CAD/CAMにおける形状処理とその応用に関する研究

昭和61年12月

塩　谷　景　一

目　　次

第1章　緒　　論

工業製品の高機能化および多機能化が進む中にあって，設計や生産業務を支援する計算機システムの必要性が従来に増し，高まっている。これは，高機能化によって，工業製品が一つの複雑なシステムとなってきたこと，あるいは，実物モデルを用いた試験が難しくなりつつあることが，主要な要因となっている。また，設計者一人あたりの設計負荷の増大も一つの要因である。

一方，LSIの急速な進歩に支えられて，システム開発に必要な環境が整えられつつあることも大きな要因であろう。1980年ごろの大型計算機に匹敵する処理能力が，今日ではマイクロプロセッサーで可能となり，設計者一人で一台のシステムを専有するエンジニアリングワークステーションが実用段階に入ってきている。

このように，設計や生産業務を支援するシステムはCAD/CAM(Computer Aided Design/Computer Aided Manufacturing)と呼ばれる。CAD/CAMシステムでは，製品の形状を計算機でどのように取り扱うかが重要な問題となる。つまり，曲面や円柱，直方体，球などの立体を論理的に完全な形で，計算機上に表現できるデータ構造が開発されなければならない。これは，単にグラフィックディスプレイに絵が画けるだけでなく，3次元形状としての種々の情報が抽出できなければ，CAD／CAMシステムのデータ構造としては，不充分であることを意味している。例えば，任意の一点を与えた場合，その点が形状の内部にあるか外部にあるか判別できなければならない。また，曲面の法線ベクトルも計算できるデータ構造が必要である。一方，設計や生産業務は思考過程を伴うから，システムはマン・マシンインタフェースに優れていなければならない。つまり，曲面式では形状修正が容易なパラメータが取り扱える必要があり，データ構造に対しても，形状修正に対する考慮が必要である。

ところで，CAD／CAMに関する研究の発展過程には大きな三つの節目があるように考えられる。第一の節目は、1960年ごろで，計算機で図形が扱え，さらに，人間と計算機が対話できることが示された時期である。M.I.T.のI.E.Sutherlandは 1962年にスケッチパッド(sketchpad)とよばれるシステムを開発し，計算機と対話形式で図形を扱う研究の成果を発表した。設計や生産においては，図形を取り扱うことが多いので，スケッチパッドの成果に基づいたCAD/CAM システムの実用化が試みられた。しかし，(1)3次元図形処理のアルゴリズムが未完成である，(2)立体を論理的に完全に表現できていない，(3)データベースについて十分検討されていない，などの問題のため，実際の設計や生産への展開は難しかった。一方，曲面の表現に関しては，S.A. Coonsによって一般的な表現式が提案され，曲面をパッチ(patch)とよばれる曲面要素に分割して表現する方法が現在でも用いられている。

第二の節目は１９７０年～１９８０年である。この時期には，立体を論理的に完全な形で表現す

るデータ構造が研究され，ソリッドモデリングと呼ばれる手法が数多く示された。この手法を用いたシステムも数多く開発され，試験的に使用された。

第三の節目は1983年ごろからである。1970年～1980年ごろの計算機の処理能力はCAD／CAMシステムにとっては低く，ソリッドモデルを用いたシステムの実用化はすすまなかった。しかし，前述したように計算機が急速に進歩し，CAD／CAMシステムを処理する能力を充分に持つようになった1983年ごろになっても，一般にはCAD／CAMシステムの実用化は進まなかった。これは，工業製品の設計や生産過程における形状処理が、立体を表現することを目的とした当時のモデリングシステムによる計算機上での処理では十分に行われなかったことに大きな原因があるものと考えられる。

以上の背景のもとに本論文では，CAD／CAMシステムへ応用することを前提にした，曲面や立体のモデリング手法について種々の提案を行ない，実用化したシステムについても述べる。

まず第２章では，自由曲面の形状制御について論じる。自由曲面を持つ製品の造形過程では，曲面形状の局所的制御と大域的制御が必要である。このためノット（ knot ）の間隔が全て１であるユニフォームB－spline 曲線を取り上げ，制御多角形と曲線の幾何学的諸量との関係を明らかにする。次にアウトラインといくつかのパラメータを用いて曲面の大域的制御を行なう手法を示し，制御頂点の操作による局所的制御と併用して所望の曲面を創成できることを示す。

第３章では，工業製品に適した自由曲面の表現について論じる。工業的応用において，曲面は図面によって要所要所の断面形状を厳密に寸法指定されることがある。この場合，断面形状は円弧と線分で構成されることがほとんどであるから，円弧のスプライン近似が問題となる。そこで，有理式B－spline による円弧表現について考察し，次に，Brown の重み関数を用いて，一般Coons 曲面に含まれる二つのロフト曲面の凸結合を行ない，複数セグメントの境界曲線を許す自由曲面式を示す。

第４章では，点群から自由曲面を創成する問題を論じる。流体力学や光学に基づいた技術計算によって曲面形状が設計される場合は，計算結果は点群として得られ，しかも格子状に並んでいないことが多い。この製品をＮＣ加工によって作成するには，任意に分布した点群から曲面を創成しなければならない。この問題に対処するには，三角形パッチを用いる手法も考えられるが，ここでは曲面創成後の処理や，他の曲面形状定義システムとの結合も考慮し，任意に分布した点群を近似的に等価な格子状の点群におきかえて，曲面を創成する手法を示す。

第５章では，立体のモデリング手法について論じる。一般に工業製品は適当な平面によって切断し，それぞれの部分にについて適当な局所座標系を用意すると x y 平面上の一価関数によって部分形状の上界が定義される場合が多い。そこで，部分形状ごとに局所座標系を用意したとき，それらの形状が一価関数で表される上界の面と定義平面ではさまれた領域を立体セグメントとし，この立

体セグメントを基礎において形状モデリングを行う手法を示す。ここでは，工業製品形状の分析に基づいた手法により，従来のモデリング手法と比較すると，サブセットではあるが計算量が軽減でき，また，設計・生産における各種形状処理にも向いたシステムを構築できることを述べる。

第6章では，金型用3次元ＣＡＤ／ＣＡＭシステムについて述べる。このシステムでは，金型形状の記述，加工手順・加工条件の指定などは専用言語によるプログラミングで行なう。従来，加工システムではＡＰＴ系の言語が用いられたが，加工工程・加工条件の変更に際してプログラムの大幅な修正が必要であり，プログラムにおける変数の有効範囲も不明確である。そこで，ＡＬＧＯＬで用いられたブロック構造の特徴を持つ専用言語により，これらの問題に対処できることを示す。次に，複雑な曲面の創成を，あらかじめ設定した１５種類の曲面の組み合わせによって行なう手法を示す。この曲面要素は，曲面を構成する曲線間の相互関係によって，金型図面に頻出するパターンを１５種類設定したものである。

第7章では、人工衛星用3次元CADシステムについて述べる。実際の設計業務へ3次元CADシステムを適用するには，形状定義が容易で，また，種々の技術計算を行なうアプリケーションモジュールへの入力データが自動作成できる立体モデリング手法が必要である。汎用のモデリングシステムを用いた場合、設計で取り扱う形状の基本単位とシステムで扱う形状の基本単位とが一致せず，形状定義に手間どることが多い。また，各種技術計算アプリケーションモジュールへの入力データの自動作成が不可能であったり，バッチ方式でデータを作成することが多く，設計作業が円滑に進み難いことがある。そこで，形状定義を容易にするため，人工衛星の設計で頻出する形状をプリミティブとし，さらに，形状全体は，プリミティブの単純な接合で定義できる手法を示す。つぎに，対話方式で円滑に設計できるシステムにするため，人工衛星の設計で用いる質量特性計算・熱特性計算などの技術計算モジュールで必要な入力データがほぼ直接抽出できるデータ構造を提案する。

第8章では、各章で得られた結果を述べ，本論文の総括を行なう。

第2章　自由曲面の形状制御

2.1　緒　言

家庭電化製品，容器等，その外形形状が機能的な要求でなく，意匠によって決まる工業製品の造形過程を計算機システムによって支援する場合，曲面の数学的な取扱いが難しいことが多い。このような曲面は方程式$f(x,y,z)=0$として表現できず，自由曲面（Sculptured Surface）とよばれている[1]。

自由曲面の研究では，Coons によって１９６７年に提示された式[2]を理論的に発展させたもの，あるいは実用的な式へと改善したものが多く用いられている。このように，Coons の式は今日の自由曲面の研究に種々な影響を与えたが，意匠設計の場合に重要な形状制御能力を持たなかった。そのため，Bézier[3]やRiesenfeld[4]は制御多角形と名付けられた形状修正の制御パラメータを持つBézierの式，B－spline の式を提案し，意匠設計への適用を試みた。しかし，初期曲面を作るときの入力過程や，曲面の大域的制御と局所的制御を併用して意図する曲面を創成する問題については十分に考慮されていなかった。

本章では，上記の問題を考察するために，まず準備として一般的な曲面論について述べ，B－splineの式との比較のためにCoons の式のサブセットであるFerguson の式を示す。次に意匠設計への適用を考えた自由曲面の表現式について論じる。ここでは，応用上有用なノット(knot)の間隔が１であるB－spline曲線について，接線ベクトルや曲線の通過点のような幾何学的特徴をよく表す量を陽に扱う形式の新しい計算式を提案する。さらに，曲面全体の特徴を表すアウトラインといくつかのパラメータの入力により，初期曲面を創成する手法を示す。この手法を用いたシステム開発例を示し，従来から使われている制御頂点の操作による局所的制御と，新たに導入した形状制御関数などを用いた大域的制御により，比較的少い労力で曲面の創成ができることを明らかにする。

