実現を目的とする多面体モデラの開発

伊藤公俊　佐々木安彦
(東工大・総合理工)　(現在 第二精工舎)

1. まえがき

幾何モデル開発の試みは古くから行なわれてきているが[1]現在のところ汎用的に多くの応用に対して実用に耐えるシステムは発表されていない。幾何モデルには、可能な限り多くの幾何形状の類(Category)を格納・編集することが要求されるだけではなく，格納・編集における信頼性・正確さが保証されなくては実用に耐えないと考えられる。後者を、筆者らはモデラのConsistencyと呼んでいる。Consistencyを保持していないモデラは、実用に際して使用者に無用な負担を強いるばかりでなく、今後各種の応用プログラムが直接幾何モデルを操作して何らかの幾何情報を得る局面が増大すると考えられるが[2]、この様な時には事実上使用不可能になる。なぜなら、応用プログラムはそのプログラムの範囲において処理できる幾何操作を，モデラの内部処理の都合とは無関係に行なおうとするからである。

幾何モデルの開発には、大まかに二つの方向があると考えられる。すなわち、部品などについてそれの持つ機能・意味などは殆んど取り込まず純粋に幾何として処理する方向(幾何モデル本来の考え方)と，機能や意味，その他加工方法など幾何とは異る情報を積極的にモデル化および処理に利用する方向(目的モデル的な考え方)が考えられる。後者についてここでの意味は 機能や意味を単に属性として記憶するのではなく，これを積極的に幾何処理に利用するということで例えば円筒穴があるときこれが軸受として使用されドリル・リーマ加工されるのか、鋳造加工して流体流路として使用するか、または極端な例として円筒面をNCフライスなどで削り出すかでは保持するべき情報の量，質ともに全く異り，前二者の場合は複雑な円筒面処理はモデル内で不要であろう。しかし、この様な問題を処理することは魅力的ではあるが、問題解決(推論を含んで)や知識などの人工知能研究の対象が十分に成熟する必要があり現状では困難であろう。

本報で述べるのは 純粋に幾何のみを扱う部類であって，決められた幾何形状の類――ここでは多面体に限定される――の範囲では，完全な幾何の記憶と処理を幾分Brute Methodを用いて実現しようとするものである。基本的な手法として古典的な凸胞分割を用い，座標表現に任意桁長の有理形式を用いる。

以下，2節ではConsistencyについて述べ，3節以降では開発した多面体モデラについてその方法を示す。

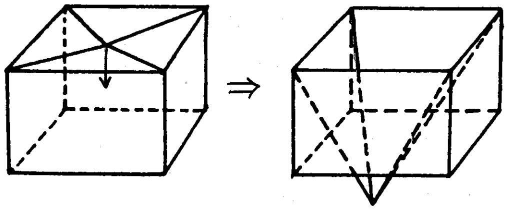
2. 幾何モデルにおけるConsistency

幾何モデル研究の分野でConsistencyという用語はRochester大学の研究者の報告に見られるが[3][4]本報で言うConsistencyはこれらと矛盾はしないが若干異る意味で用いている。以下にそれを述べる。

2−1. 処理におけるConsistency

モデラが幾何処理を行った時，幾何学的実体と対応しないデータ構造を生じることがある。また，対応するデータ構造が生成されたとしても幾何学的には演

算の定義も満足しないことがあり得る。この様な結果を生じないことを処理のConsistencyと呼ぶ。前者はモデルによっては検出が可能と考えることができるが後者はモデラ自身には検出することが出来ない。



(図-1)

モデラを設計する場合、まえがきに述べた様にConsistencyを保証する様なデータ構造と処理アルゴリズムを考える必要がある。すなわち、計算コストの面からもデータ構造自体が対応する幾何実体の存在を保証していることが理想であり，また処理アルゴリズムも数学的な裏付けを持つことが望ましい。この点で考えると幾つかのシステムで採用されているWinged Edge法による幾何の格納はエレガントではあるが、データ構造自身ではConsistencyを保証しておらず、(図－1)の様な変形が何らかの理由で行われる可能性がある時Consistencyを検証するためには多大の計算コストを要する。すなわち、この方法では各稜線は凸でも凹でも許容されているから図の様な矛盾は構成面相互の交叉をすべての組み合わせについて検査しなければ検出することができないと考えられる。さらに、この方式による幾何処理のための統一的アルゴリズムは自明ではないし筆者の知る限り公表されてもいないが、もしWinged Edge法を用いた幾何処理アルゴリズムが明確に示せない(プログラムに埋め込まれてしまっている)とすれば、幾何処理が外部的な定義を満足していることを保証する方法がないことになる。この点は処理ソフトウエアのバグとは峻別されるべき性質のもので、本質的にはConsistentではないと考えられる。