2.2　曲面論

2.2.1　パラメトリック表現

曲面の数式による表現方法は，パラメトリック（parametric）なベクトル値関数による表現と，ノンパラメトリック（non－parametric）な表現がある。x，y，zの成分を持つユークリッド空間（Euclideanspace）で，上記表現方法の一般形式を次に示す。

$$f(x,y,z)=0 \qquad (2.1)$$

$$\boldsymbol{S}(s,t)=(x(s,t),y(s,t),z(s,t)) \qquad (2.2)$$

式(2.1)は，曲面のノンパラメトリックな表現である。式(2.2)は，二つのパラメータ s , t によるベクトル値関数として表現した曲面のパラメトリックな表現である。自由曲面は，第2.1節で定義を与えたように，$f(x,y,z)=0$ として表現できない。しかし，式(2.2)のパラメトリックな表現では定義が可能である。一方，$f(x,y,z)=0$ と表現可能な曲面でも，パラメトリックな表現が用いられることが多い。その理由は，曲面をパラメトリックに表現すると，以下に示す利点があるためである。

(1) 座標軸に独立な表現である。

(2) 1階の導関数が無限大になることが避けられる。

(3) 多価曲面を明確に表現できる。

(4) 曲面上の点の算出が容易である。

(5) アフィン変換，射影変換が容易である。

これらの理由から，曲面はほとんどの場合パラメトリックな表現が用いられる。本論文でも，曲面は，パラメータ s , t を用いたパラメトリック表現を用いる。また，同様の表現は，曲線についても用いられ，一つのパラメータ s を用いて式(2.3)で表現する。

$$\boldsymbol{C}(s)=(x(s),y(s),z(s)) \qquad (2.3)$$

2.2.2 曲面の特性量

後述する曲面式導入で関係する重要な曲面特性量の定義を示す。

(1) s 方向の曲線，t 方向の曲線

曲面を $\boldsymbol{S}$ とし，$\boldsymbol{S}=\boldsymbol{S}(s,t)$ で表す。$s=s_0$ とパラメータ s を一定値にすると，$\boldsymbol{S}(s_0,t)$ は曲面 $\boldsymbol{S}$ 上の曲線を表す。この曲線を t 方向の曲線と呼ぶ。同様に，$t=t_0$ とパラメータ t を一定値にすると，$\boldsymbol{S}(s,t_0)$ は $\boldsymbol{S}$ 上の曲線を表す。この曲線を s 方向の曲線と呼ぶ(図2.1)。

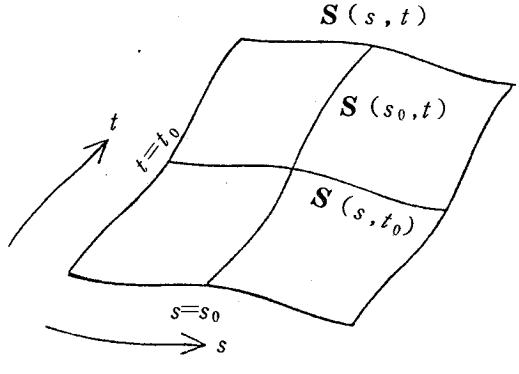


図2.1　s 方向の曲線と t 方向の曲線

(2) C^{γ} 級の曲線

$$J_x \equiv \frac{\partial(y,z)}{\partial(s,t)} = \begin{vmatrix} \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{vmatrix}$$

$$J_y \equiv \frac{\partial(z,x)}{\partial(s,t)} = \begin{vmatrix} \frac{\partial z}{\partial s} & \frac{\partial x}{\partial s} \\ \frac{\partial z}{\partial t} & \frac{\partial x}{\partial t} \end{vmatrix} \qquad (2.4)$$

$$J_z \equiv \frac{\partial(x,y)}{\partial(s,t)} = \begin{vmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} \end{vmatrix}$$

J_x, J_y, J_z は，ヤコビアン（Jacobian）である。C^{γ} 級の曲面とは，ヤコビアンが同時には零にならないで，つまり，

$$J_x^2 + J_y^2 + J_z^2 \neq 0 \qquad (2.5)$$

を満足し，曲面 $\boldsymbol{S}(s, t)$ の成分 x, y, z の s, t に関する γ 階までの導関数が存在して，しかも，それが連続であるものをいう。式（2.5）の条件は，曲面 $\boldsymbol{S}(s, t)$ が点や曲線に退化したり，とがった点のような特異点を持たないためのものである。

(3) 法線ベクトル

法線ベクトルを $\boldsymbol{N}$ で表す。法線ベクトルは，式（2.4）のヤコビアンを用いて次式で表される。

$$\boldsymbol{N} = \frac{\partial \boldsymbol{S}(s,t)}{\partial s} \times \frac{\partial \boldsymbol{S}(s,t)}{\partial t}$$

$$= (J_x, J_y, J_z) \qquad (2.6)$$

(4) 接 平 面

曲面上の点 $\boldsymbol{S}(s_0, t_0)$ において，s 方向の曲線の接線ベクトルを $\boldsymbol{S}_s(s_0, t_0)$，$t$ 方向の曲線の接線ベクトルを $\boldsymbol{S}_t(s_0, t_0)$ とする。$\boldsymbol{S}_s(s_0, t_0)$，$\boldsymbol{S}_t(s_0, t_0)$ は式（2.7）で定義される。

$$\boldsymbol{S}_s(s_0,t_0) = \left.\frac{\partial \boldsymbol{S}(s,t)}{\partial s}\right|_{\substack{s=s_0\\t=t_0}}$$

$$\boldsymbol{S}_t(s_0,t_0) = \left.\frac{\partial \boldsymbol{S}(s,t)}{\partial t}\right|_{\substack{s=s_0\\t=t_0}} \qquad (2.7)$$

このとき，$\boldsymbol{S}_s(s_0,t_0)$，$\boldsymbol{S}_t(s_0,t_0)$ の張る平面点を$\boldsymbol{S}(s_0,t_0)$における接平面とよぶ（図2.2）。

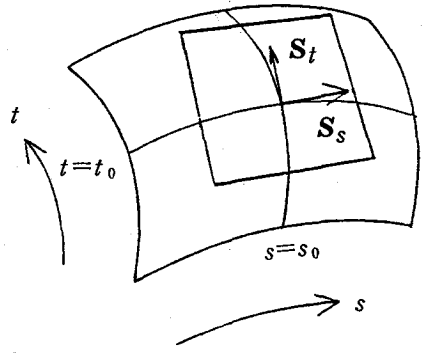


図2.2　接　平　面

(5)　曲面の接続

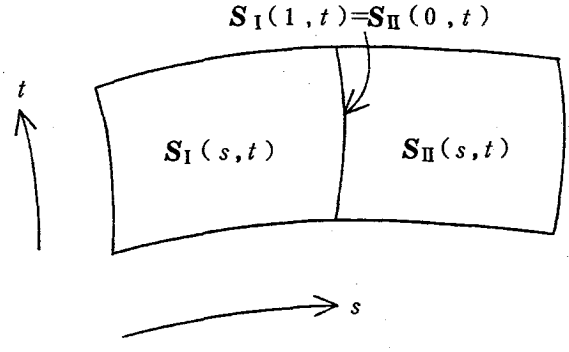


図2.3　二つの曲面の接続

二つの曲面を境界曲線を境にして接続する場合，境界曲線上における連続性の問題が生じる（図2.3）。図2.3に示す二つの曲面$\boldsymbol{S}_{\mathrm{I}}(s,t)$，$\boldsymbol{S}_{\mathrm{II}}(s,t)$の接続を例に接続条件式を示す。なお，$\boldsymbol{S}_{\mathrm{I}}(s,t)$，$\boldsymbol{S}_{\mathrm{II}}(s,t)$には次の(a),(b)の条件を与えておくが，一般性は失われない。

(a) $\boldsymbol{S}_{\mathrm{I}}, \boldsymbol{S}_{\mathrm{II}}$ の変域は $0 \leqq s \leqq 1$, $0 \leqq t \leqq 1$ である。

(b) $\boldsymbol{S}_{\mathrm{I}}, \boldsymbol{S}_{\mathrm{II}}$ は $\boldsymbol{S}_{\mathrm{I}}(1, t), \boldsymbol{S}_{\mathrm{II}}(0, t)$ の各境界曲線で接続されており，次式を満たしている。

$$\boldsymbol{S}_{\mathrm{I}}(1, t) = \boldsymbol{S}_{\mathrm{II}}(0, t) \quad , \quad 0 \leqq t \leqq 1 \qquad (2.8)$$

条件式（2.8）は両曲面の境界曲線が一致する条件である。

ここで，$\boldsymbol{S}_{\mathrm{I}}$ と $\boldsymbol{S}_{\mathrm{II}}$ の s 方向，t 方向の接続ベクトルをそれぞれ $\boldsymbol{S}_{\mathrm{I},s}$, $\boldsymbol{S}_{\mathrm{I},t}$ と $\boldsymbol{S}_{\mathrm{II},s}$, $\boldsymbol{S}_{\mathrm{II},t}$ で表すことにする（式（2.7）参照）。二つの曲面が接している境界曲線上で接平面が連続である条件は，$\gamma(t)$ を任意のスカラー量として，接平面の定義から式（2.9）となる。

$$\boldsymbol{S}_{\mathrm{II},s}(0,t) = \mu(t)\cdot\boldsymbol{S}_{\mathrm{I},s}(1,t) + \gamma(t)\cdot\boldsymbol{S}_{\mathrm{I},t}(1,t) \qquad (2.9)$$
$$0 \leqq t \leqq 1$$

ここで，$\mu(t)$ は次式を満たす正のスカラー関数である。次式は，法線ベクトルの方向が二つの曲面の境界曲線上で連続の条件である。

$$\boldsymbol{S}_{\mathrm{II},s}(0,t) \times \boldsymbol{S}_{\mathrm{II},t}(0,t) = \mu(t)\cdot\boldsymbol{S}_{\mathrm{I},s}(1,t) \times \boldsymbol{S}_{\mathrm{I},t}(1,t) \qquad (2.10)$$
$$0 \leqq t \leqq 1$$

ところで，$\gamma(t)$ は任意のスカラー量であるから，これを零とおくと，式（2.9）から

$$\boldsymbol{S}_{\mathrm{II},s}(0,t) = \mu(t)\cdot\boldsymbol{S}_{\mathrm{I},s}(1,t) \qquad (2.11)$$

式（2.11）は，s 方向の曲線が二つの曲面にわたって連続であるための条件式である。

二つの曲面の接続条件として，三つの式（2.8），（2.9），（2.11）のいずれかを用いるが，条件としては式（2.8）が最もゆるく，式（2.11）が最も厳しいものとなっている。

2.3 自由曲面の表現

自由曲面は，形状が複雑であり広い範囲の表面形状を定義の対象とすることが多いため，一つの数式で表現することは難しい。そのため，自由曲面をパッチ（patch）とよばれる三角形や四角形の曲面要素に分割し，パッチごとに曲面式をあてはめ，パッチを接続することで曲面全体を定義する（図2.4）。そのため，自由曲面の定義式では，パッチの表現式の問題とパッチの接続の問題がある。本節では，比較のため，自由曲面式の一手法であるFerguson の式について述べる。

Ferguson は，パッチを四角形に限定し，パッチの四隅の点における位置ベクトルと，接線ベクトルによって曲面要素を定義する手法を考えた[8]。この手法によると，格子状に点が与えられ，

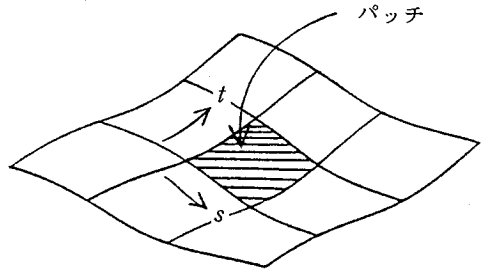


図2.4　パ　ッ　チ

かつ，各点におけるs方向，t方向の接線ベクトルが与えられると，与えられた点を四隅の点とするパッチが創成できる。

このとき，各パッチは隣接するパッチと同一の境界曲線を有し，かつ境界曲線を横切る方向の接線ベクトルも共有されるので，各曲面パッチはスロープまで連続に接続される。

以下に，Fergusonの式を示す。一つのパッチを$\boldsymbol{S}(s,t)$，$0 \leqq s, t \leqq 1$で表すと，データとして与えられるのは，次に示す１２個のベクトルである（図2.5）。

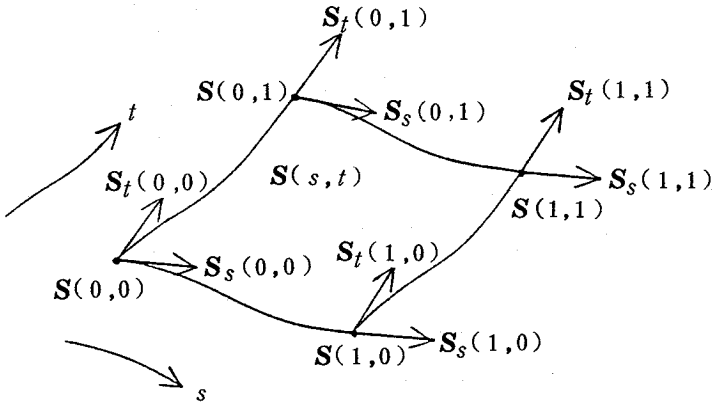


図2.5　Fergusonの曲面を定義するベクトル

$\left.\begin{matrix}\boldsymbol{S}(0,0)\\ \boldsymbol{S}(0,1)\\ \boldsymbol{S}(1,0)\\ \boldsymbol{S}(1,1)\end{matrix}\right\}$　位置ベクトル

$\left.\begin{matrix}\boldsymbol{S}_s(0,0)\\ \boldsymbol{S}_s(0,1)\\ \boldsymbol{S}_s(1,0)\\ \boldsymbol{S}_s(1,1)\end{matrix}\right\}$　s方向の接線ベクトル

$$\left.\begin{matrix} \boldsymbol{S}_t(0,0) \\ \boldsymbol{S}_t(0,1) \\ \boldsymbol{S}_t(1,0) \\ \boldsymbol{S}_t(1,1) \end{matrix}\right\} \quad t \text{ 方向の接線ベクトル}$$

このとき，パッチは式(2.1 2)で定義される。

$$\boldsymbol{S}(s,t)=[H_1(s),H_2(s),H_3(s),H_4(s)] \times \begin{bmatrix} \boldsymbol{S}(0,0) & \boldsymbol{S}(0,1) & \boldsymbol{S}_t(0,0) & \boldsymbol{S}_t(0,1) \\ \boldsymbol{S}(1,0) & \boldsymbol{S}(1,1) & \boldsymbol{S}_t(1,0) & \boldsymbol{S}_t(1,1) \\ \boldsymbol{S}_s(0,0) & \boldsymbol{S}_s(0,1) & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{S}_s(1,0) & \boldsymbol{S}_s(1,1) & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} H_1(t) \\ H_2(t) \\ H_3(t) \\ H_4(t) \end{bmatrix}$$

$$0 \leqq s,\ t \leqq 1 \qquad (2.12)$$

ここで，H_1，H_2，H_3，H_4は3次の Hermite 補間関数で，次式で表される(変数がsのときも同じ)。

$$\begin{aligned} H_1(t) &= 2t^3 - 3t^2 + 1 \\ H_2(t) &= -2t^3 + 3t^2 \\ H_3(t) &= t^3 - 2t^2 + t \\ H_4(t) &= t^3 - t^2 \end{aligned} \qquad (2.13)$$

なお，H_1，H_2，H_3，H_4は一種の重み関数 (weight function)であり，自由曲面の研究分野では，混ぜ合せ関数(blending function) といわれる。

このように， Ferguson の式はパッチの四隅の点におけるデータを与えると簡単なマトリックス表現でパッチを表すことができ，しかも隣接パッチとの連続性は $\boldsymbol{C}^1$ 級である。しかし，次に示すような問題点がある。

(1) 曲面を創成するために，格子状の点群とその点における接続ベクトルをデータとして与える必要がある。しかし，設計者が意図する形状をこれらのデータで表現するのは難しい。

(2) 曲面形状の修正に際して，制御できるパラメータは位置ベクトルと接線ベクトルであり，これらのパラメータを制御して修正される曲面形状の予想が難しい。

第2.4節では，上記の問題点に対処するための理論式を提案する。

2.4 ユニフォームB－splineの計算法

B－spline を用いた自由曲線あるいは曲面を実際の設計に応用する場合，二つの点が考慮されなければならない。一つは制御多角形と曲線の幾何学的関係を直感的に把握しやすいものにすることである。これにより，B－spline 曲線の特徴である局所的形状制御を有効に使うことができ，設計者と計算機とのマンマシン・インタフェースの向上を図ることができる。第二の点は，設計者が初期曲面を作るときの入力過程の問題である。あらかじめ数値データがある場合には，通過点を与えるという入力方法も有効であるが，この方法には多くの労力が要求される。もともと設計者が，形状の全体的なイメージを持っている場合には，通過点の入力はわずらわしいものであり，もっと簡便な方法が考えられなければならない。この点については，山口[9)~11)]が問題として取り上げ，いくつかの提案を行っている。

本節では，まず，応用上重要なノットの間隔がすべて１であるユニフォームB－spline 曲線を取り上げ，制御多角形との幾何学的関係を明らかにする。そして，曲線の存在領域，通過点，接線ベクトルと制御多角形の関係を明確に示すと共に，幾何学的な意味に密着した新しい計算式を導入する。

以下に述べる曲線式は B－spline の特別な場合（ノット間隔がすべて１）である。あえてこれを取り上げる理由は，応用上重要であることと，議論をこの場合に限ることによって，制御多角形と曲線の幾何学的関係が明確になるからである。ここでは，煩雑さを避けるため，用途の多いオーダ４の場合について具体的に述べ，補足的にオーダ３の場合についても触れる。