２－２．幾何表現における Consistency

処理における Consistency と同様にモデラが持つ表現そのもののそれを考えることができる。すなわち、ある幾何の表現は必ず対応するべき幾何実体の情報を完全に保持することが必要となる。この情報の中で最も基本となるのは座標値精度である。

モデラ内で平面として格納されている情報は必ず幾何学的に厳密な平面と対応していることが要求される。広く知られている様にＩＢＭ形式の３２bit浮動小数点数は精度の面でかなり不足であって、この形式で座標計算を行うとすると無視できない誤差を生ずる。もし，立体を構成する面を多角形の各頂点座標の系列として記憶したとする時、座標値に誤差を含めば平面性はなくなることになる。確に、加工を前提とする時は加工精度内に数値が収っていれば実質的な問題は全く生じないが、幾何モデルは形状編集操作を前提としているから誤差は形状編集中に累積されて加工精度の許容値を越えることになる。また、部品などの形状では一部に座標(寸法)精度を全く問題にしない部分がありNominalな精度を常に保つことが要求されるわけではないが、まえがきに述べた様に必要ならば精度を保証できなくてはならない。

さらに、ある立体から極く僅かずらして同じ形状を除去した時などに生ずるDangling Face・Dangling Edgeの問題も(5)座標値精度の問題が主である。非常に近接した値を浮動小数点数近似すれば誤差によって等しくなるか，演算によっては大小が逆転することすらあるから、数学の幾何公式をそのまま使用することは不可能

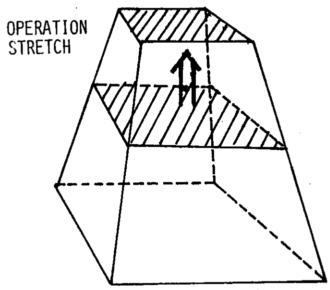
となる。同様に内外判定などの実数値(数学的意味の)→論理値の写像を含む計算も場合によって誤差の影響を強く受ける(すなわち論理値が逆転する)ことは明らかである。

数値誤差の問題は直交座標系では斜めを含む時に多く生じ円柱を多角柱近似する場合などは誤差を生じやすい。特にその様な図形の相貫を処理する際は大きな問題となる。

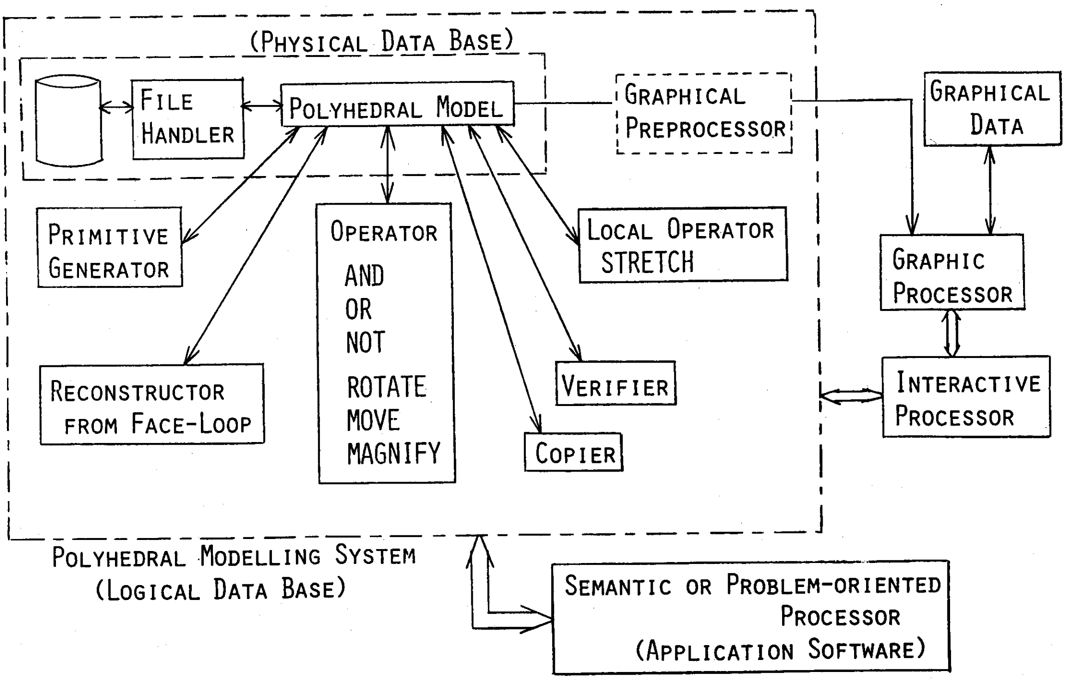
以上の議論では、機械工学に現われる寸法公差については無視して来たが、これは一つには、この問題をどの様に扱うかが確立されていないことによるが他方公称寸法や公差を幾何座標値としてモデル化できれば一応幾何モデラの必要条件は満足すると考えたことによる。