2.4.1 幾何学的な曲線の定義

ここでは，曲線を幾何学的に定義する。オーダが３の場合を図2.6(a)に示す。

$\boldsymbol{P}_{i-2}$ ，$\boldsymbol{P}_{i-1}$ ，$\boldsymbol{P}_i$ を制御頂点とするとき，$\boldsymbol{P}_{i-2}$ と$\boldsymbol{P}_{i-1}$ の中点と$\boldsymbol{P}_{i-1}$ を結ぶ線分を $s:1-s$ に内分する点を$\boldsymbol{P}_{i-1}^{[1]}$ とする。同様に$\boldsymbol{P}_{i-1}$ と$\boldsymbol{P}_i$ の中点を求め，$\boldsymbol{P}_{i-1}$ とこの中点を結ぶ線分を $s:1-s$ に内分する点を$\boldsymbol{P}_i^{[1]}$ とする。s を０から１まで動かすと，$\boldsymbol{P}_{i-1}^{[1]}$ と$\boldsymbol{P}_i^{[1]}$ は制御頂点を結ぶ線分を $s:1-s$ に内分する点 $\boldsymbol{C}(s)$ の軌跡によって曲線が定義される。すなわち，式で表せば，一つの曲線セグメントは次のようになる。

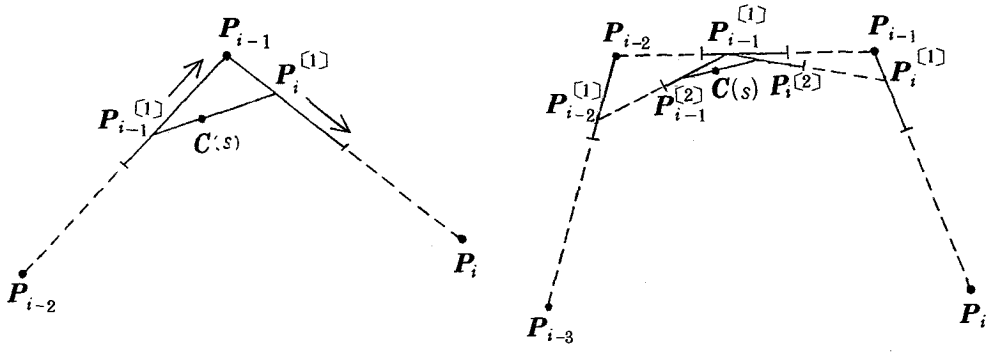
$$\boldsymbol{C}(s)=s\cdot\boldsymbol{P}_i^{[1]}(s)+(1-s)\cdot\boldsymbol{P}_{i-1}^{[1]}(s)\ ,\quad 0\leqq s\leqq 1 \qquad (2.14)$$

オーダ4の場合を図2.6(b)に示す。制御頂点４点が与えられたとき，それらを結ぶ線分のそれぞれを３等分して，実線で示した部分を動く点と考える。すなわち，

$$\boldsymbol{P}_{i-2}^{[1]}(s)=(1-s)\cdot\frac{2\boldsymbol{P}_{i-2}+\boldsymbol{P}_{i-3}}{3}+s\cdot\boldsymbol{P}_{i-2} \qquad (2.15)$$

$$\boldsymbol{P}_{i-1}^{[1]}(s)=(1-s)\cdot\frac{\boldsymbol{P}_{i-1}+2\boldsymbol{P}_{i-2}}{3}+s\cdot\frac{2\boldsymbol{P}_{i-1}+\boldsymbol{P}_{i-2}}{3} \qquad (2.16)$$

$$\boldsymbol{P}_{i}^{[1]}(s)=(1-s)\cdot\boldsymbol{P}_{i-1}+s\cdot\frac{\boldsymbol{P}_{i}+2\boldsymbol{P}_{i-1}}{3} \qquad (2.17)$$



(a) オーダ3の場合　　(b) オーダ4の場合

図2.6　幾何学的な曲線の定義

である。次に，sによって定められるこれら3点を用いて，オーダ3と同じように$\boldsymbol{P}_{i-1}^{[2]}(s)$と$\boldsymbol{P}_{i}^{[2]}(s)$を求める。

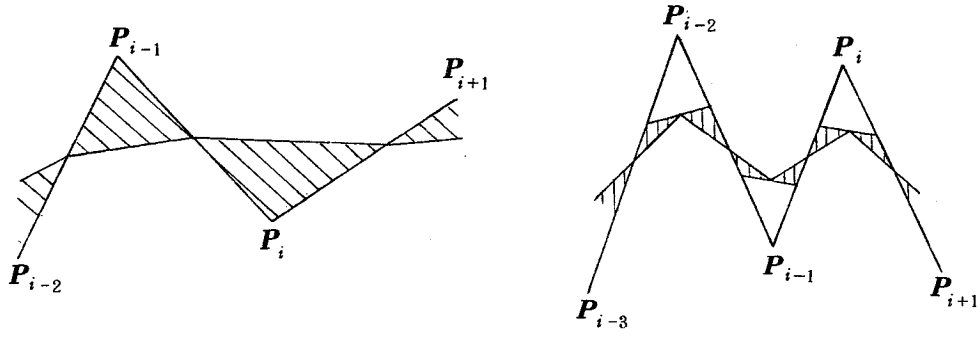
$$\boldsymbol{P}_{i-1}^{[2]}(s)=(1-s)\cdot\frac{\boldsymbol{P}_{i-2}^{[1]}(s)+\boldsymbol{P}_{i-1}^{[1]}(s)}{2}+s\cdot\boldsymbol{P}_{i-1}^{[1]}(s) \qquad (2.18)$$

$$\boldsymbol{P}_{i}^{[2]}(s)=(1-s)\cdot\boldsymbol{P}_{i-1}^{[1]}(s)+s\cdot\frac{\boldsymbol{P}_{i-1}^{[1]}(s)+\boldsymbol{P}_{i}^{[1]}(s)}{2} \qquad (2.19)$$

これら2点を用いると，一つの曲線セグメントは次式で与えられる。

$$\boldsymbol{C}(s)=s\cdot\boldsymbol{P}_{i}^{[2]}(s)+(1-s)\cdot\boldsymbol{P}_{i-1}^{[2]}(s),\quad 0\leqq s\leqq 1 \qquad (2.20)$$

このように幾何学的に曲線を定義する方法は，Cox,と De Boor[12]による線形計算の繰返しを用いたB−spline曲線の計算方法に基づいている。しかし，論議をノットの間隔が1である場合に限ってしまうと，例えば曲線の存在する領域は図2.7に示すように限定される。



(a) オーダ3の場合　　(b) オーダ4の場合

図2.7　曲線の存在する領域

2.4.2　曲線の計算式

上に述べた曲線の定義は，オーダが大きくなっても同様に定義できるが，幾何学的な説明は煩雑になるので省略する。このような線形計算の繰返しで曲線を定義することにすれば，一般にオーダMの曲線は形式的に

$$\begin{aligned} \boldsymbol{C}_i^M(s) &= s\cdot\boldsymbol{P}_i^{[M-2]}(s)+(1-s)\cdot\boldsymbol{P}_{i-1}^{[M-2]}(s) \\ &= \boldsymbol{P}_{i-1}^{[M-2]}(s)+s\cdot\{\boldsymbol{P}_i^{[M-2]}(s)-\boldsymbol{P}_{i-1}^{[M-2]}(s)\} \\ &= \boldsymbol{f}_i^M(s)+s\cdot\boldsymbol{g}_i^M(s) \qquad 0\leqq s\leqq 1 \end{aligned} \qquad (2.21)$$

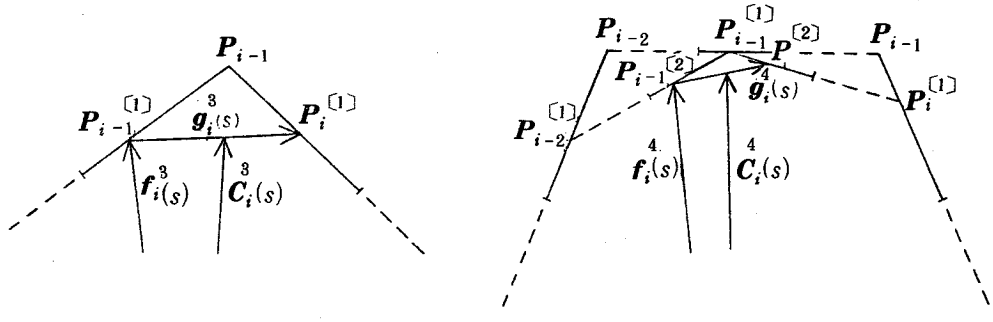
と書くことができる。

オーダが3と4の場合について，ベクトル$\boldsymbol{f}_i^M(s)$と$\boldsymbol{g}_i^M(s)$のもつ幾何学的意味を図2.8に示す。例えばオーダ4の場合

$$\left.\begin{aligned} \boldsymbol{f}_i^4(s) &= \boldsymbol{P}_{i-1}^{[2]}(s) \\ \boldsymbol{g}_i^4(s) &= \boldsymbol{P}_i^{[2]}(s)-\boldsymbol{P}_{i-1}^{[2]}(s) \end{aligned}\right\} \qquad (2.22)$$

であり，$\boldsymbol{g}_i^4(s)$はパラメータ$\boldsymbol{S}$を指定したときの曲線上のその点における接線の方向を示す。一般に曲線上の任意の点の接線ベクトルは$(M-1)\cdot\boldsymbol{g}_i^M(s)$で与えられる。

接線ベクトルを陽に表した式(2.21)の形式で曲線を表すものとして，幾何学的定義から計算式を誘導すると次のようになる。$\boldsymbol{f}_i^M(s)$，$\boldsymbol{g}_i^M(s)$はそれぞれ制御頂点を用いて，



(a) オーダ3の場合　　(b) オーダ4の場合

図2.8　各ベクトルの幾何学的な意味

$$
\begin{aligned}
\boldsymbol{f}_i^{M}(s) &= \sum_{l=i-(M-1)}^{i} \boldsymbol{P}_l \cdot \boldsymbol{F}_l^{[0]}(s) \\
&= \sum_{l=i-(M-1)+1}^{i} \boldsymbol{P}_l^{[1]} \cdot \boldsymbol{F}_l^{[1]}(s) \\
&= \sum_{l=i-(M-1)+2}^{i} \boldsymbol{P}_l^{[2]} \cdot \boldsymbol{F}_l^{[2]}(s) \\
&\quad \vdots \\
&= \sum_{l=i-1}^{i} \boldsymbol{P}_l^{[M-2]} \cdot \boldsymbol{F}_l^{[M-2]}(s)
\end{aligned}
\qquad (2.23)
$$

同様に

$$
\begin{aligned}
\boldsymbol{g}_i^{M}(s) &= \sum_{l=i-(M-1)}^{i} \boldsymbol{P}_l \cdot \boldsymbol{G}_l^{[0]}(s) \\
&= \sum_{l=i-(M-1)+1}^{i} \boldsymbol{P}_l^{[1]} \cdot \boldsymbol{G}_l^{[1]}(s) \\
&= \sum_{l=i-(M-1)+2}^{i} \boldsymbol{P}_l^{[2]} \cdot \boldsymbol{G}_l^{[2]}(s) \\
&\quad \vdots \\
&= \sum_{l=i-1}^{i} \boldsymbol{P}_l^{[M-2]} \cdot \boldsymbol{G}_l^{[M-2]}(s)
\end{aligned}
\qquad (2.24)
$$

と表される。ここで $\boldsymbol{F}_l^{[n]}$ と $\boldsymbol{G}_l^{[n]}$（$n=0, 1, 2\cdots, M-2$）は

$$F_l^{[M-2]}(s) = \begin{cases} 1\ (l=i-1) \\ 0\ (l \neq i-1) \end{cases}$$

$$G_l^{[M-2]}(s) = \begin{cases} 1\ (l=i) \\ -1\ (l=i-1) \\ 0\ (\text{その他}) \end{cases}$$

を初期値としてそれぞれ次式で計算できる。

$$\begin{aligned} F_l^{[n]}(s) &= \frac{M-n-i+l-s}{M-n-1} \cdot F_{l+1}^{[n+1]}(s) + \frac{s+i-l}{M-n-1} \cdot F_l^{[n+1]}(s) \\ G_l^{[n]}(s) &= \frac{M-n-i+l-s}{M-n-1} \cdot G_{l+1}^{[n+1]}(s) + \frac{s+i-1}{M-n-1} \cdot G_l^{[n+1]}(s) \end{aligned} \qquad (2.25)$$

上に示した計算式が，ノットの間隔が1であるB−spline曲線の計算式になっていることを，オーダ3の場合について具体的に示す。あわせて，ベクトル $\boldsymbol{g}_i^M$ が接線ベクトルの方向を示していることも示す。

$M=3$ のとき，式(2.25)と初期値から $F_l^{[0]}(s)$，$G_l^{[0]}(s)$ を計算する。

$$F_{i-1}^{[0]}(s) = 0 + \frac{s+i-(i-1)}{3-0-1} F_{i-1}^{[1]}(s)$$

$$= \frac{s+1}{2}$$

$$F_{i-2}^{[0]}(s) = \frac{3-0-i+(i-2)-s}{3-0-1} F_{i-1}^{[1]}(s) + 0$$

$$= \frac{1-s}{2}$$

$$G_i^{[0]}(s) = 0 + \frac{s+i-(i)}{3-0-1} G_i^{[1]}(s)$$

$$= \frac{s}{2}$$

$$G_{i-1}^{[0]}(s) = \frac{3-0-i+(i-1)-s}{3-0-1} \cdot G_i^{[1]}(s) + \frac{s+i-(i-1)}{3-0-1} G_{i-1}^{[1]}(s)$$

$$= \frac{1-2s}{2}$$

$$G_{i-2}^{[0]}(s) = \frac{3-0-i+(i-2)-s}{3-0-1} G_{i-1}^{[0]} + 0$$

$$= \frac{s-1}{2}$$

式(2.2 1), (2.2 3) , (2.2 4) と上記結果から

$$\boldsymbol{C}_i^{3}(s) = \{(\frac{1-s}{2}) + s(\frac{s-1}{2})\}\boldsymbol{P}_{i-2} + \{(\frac{s+1}{2}) + s(\frac{1-2s}{2})\}\boldsymbol{P}_{i-1} + s\cdot\frac{s}{2}\boldsymbol{P}_i$$

$$= \frac{(s-1)^2}{2}\boldsymbol{P}_{i-2} + (-s^2+s+\frac{1}{2})\boldsymbol{P}_{i-1} + \frac{s^2}{2}\boldsymbol{P}_i$$

となり，オーダ3の B−spline曲線と一致する。

次に，ベクトル $\boldsymbol{g}_i^{M}$ が接線ベクトルの方向を示していることを示す。オーダ3のとき式(2.21) から

$$\frac{d\boldsymbol{C}_i^{M}(s)}{ds} = \frac{d}{ds}\{s\cdot\boldsymbol{P}_i^{[1]} + (1-s)\cdot\boldsymbol{P}_{i-1}^{[1]}\}$$

$$= s\cdot\frac{d\boldsymbol{P}_i^{[1]}}{ds} + (1-s)\frac{d\boldsymbol{P}_{i-1}^{[1]}}{ds} + \boldsymbol{P}_i^{[1]} - \boldsymbol{P}_{i-1}^{[1]}$$

$$= s\cdot\frac{d}{ds}\{s\cdot\frac{\boldsymbol{P}_i+\boldsymbol{P}_{i-1}}{2} + (1-s)\boldsymbol{P}_{i-1}\}$$

$$+ (1-s)\frac{d}{ds}\{s\cdot\boldsymbol{P}_{i-1} + (1-s)\frac{\boldsymbol{P}_{i-1}+\boldsymbol{P}_{i-2}}{2}\} + \boldsymbol{P}_i^{[1]} - \boldsymbol{P}_{i-1}^{[1]}$$

$$= s\cdot\frac{\boldsymbol{P}_i+\boldsymbol{P}_{i-1}}{2} + (1-s)\boldsymbol{P}_{i-1} - \{s\cdot\boldsymbol{P}_{i-1} + (1-s)\frac{\boldsymbol{P}_{i-1}+\boldsymbol{P}_{i-2}}{2}\}$$

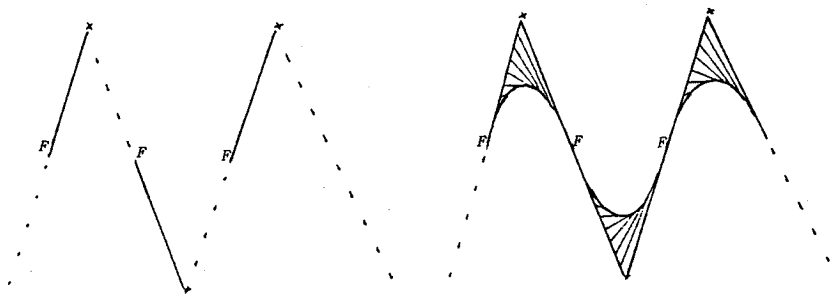
$$+ \boldsymbol{P}_i^{[1]} - \boldsymbol{P}_{i-1}^{[1]}$$

$$= \boldsymbol{P}_i^{[1]} - \boldsymbol{P}_{i-1}^{[1]} + \boldsymbol{P}_i^{[1]} - \boldsymbol{P}_{i-1}^{[1]}$$

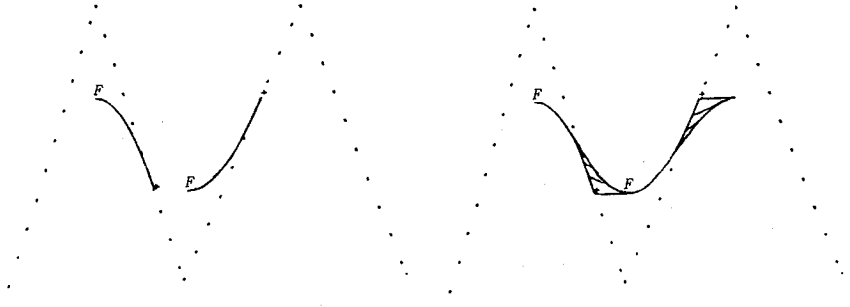
$$= 2(\boldsymbol{P}_i^{[1]} - \boldsymbol{P}_{i-1}^{[1]})$$

$$= 2\boldsymbol{g}_i^{3}$$

と証明できる。



(a)　オーダ３の場合



(b)　オーダ４の場合

図2.9　パラメータ s を変えたときのベクトル $\boldsymbol{f}_i^M$, $\boldsymbol{g}_i^M$, $\boldsymbol{C}_i^M$ の表示例