3. 多面体モデリング・システムの概要

開発した多面体モデラのシステム構成を(図－2)に示す。形状編集用としてAND, OR, NOTの論理オペレータと、局所処理を行うStretch(図－3)などがある。また主に図面認識システム(6)からの入力用に投影サイクルからの入力ルーチンを持っている。後述する様に本システムは凸胞分割によるモデリングを行っているから、このルーチンによって最初から非凸で与えられる立体に対処することができる。Primitiveとしては立・直方体、正多角柱、正多角錐を発生する。Magnifyは直交座標系の3軸方向それぞれについて座標値に係数をかける線型変換オペレータである。



(図－3)



(図－2)

また、Verifierは前述したConsistencyのうちシステム内で認識可能な性質を検証するもので プログラムのバグ対策用も兼ねて作成したものである。座標値を固定小数点表現した際に生ずる2面体などを検出することができる。

全体の考え方としては、形状情報の更新には必ずモデラのオペレータを用いることとし，図に示すオペレータと形状モデルを合わせて一種の論理データベースとする。これはConsistencyを保証するためにはシステムが閉じていることが要求されるからであって、ある程度の非能率は止むを得ないものと考えている。ただし参照については閉じておらず任意の参照を許容している。このため更新については決ったインターフェイス(オペレータへのパラメータとして)が存在するが、システム外への参照についての決ったインターフェイスは提供していないことになる。

4. 対称3値論理を用いた立体演算

4−1. 虚立体(Anti−Volume)

機械部品などにおいては、ハウジングの様にその部品が囲む内部の空間が他の部品の運動の保証をしなくてはならない場合や,何らかの幾何モデルがモデル化しない(できない)物質が存在するべき空間を保証しなくてはならない場合(たとえばタンクなど)がしばしば現われる。この様な空間は単に立体が存在しないということ以上の意味を持っており、またそれを形成する部品と一体に考えた方がよいことがある。他に建築設計などを考えても、建物の内部の部屋という空間は建物の外の空間とは意味的に区別されるべきもので，部屋が狭くなることは柱が細くなることと全く同等またはそれ以上の重要性がある。部品などのもつ機能を組み込む基礎として筆者らは、ハウジング内部などの空間のモデル化のために虚立体(Anti−Volume)を考えた。虚立体の意味解釈はモデラの外で行わなくてはならないが、幾何処理としてはモデラがその定義に基づいて実行する。

4−2. 対称3値論理(Symmetric Ternary)

幾何モデルシステムでは立体間の演算を行う際には空間に論理値を考えることが多いが、本システムでは通常の実立体に論理値+1，虚立体に−1，その他の空の空間に0を割当て，立体間の論理演算には対称3値論理(7)を使用した。この論理の真理値表を(図−4)に示す。この論理は一種の，差演算の拡張であって、項の間で可換則が成立する。可換則の成立は例えばPADLシステムのDIF操作が可換でないことと大きな対照をなす。可換則に注目したのは、現在のところ組込んではいないが、幾何演算のコマンドが与えられた時、直接演算してしまうのではなく，CSG(Constructive Solid Geometry)の様な演算木を考えてこれを保持しておき必要な時点で初めて幾何操作を行なおうとする場合、演算が可換であればSymbolic操作のみで幾何処理の一部を省略することが可能となると考えたことによる。特に本システムでは凸胞分割を行っているから異なる立体でも成分凸胞が等しい場合が多々存在する。

しかし、人間が幾何操作を考える場合にしばしば現われる切除の操作は，切除を行う立体の形状に考慮を払わなければ、予期せぬ虚立体が生ずる結果となる。また幾何操作自身も対称3値論理の採用で重くなっている。(図−5)にこの論理を用いた論理値の決定を模式的に示す。