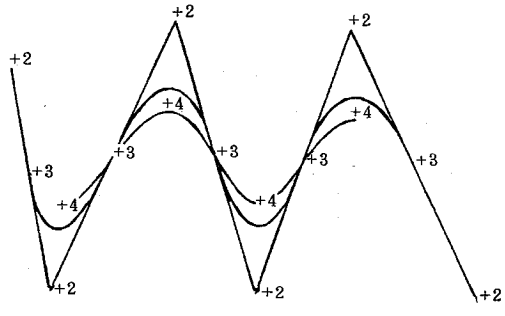


図2.10　同一の制御頂点に対するオーダ２，３，４の曲線

図2.9には，パラメータ s を変えたときのベクトル $\boldsymbol{f}_i^M$ の軌跡と，これにベクトル $\boldsymbol{g}_i^M$ を組み合わせることによって，曲線が創成される様子が図示されている。曲線セグメントの端点 $\boldsymbol{f}_i^M(0)$ と接線ベクトルを陽に取り扱うことによって，制御多角形と曲線の幾何学的関係が容易に理解されるであろう。図2.10には同じ制御頂点が与えられたときのオーダ２，３，４の曲線が示してある。

2.5 曲面の設計

本節では，数本の空間曲線および新たに導入した形状制御関数を用いて，初期曲面を創成する手法を提案する。このとき，空間曲線として，曲面形状の特徴をよく表す曲線を入力することによって，設計意図に沿う初期曲面を容易に作成することができる。以下ではこの空間曲線をアウトラインとよぶ。

2.5.1 曲面の数式表現

数本の空間曲線（アウトライン）から曲面形状を定める方法について述べる。

空間曲線は，式（2.21），（2.23），（2.24）から

$$\boldsymbol{C}_i^M(s) = \sum_{l=i-(M-1)}^{i} \boldsymbol{P}_l \left(F_l^{[0]}(s) + s \cdot G_l^{[0]}(s) \right) \qquad (2.26)$$

となる。このとき，曲面はカルテジアン積曲面（Cartesian product surface）として，パラメトリックに式（2.27）に示す形となる。

$$\boldsymbol{S}_{ij}^M(s,t) = \sum_{k=i-(M-1)}^{i} \sum_{l=j-(M-1)}^{j} \boldsymbol{P}_{kl} \left(F_k^{[0]}(s) + s \cdot G_k^{[0]}(s) \right) \cdot \left(F_l^{[0]}(t) + t \cdot G_l^{[0]}(t) \right)$$

$$0 \leqq s \leqq 1, \quad 0 \leqq t \leqq 1 \qquad (2.27)$$

ただし，$F_k^{[0]}(s)$，$G_k^{[0]}(s)$，$F_l^{[0]}(t)$，$G_l^{[0]}(t)$は，オーダMを与えると式（2.25）で計算される。したがって，曲面形状を定めるには，上式における2次元の制御ネット$\{\boldsymbol{P}_{kl}\}$を決定すればよい。

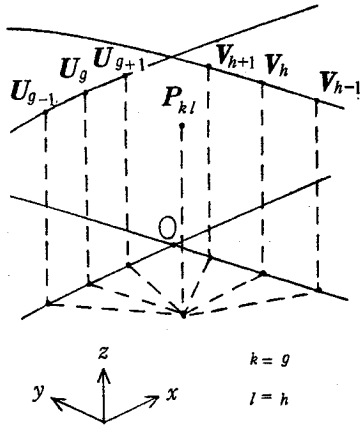


図2.11 $\boldsymbol{P}_{kl}$ の決定

本研究では，入力されたアウトラインから制御ネットを自動的に生成する方法を提案する。説明の便宜上，2本のアウトラインから2次元の制御ネット $\{P_{kl}\}$ を決定する場合を考える。2本のアウトラインは，図2.11に示すように各々制御頂点 $\{U_1, U_2, \cdots, U_g, \cdots, U_m\}$ および $\{V_1, V_2, \cdots, V_h, \cdots, V_n\}$ によって式（2.26）で表現されているものと仮定する。これらのアウトラインの制御頂点だけから制御ネット $\{P_{kl}\}$ を合成する。

説明の便宜上，入力したアウトラインは適当な座標系 $x-y-z$ を用いると，z に関して一価関数であるとする。この座標系の $z=0$ なる $x-y$ 平面に2本のアウトラインの制御多角形を投影したときの交点をOとし，座標を（x^0, y^0, 0）とする（図2.11）。

このとき

$$P_{kl} = \{x_{kl}, y_{kl}, z_{kl}\}$$
$$U_g = \{x_g{}^U, y_g{}^U, z_g{}^U\} \qquad g=1, 2, \cdots, m$$
$$V_h = \{x_h{}^V, y_h{}^V, z_h{}^V\} \qquad h=1, 2, \cdots, n$$

で表す。

ここで，x_{kl}，y_{kl} は次式で求める。

$$\left.\begin{aligned} x_{kl} &= x_g{}^U + x_h{}^V - x^0 \\ y_{kl} &= y_g{}^U + y_h{}^V - y^0 \end{aligned}\right\} \qquad (2.28)$$

ただし，添字の関係は $k=g$，$l=h$ とする。この結果，$m \times n$ 個の x_{kl}，y_{kl} が計算される。次に各 x_{kl}，y_{kl} に対応する z_{kl} を計算すれば $\{P_{kl}\}$ は決定できる。ここで，式（2.29），（2.30）で計算される $\triangle Q_{gkl}$，$\triangle R_{hkl}$ を導入する*。

$$\frac{1}{\triangle Q_{gkl}} = \frac{S^Q_{gkl}}{\sqrt{(x_{kl}-x_g{}^U)^2 + (y_{kl}-y_g{}^U)^2}} \qquad (2.29)$$

$$g=1, 2, \cdots, m$$

$$\frac{1}{\triangle R_{hkl}} = \frac{S^R_{hkl}}{\sqrt{(x_{kl}-x_h{}^V)^2 + (y_{kl}-y_h{}^V)^2}} \qquad (2.30)$$

S^Q_{gkl}，S^R_{hkl} については後述する。式（2.29），（2.30）で定まる $\triangle Q_{gkl}$，$\triangle R_{hkl}$ を用いて式（2.31）で z_{kl} を計算することにする。

$$z_{kl} = \frac{\sum_{g=1}^{m} \frac{z_g{}^U}{\triangle Q_{gkl}} + \sum_{h=1}^{n} \frac{z_h{}^V}{\triangle R_{hkl}}}{\sum_{g=1}^{m} \frac{1}{\triangle Q_{gkl}} + \sum_{h=1}^{n} \frac{1}{R_{hkl}}} \qquad (2.31)$$

＊ 式（2.28）で用いた添字の関係は $k=g$，$l=h$，式（2.29），（2.30），（2.31）においては用いない。

式(2.3 1)によって z_{kl} を計算する考え方は，式(2.2 9)，(2.3 0)で導入した計算式において $S^{Q}_{gkl}=1$，$S^{R}_{hkl}=1$ とすると，アウトラインの制御頂点のz座標値に距離の重みを付与して $\{\boldsymbol{P}_{kl}\}$ のz座標値を決定するものである。S^{Q}_{gkl}，S^{R}_{hkl} は距離の重みを制御する関数として導入したものであり，形状制御関数と名付ける。

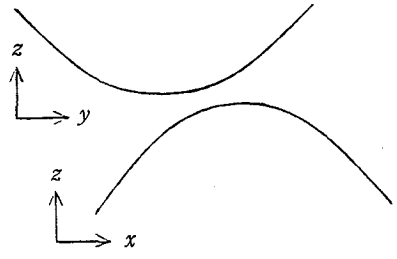
2.5.2　曲面創成の実例

式(2.2 9)，(2.3 0)における形状制御関数の一例として，式(2.3 2)，(2.3 3)を考える。

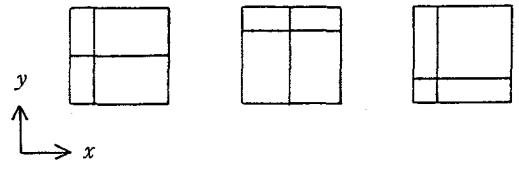
$$S^{Q}_{gkl} = \omega^{Q}\cdot\delta_{gk} \qquad (2.32)$$

$$S^{R}_{hkl} = \omega^{R}\cdot\delta_{hl} \qquad (2.33)$$

ここで，δ_{gk} と δ_{hl} はクロネッカ(Kronecker)の δ である。ω^{Q} と ω^{R} は一方のアウトラインの特徴を他方に比べて強める場合に用いる。$\{\boldsymbol{P}_{kl}\}$ の決定に際して(図2.1 1参照)，式(2.3 2)，(2.3 3)で示した形状制御関数を用いると，入力データとして与えるアウトライン

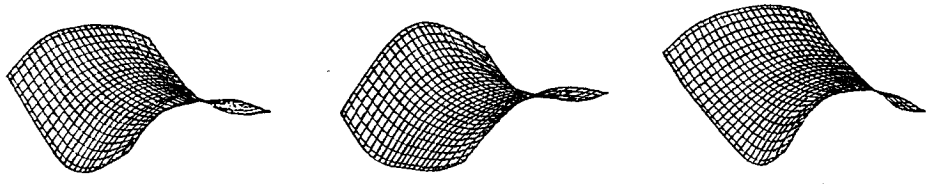


(a)　アウトラインの形状



(i)　曲面(c)の場合　(ii)　曲面(d)の場合　(iii)　曲面(e)の場合

(b)　2本のアウトラインの位置



(c)　(d)　(e)

図2.12　アウトラインの曲面上の位置を変えることによる創成曲面の変化

の制御頂点 $\{ \boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_g, \cdots, \boldsymbol{U}_m \}$ および $\{ \boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_h, \cdots, \boldsymbol{V}_n \}$ において，$\boldsymbol{U}_k, \boldsymbol{V}_l$ のみが $\boldsymbol{P}_{kl}$ の決定＊に関与する。