SYMMETRIC TERNARY

AND

	-1	0	1
-1	-1	0	0
0	0	0	0
1	0	0	1

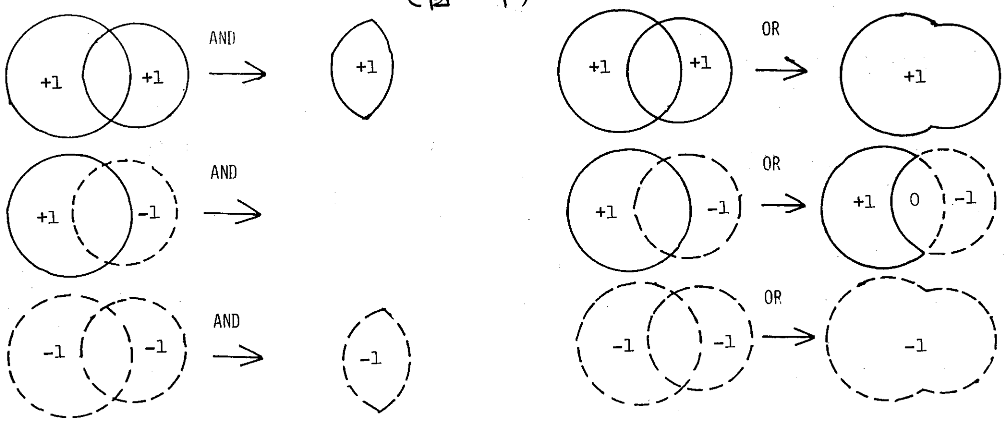
OR

	-1	0	1
-1	-1	-1	0
0	-1	0	1
1	0	1	1

NOT

1	-1
0	0
-1	1

(図−4)



(図−5)

5、多面体モデラのデータ構造

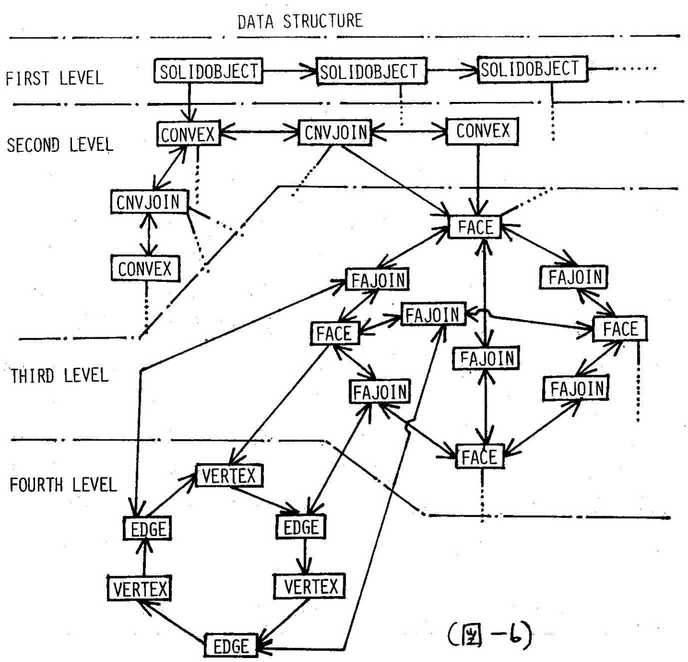
本システムでは前述した様に凸胞分割による記憶を行なうが、そのためのデータ構造の概略を(図−6)に示す。完全なリスト構造によって表現されるが、幾何の格納は概念的には単純な段階的詳細化になっている。以下にレベルごとの意味を示す。

第一レベル：立体のEntryであってファイル情報などが記憶される。

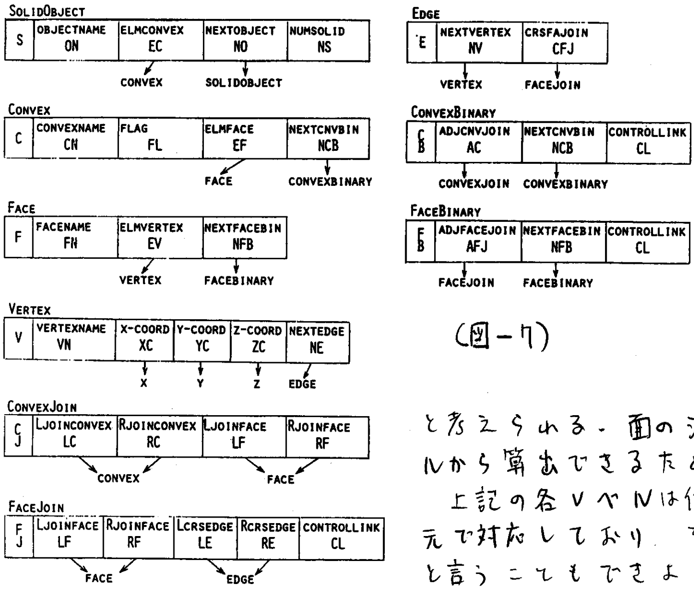
第二レベル：各立体の成分凸胞の接合関係を保持する。レベルを越えて接合面へのポインタを持つ。