図2.12に，アウトラインの曲面上の位置（入力位置）を変えることによる，創成曲面の変化の一例を示す。

本例では，図2.12(a)に示す互いに直交する2本のアウトラインを用いた。図2.12(b)の(i)～(iii)に示す曲面上の位置に図2.12(a)に示すアウトラインを定義した場合に創成される曲面形状を，図2.12(c)～(e)に示す。このときの形状制御関数では，式（2.32），（2.33）において $\omega^R = \omega^Q$ としている。

次に形状制御関数による創成曲面の変化例を示す。

図2.13(f)に示した曲面上の位置に，図2.12(a)に示した形状のアウトラインを定義した場合を考える。式（2.32），（2.33）で示した形状制御関数を用い，これら2式の中の ω^Q と ω^R の比を曲面創成のための制御量とする。ω^Q と ω^R の値を，図2.13(g)に示すように変化させた場合の創成曲面の変化例を図2.13(a)～(e)に示す。

図2.13に示した創成曲面を見ると(a)→(e)へ移るに従って，次第に y 方向のアウトラインが曲面

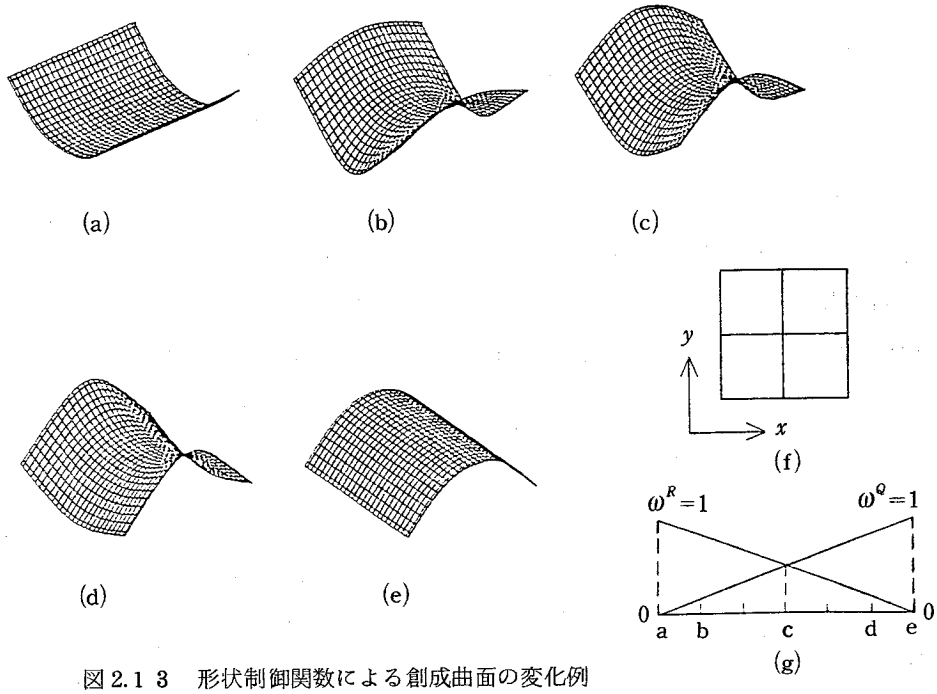


図2.13　形状制御関数による創成曲面の変化例

＊　$\{ \boldsymbol{P}_{kl} \}$ という集合を表すのではなく，$\boldsymbol{P}_{kl}$ という一つのベクトルの決定において，$\boldsymbol{U}_k$，$\boldsymbol{V}_l$ が関与することを示している。

形状の特徴を強く表す形状となっていることがわかる。(a)と(e)の場合がその極端な例で，$\omega^Q = 0$ または $\omega^R = 0$ とすることにより，一方のアウトラインの効果だけが現れ，他方のアウトラインの効果は消滅している。

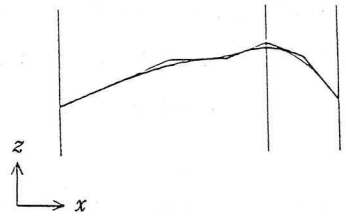
この例に見られるように，わずかのパラメータの変更によって，大域的な性質の曲面を創り出すことができる点が，ここで，提案している手法の特徴になっている。

2.6 曲面設計・加工システムの構成

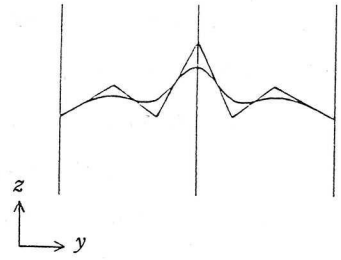
第2.4節及び第2.5節で述べた原理を応用した曲面設計・加工システムについて述べる。基本的にB－spline曲線あるいは曲面を用いているので，局所的制御も容易である。そのため，ＣＲＴディスプレイを用いた対話処理により，大域的制御と局所的制御の両方を併用して，設計過程を進める。図2.14に示す適用例に従って，この過程を順を追いながら説明する。

まず図2.14(a)，(b)に示すように，x 軸方向と y 軸方向の２本のアウトラインを定める。これらは第2.4節で述べたノットの間隔が1であるオーダ４のB－spline曲線で，対話処理により制御頂点を動かせば，形状変更は自由にできる。次に設計者の持っている大域的なイメージから，曲面上に占めるアウトラインの位置（図2.14(c)）と形状制御関数の重み ω^Q と ω^R を入力することによって，大域的なイメージの設計が行われる。

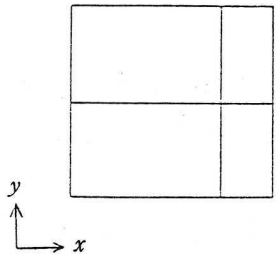
これで曲面の創成が一応できるが，一般に，大域的な制御だけで所望の曲面を創り出すことは難しいので，図2.14(d)に示すように，自動生成された制御ネットを一度画面に表示する。設計の第２段階として，制御ネットに局所的に変更を加えて，設計者の意図する曲面を得る（図2.14(e)）。基本的に，B－spline を用いた曲面であるから，局所的制御は従来から行われている手法であるが，大域的制御と併用することより効果的に利用することができる。図2.14(f)，(g)，(h)はＮＣ加工を行うための工具軌跡の生成と，実際の加工例を示したものである。



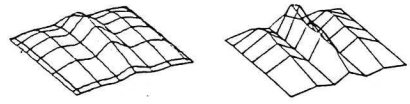
(a)　x 軸方向のアウトラインの創成



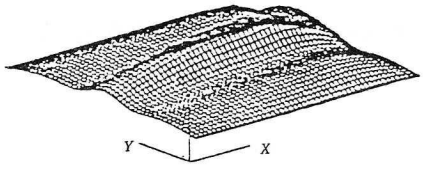
(b)　y 軸方向のアウトラインの創成



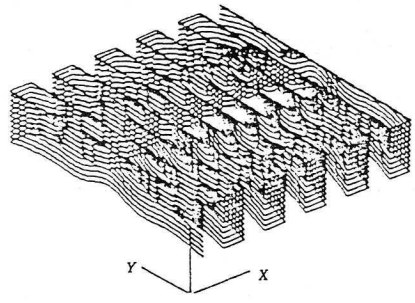
(c)　アウトラインの位置決め



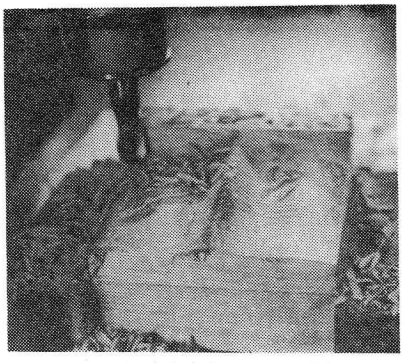
(d)　曲面形状の局所的な変更



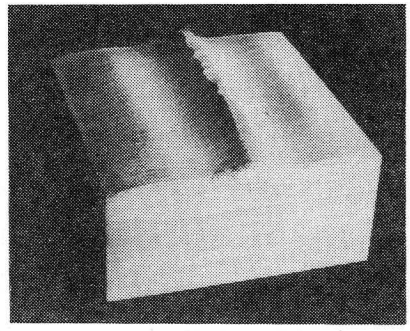
(e)　創成された曲面



(f)　工具軌跡の生成例



(g)　ＮＣ加工



(h)　加工結果

図2.14　システムの適用例

2.7 結 言

B−spline による曲面モデルを実際の応用において取り扱う場合，設計者の手元には通過点のような数値データがないことが多く，初期曲面の創成に労力が費やされることがしばしば起こる。本研究では，曲面全体の特徴を表すアウトラインの入力によって，容易に所望の形状に近い曲面を創成する一つの手法を開発した。結論を要約すると以下のようになる。

(1) 応用上有用なノットの間隔が1であるB−spline 曲線について，接線ベクトルや曲線の通過点（曲線セグメントの端点）のような幾何学的特徴をよく表す量を陽に扱う形式で，新しい計算式を導いた。

(2) アウトラインといくつかのパラメータという比較的少ない入力データを用いて，曲面の大域的制御が可能であることを示し，初期曲面の創成が容易に行えることが明らかになった。

(3) 従来から使われている制御頂点の操作による局所的制御は，大域的制御と併用して用いることができ，両手法を効果的に組み合わせることによって所望の曲面が比較的少ない労力で創成できる。

(4) 実際に曲面設計からＮＣ加工に至るまでのシステムを構築し，本研究で提案した手法の有用性を確かめた。

本章では，四角形のパッチ式のみを取扱ったが，実際の応用を考えた場合，三角形や五角形のパッチの表現も重要となる。さらに，システムを開発する場合は，マンマシン・インタフェースの入出力装置の問題，あるいは，コンピュータグラフィックスによる表示技術の問題等がある。これらの研究成果と本章で述べた自由曲面の表現式の研究成果により，意匠設計への計算機システムの導入が進んでいくものと考えられる。

参 考 文 献

1） 研野和人：自動設計法，コロナ社（1969）59．

2） S．A．Coons：Surface for Computer Aided Design of Space Forms, MIT Project MAC. TR−41（1967）．

3） P．Bézier：Numerical Control−Mathematics and Applications, John Wiley & Sons Inc., London（1972）．

4） R．F．Riesenfeld：Applications of B−spline Approximation to Geometric Problems of Computer Aided Design, Univ. of Utah UTEC−CSc−73−126（1973）．

5） D. F. Rogers and J. A. Adams : Mathematical Elements for Computer Graphics, McGraw-Hill (1976).