第三レベル：各凸胞の構成凸多角形の接合関係を保持する。下位レベルの辺との間に双方向ポインタを持つ。

第四レベル：各凸多角形の構成頂点が、凸胞の外部から見た時、時計まわり方向になる様なサイクルとして記憶される。



(図−6)



(図ー7)

以上、第二レベルから第四レベルまではすべてトポロジーの記憶である。第四レベルの下に座標値がありジオメトリーを格納している。すなわち、この構造ではジオメトリーは頂点座標のみが存在することになる。これは後述する座標処理ルーチンで集中管理するためでもあるし、これだけで十分と考えられる。面の法線ベクトルも頂点のサイクルから算出できるため陽的には保持していない。

上記の各レベルは位相幾何学でいう胞体に各次元で対応しており、その意味でも最も自然な構造と言うこともできよう。また、可能な限り木構造に近くなる様に配慮し、双方向ポインタの使用も極力減らす様に努力した。レベルにまたがるポインタの処理は若干困難を伴いCNVJOINとFAJOINの処理が中心である。

(図ー7)に各リストアイテムの詳細を示す。CN, FN, VNの各名前は実際の処理で使用するためではなくidentificationのために用いる。凸胞や面の接合関係は任意個存在する。このため、それぞれConvexBinary, FaceBinaryと呼ぶ2進木表現で対処する様にしている。さらに、一部のモデラではリスト構造と共に表形式の記憶を用いて特定のアイテムの列挙を行っているが、ここではコントロールリンクと呼ぶ、ポインタにTagを付加する方法を用いて、隣接または接合関係を示すポインタの中に木構造を作り、これを列挙に用いる様にしている。図ではCLで示すのがコントロールリンク用のTag Fieldである。幾何演算中のコントロールリンクの再構成は、処理中にアイテムを作業スタックに積むため容易に行なうことができる。

また、(図ー7)からも明らかな様に立体の論理値は凸胞ごとに管理される。よって一つの立体に属する複数の凸胞が異る論理値を持っていてもそれらが接合していればよい。ここで接合とは、2つの凸胞がその構成面を共有することを言い、点のみを共有するときは分離していると考える。

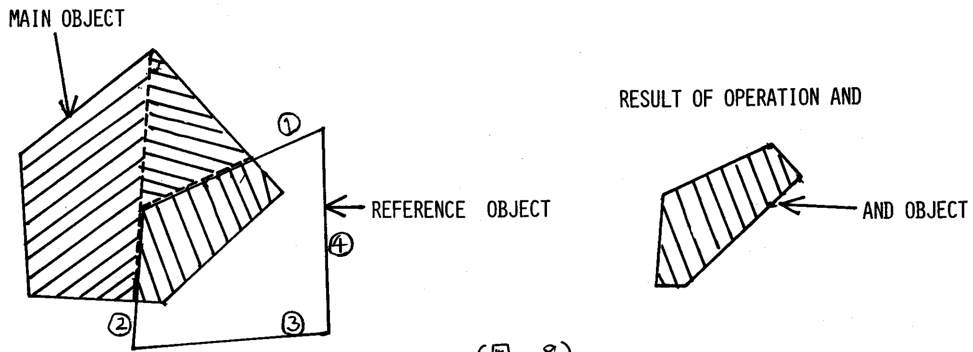
Consistencyについても、凸胞それぞれの検証は非常に容易である。凸胞間の接合関係のうち、接合関係が抜けている場合の検査は若干計算コストを要するが、各凸胞の凸性検証後はこの性質を使って能率を上げることができる。これらの検査は前述のVerifierで行うことができる。

6. 立体演算における幾何処理アルゴリズム

6－1. 凸胞間の演算アルゴリズム

凸胞間の演算アルゴリズムは、基本的な幾何学の定理である、一つの凸多面体は無限平面で高々2つに分割される、という性質のみを利用している。

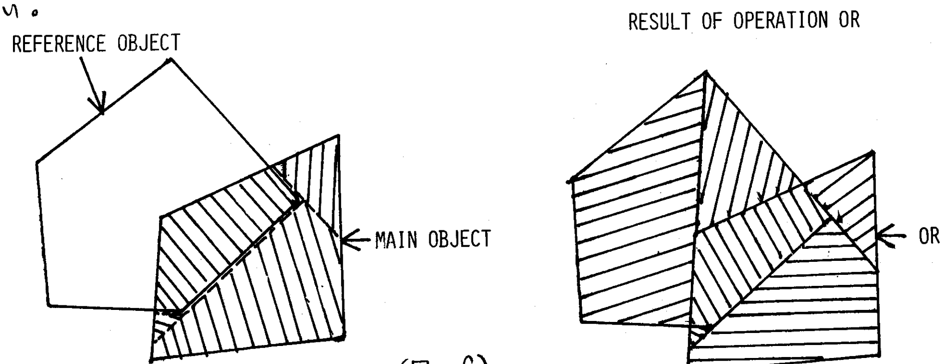
ここでまず、2つの凸胞について、一方を参照立体、もう一方を主立体とする。



(図－8)

参照立体は、その幾何情報は参照されるが自身には全く変化を生じない立体であり、主立体は参照立体の情報に基づいて切断され構造が変化するものである。また保存側とは、参照立体の1つの構成面を通る無限平面について参照立体と同じ側を呼び，非保存側とはその反対側を呼ぶことにする。

上記の定理を利用した凸胞間のANDを求めることは特に容易であって、まず参照立体の一つの構成面(に基づく無限平面)で、主立体を分割する。次に、生成されている保存側立体について参照立体の別の構成面で分割する。同様にすべての構成面について繰返す。もし途中で保存側立体が空になれば、AND部は空であり、立体が残ればこれが結果のAND部である。(図－8)に以上の過程を2次元的に表わした図を示す。Reference Objectの②の辺で最初に切断され次に①で切断されているのがわかる。③、④の辺についてはMain Object全体が保存側にあるので分割は生じない。



(図－9)

さらに、AND演算を利用してOR演算を行うことができる。ただしこの場合には切断面はすべて接合面としてポインタを結んでおかなくてはならない。まずANDによって接合されたAND部分と残りの凸胞の集合を作り、次に最初の参照立体を主立体，新しく生成されたAND部分の凸胞を参照立体として切断面を接合面とするAND操作を行う。最終に新しく生成された立体と既に生成されていた立体との間の接合関係を作って終了する。なぜなら切断面は完全に一致するからである。(図－9)にその様子を示す。図からもわかる様に、ORの場合、ANDと異り演算対象の2つの立体は両方とも構造が変更される。また、OR演算の場合、一回だけAND操作をして接合すればよい様に見えるが、対称3値論理を使用しているためAND部分の論理値が周囲と異る可能性があるから別々にする必要がある。

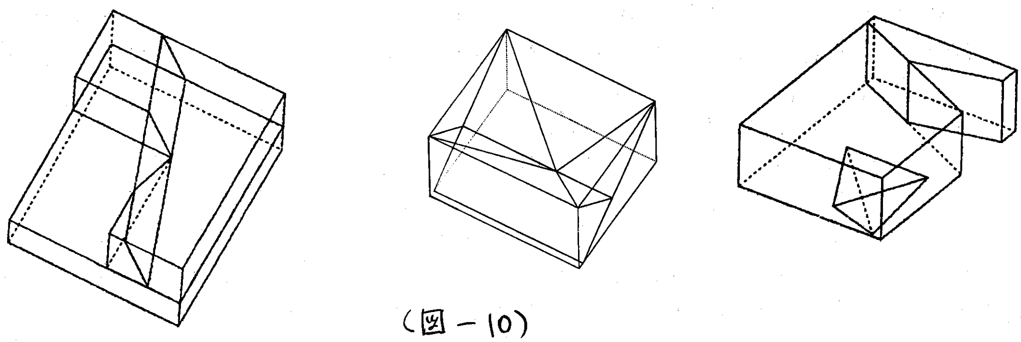
また、無限平面による凸胞の切断は成分凸多角形の切断に還元され非常に容

易に行うことができる。

凸胞分割からも明らかな様に、アルゴリズムは単純でかつ強靭であるから、幾何処理中にConsistentでない実体を発生する可能性は、アルゴリズム上もソフトウエア上も殆んどない。

6-2. 非凸立体の凸胞分割

一般の多面体の凸胞分割は容易でないが、本システムで作成したルーチンは、幾つか大きな制限を設けた上で分割を行うものである。すなわち、多面体を構成する稜線はすべて連結であること，貫通穴は一切存在しないことを要求し，この条件が満足される場合には凹稜線に沿って切断面を作って分割する。(図-10)にその例を示す。中央の図は(図-7)の図形を分割したものである。

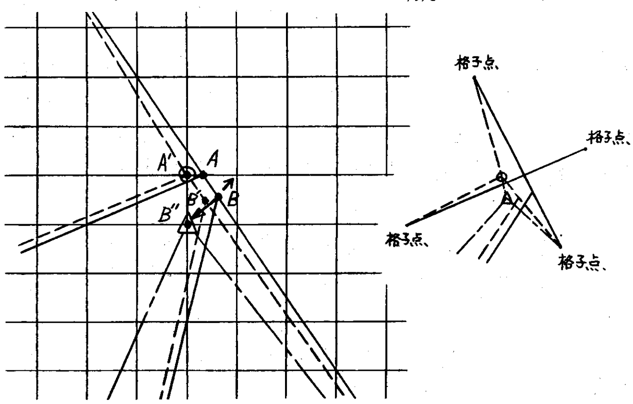


(図-10)

7. 多面体モデラの座標値処理

7-1. 幾何モデルにおける座標値のクラス

これ以降の議論については，一応座標系は3次元直交座標系に限定しておくが他の座標系においても大方適用するものと思われる。周知の様に座標値は一般的に数学的な実数であってこれをすべて計算機で記憶することはできない。しかし、2-2でも触れたが、幾何モデルに要求されるのは、結果の値としては外部から定る固定小数点数系の範囲にあればよい。それでも固定小数点数を幾何モデルの座標表現に用いることは厳密な意味では不可能である。この事情を一応例示したのが(図-11)であって、マス目が固定小数点数の1ユニットに対応している。元々格子点上にあって正確に表わされていた直線も2回切断を行えば無視出来ない誤差を生じている。A,Bが理論上の交点であるので誤差が累積しなければBは右上の格子点で近似されなくてはならないがB''に近似される結果となっている。



(図-11)

ところで、人間が直接取扱う

ことの出来る数値のクラスは有理数である。有理数を真の実数に写像する関数は数多いが幾何モデルに幾何学的に深く関係するのは距離としての平方根と回転における三角関数であろう。しかし、これらは通常幾何演算中には表われず、演算中に表われる数値演算は四則演算だけである。よってモデラの外部から与えられる幾何実体の座標は、方程式で与えられようと点列で与えられようと有理数であり多面体モデラにおいて座標値は有理数を表現すればよく，逆にConsistencyの目的のためには有理数を表現できなくてはならない。

7－2．多精度有理形式による座標値の無誤差表現

前記の様にConsistencyを重視するならば座標値表現として有理数が必要となる。周知の様に有理数は整数の２項組(分数)で表現されるので、本システムにおいては座標値を分子・分母の２整数で記憶することとした。この方法はRational Arithmeticとして知られているが(8)，近年Floating Slashとして数値計算の分野でも注目されてきている。ここでは、有理数を完全に表現しようとするのであるから、分子、分母の桁数はかなり大きくなる。本システムには２つの版があり、簡易型として64bit／64bitで一つの数値を表わすものと、高級版として10進15ケタを基数とする任意長整数を分子・分母に持つものである。現在、使用している計算機がIBM Typeであるため10進演算命令が有利と考えられるため採用したが記憶容量の点などで現在の所，主流ではなく、固定長の簡易型を主に使用して実験を行っている。

任意長整数ルーチンによる有理表現は無誤差ではあるが記憶域を消費する。演算中は必要であるが、演算終了後には何らかの組織的な方法を用いて精度を縮約させる必要があると考えられる。

8．PASCAL言語によるImplementと処理例．

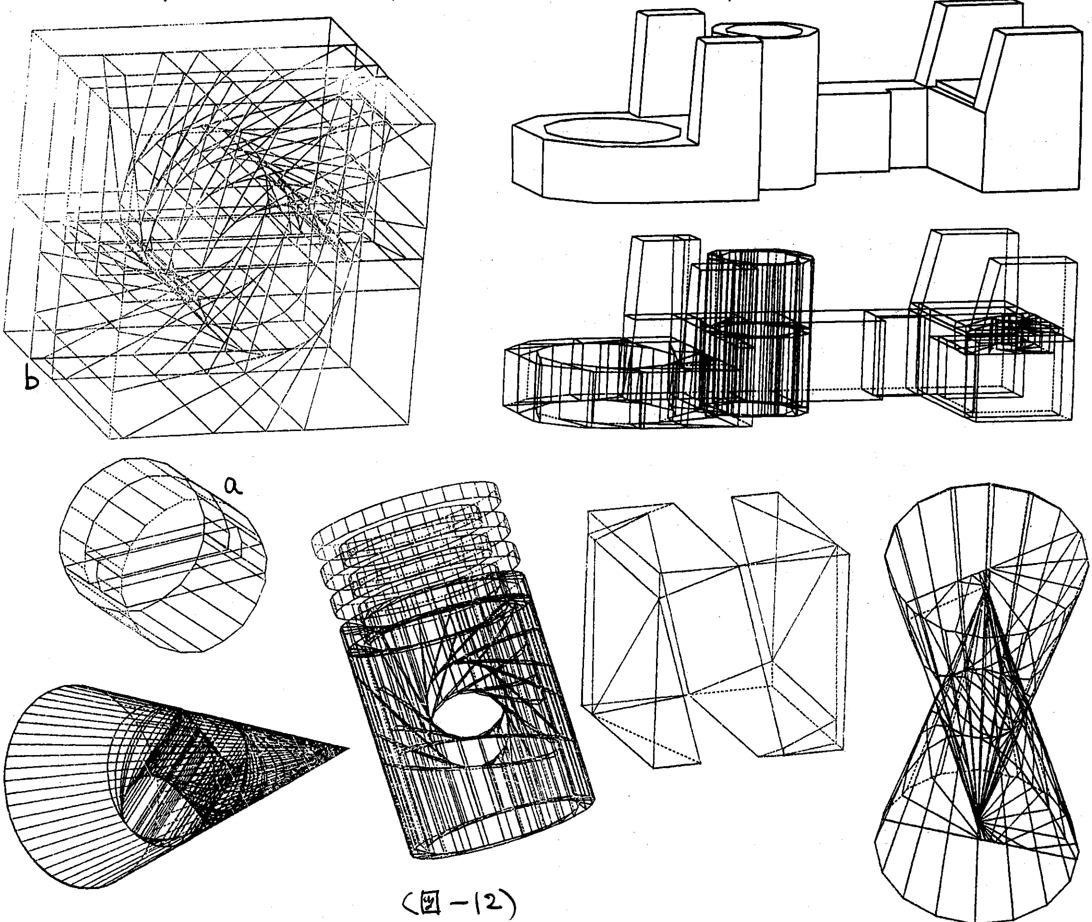
以上に述べたモデラは、一部の下位ルーチンを除いてPascalで記述されている。現在のところ全体で7000行，核の部分が3000行程度である。Pascalは強い構造データ型を持っているので(図－7)の不定長のリストを容易に取扱うことができる。この様な例としてALGOL68Cを用いた例があり(9)，演算子のユーザ定義機能などは有理ルーチン作成上有利と考えられる。現在処理系はPASCAL8000を用いているが、Heapの管理にmark/releaseを使用しており動的なGarbage collectionができない，オブジェクトコード長に制限があるなどの問題がある。これとは別に研究室のミニコン(HP 2111F 640kB)にモデラの移植を進めている。ミニコンにはPASCAL1000と呼ばれる処理系がありOverlayやHeapの動的Garbage Collectionが可能でありかなり強力な能力を持っている。

(図－12)に本報のモデラの処理例を示す。特に多面体であることを示すために稜合線はそのまま表示している。図のaとbの関係は、立方体から予め作っておいたaと同じ形状を立方体から除去するとbができ，さらに立方体からbを除去するとaが得られるという２重の反転を示している。

9．あとがき

凸胞分割と対称３値論理および座標値の有理表現を用いた多面体モデラについて述べた。今後、速度などの効率を高めること、立体への意味の付与などを行

なう予定である。また、位相幾何学でスパインと呼ばれる立体のベッチ数との関連のある性質を凸胞分割を利用して求めてゆきたいと考えている。



(図-12)

[謝辞]

筆者らが日頃御鞭撻いただく本学 林 国一教授に御礼申し上げる。また高精度計算について御教示いただいた高田昌之氏，本稿作成にあたって御尽力いただいた河野真儀氏，篠崎健二氏の諸兄に御礼申し上げる。

[文献]

(1) A Baer et al ; Geometric Modelling : a survey , CAD Vol 11 No 5 (1979)

(2) 伊藤,林 他；検図の自動化の基礎としての機構の分解・組立可能性の自動認識,機講論 800-3 (1980)

(3) H. B. Voelcker et al ; Geometric Modeling of Mechanical Parts & Processes, Computer December 1977

(4) G. Goos et al (ed) ; Lecture Notes in Computer Science 89, Springer (1980)

(5) A. A. G. Requicha ; Representations for Rigid Solids, Computing Surveys(1980)

(6) 伊藤,堀内他；階層プランニングを用いた画像認識の一手法,電子通信学会 PRL80-104 (1981)

(7) 田山典男他；対称3値論理系の提案,電子通信学会論文誌 T59-D (1976)

(8) D. E. Knuth ; The Art of Computer Programming 2 ,Addison Wesley

(9) I. C. Braid et al ; Geometric Modelling in ALGOL68 , ACM Sigplan Notices(1977)