6） 山口富士夫： 自由曲面の設計，精密機械，43. 1 (1977) 49.

7） 山口富士夫： 形状処理工学〔Ⅰ〕，日刊工業新聞社(1982).

8） G. O. Gellert : Geometric Computing-Electric Geometry for Semiautomated Design, Machine Design, March (1965).

9） 山口富士夫： コンピュータディスプレイによる自由曲面の一設計方式(第1報)—— B-spline曲線に基づくカーブフィッティング，精密機械，43.2 (1977) 168.

10） 山口富士夫： コンピュータディスプレイによる自由曲面の一設計方式(第2報) — 曲線設計と初期曲面モデルの創成法(1)，(2)，精密機械，43.9 (1977) 1005.

11） 山口富士夫： コンピュータディスプレイによる自由曲面の一設計方式(第3報) — 初期曲面モデルの創成法(3)と曲面形状制御法，精密機械，43.10 (1977) 1141.

12） M. G. Cox : The Numerical Evaluation of B-spline, J. Inst. Maths. Applics, 10(1972)134.

13） 塩谷景一，山縣敬一，牧之内三郎： アウトラインの入力による自由曲面の設計方式，精密工学会誌，52. 3(1986)478.

第3章　工業製品に適した自由曲面の表現

3.1　緒　言

第2章では，設計者と計算機の対話により自由曲面を創成する新しい手法を提案した。この表現式は，設計者の頭の中にある曲面のイメージを計算機上に表現しやすいパラメータを持つ特徴がある。

一方，工業的応用において，曲面は図面によって要所要所の断面形状を厳密に寸法指定されることがある。この理由は，例えば金型曲面であれば，プラスチックが硬化するときの収縮率や，型分割，抜き勾配付与等の金型設計に基づいて形状が決められており，技術計算結果による厳密な寸法指定が行われるためである。この場合，曲面は四辺形になるとは限らず，さらに境界曲線の形状は複雑になる。この結果，一セグメントの曲線定義式では，境界曲線の定義が難しくなる。

本章では，図面に寸法指定された断面曲線から，指定通りの曲面を創成する手法を考える。そのために，まず，円弧のスプライン表現について考察し，従来ほとんど議論されていない有理式 B−spline を用いて厳密に定義できることを示す。しかし，有理式 B−spline は実用上いくつかの問題があるため，多項式の Bézier 式を用いて円弧を表現する手法を提案し，実用上十分な近似精度が得られることを示す。

このように，断面曲線を寸法指定通り厳密に表現すると，上述したように曲面を構成する境界曲線のセグメント数は多くなり，かつ各境界曲線のセグメント数は異る。このため，Brown の重み関数を用いて，一般Coons曲面に含まれる二つのロフト曲面の凸結合によって曲面を創成する方法を提案する。この方法を用いると，複数のセグメントをもつ自由曲面の創成が可能となり，さらに，三角形曲面や五角形曲面の定義が可能となることを示す。

3.2　円弧のスプライン近似

金型図面では，曲面の要所要所の断面曲線によって，曲面形状の指示が行われ，断面曲線は，円弧と線分によって構成されていることが多い。これらの円弧・線分は厳密に寸法指定される。つまり，図面に指示された曲面を創成するためには，寸法指定通りの断面曲線を定義しなければならない。一般に曲面は，形状表現力の優れているスプラインで表現するため，自動的に曲面の構成要素である断面曲線もスプライン表現となる。そこで，本節では，円弧のスプライン近似について考える。

3.2.1　有理式表現

円弧は式(3.1)に示す多項式では厳密に表現することはできない。[1)]　ここで，$\boldsymbol{a}_i$ ($i=0$, …, n)はベクトルである。

$$\boldsymbol{a}_n t^n + \boldsymbol{a}_{n-1} \cdot t^{n-1} + \cdots + \boldsymbol{a}_0 \qquad (3.1)$$

第2章で示した，B−spline やFerguson の式もスプラインの一種であるが，これらは，多項式であるから，厳密に円弧を表現することはできない。そこで，有理式を用いて円弧を厳密に表現する手法が提案された[2)]。以下において，有理式B−spline を用いた円弧の厳密な表現を示す[3)]。なお，第2章で考えたB−spline は多項式B−spline (Polynomial B−spline)であり，以下，本章で述べるB−splineは 有理式B−spline (Rational B−spline)であることに注意する必要がある。

以下の議論は，$x-y$平面上の第一象限にある半径1，中心角90°の円弧$\boldsymbol{A}(t)$上で具体的に考える。一般によく知られた円弧のパラメータ表現を用いると，$\boldsymbol{A}(t)$は次式で表される。

$$\begin{aligned}\boldsymbol{A}(t) &= (x(t),\ y(t),\ 0) \\ &= \left(\frac{1-t^2}{1+t^2},\ \frac{2t}{1+t^2},\ 0\right)\end{aligned} \qquad (3.2)$$

式(3.2)における各成分は有理式であるが，同次座標表現を用いると式(3.3)のように，各成分を多項式で表現することができる。通常座標で表現される円弧$\boldsymbol{A}(t)$と区別して同次座標表現での円弧を$\boldsymbol{A}^h(t)$と表すことにする。

$$\begin{aligned}\boldsymbol{A}^h(t) &= (h(t)\cdot x(t),\ h(t)\cdot y(t),\ 0,\ h(t)) \\ &= (1-t^2,\ 2t,\ 0,\ 1+t^2)\end{aligned} \qquad (3.3)$$

式(3.3)とB−spline 式の係数を比較することにより，制御頂点$\boldsymbol{P}$が定まれば，B−spline式による円弧の厳密な表現が可能となる。

まず，通常座標におけるオーダMのB−spline は式(3.4)の成分をもつ制御頂点により式(3.5)で表される[2)]。

$$\boldsymbol{P}_l = (x_l,\ y_l,\ z_l) \qquad (3.4)$$

$$\boldsymbol{C}(t) = \sum_{l=i-(M-1)}^{i} \boldsymbol{B}_l(t)\cdot \boldsymbol{P}_l \qquad (3.5)$$

ここで，$\boldsymbol{B}_l(t)$ はB−spline の基底関数である。

一方，同次座標表現のB−spline を$\boldsymbol{C}^h(t)$，制御頂点を$\boldsymbol{P}_l^h$ で表す。

$\boldsymbol{P}_l^h$ ，$\boldsymbol{C}^h(t)$ は，式(3.6)，(3.7)で表される。

$$\boldsymbol{P}_l^h = (h_l x_l, h_l y_l, h_l z_l, h_l) \qquad (3.6)$$

$$\boldsymbol{C}^h(t) = \sum_{l=i-(M-1)}^{i} \cdot \boldsymbol{B}_l(t) \cdot \boldsymbol{P}_l^h \qquad (3.7)$$

式(3.3)による多項式を用いた円弧表現と係数が比較できるのは，同じ座標表現による式(3.7)である。ここで，係数を比較するために，B−spline 式として2次式を用いる[3]。すなわち，

$$\begin{aligned}\boldsymbol{C}^h(t) &= \sum_{l=i-2}^{i} \cdot \boldsymbol{B}_l(t) \cdot \boldsymbol{P}_l^h \\ &= (1-t)^2 \boldsymbol{P}_{i-2}^h + 2t(1-t)\boldsymbol{P}_{i-1}^h + t^2 \boldsymbol{P}_i^h \end{aligned} \qquad (3.8)$$

式(3.3)と式(3.8)の係数を比較すると，

$$\begin{aligned}
1-t^2 &= (1-t)^2 \cdot h_{i-2} x_{i-2} + 2t(1-t) \cdot h_{i-1} x_{i-1} + t^2 h_i x_i \\
2t &= (1-t)^2 \cdot h_{i-2} y_{i-2} + 2t(1-t) \cdot h_{i-1} y_{i-1} + t^2 h_i y_i \\
0 &= (1-t)^2 \cdot h_{i-2} z_{i-2} + 2t(1-t) \cdot h_{i-1} z_{i-1} + t^2 h_i z_i \\
1+t^2 &= (1-t)^2 \cdot h_{i-2} + 2t(1-t) \cdot h_{i-1} + t^2 h_i
\end{aligned} \qquad (3.9)$$

式(3.9)が t の値に関係なく成り立つためには，$\boldsymbol{P}_{i-2}^h$, $\boldsymbol{P}_{i-1}^h$, $\boldsymbol{P}_i^h$ が次の値であればよい。

$$\begin{aligned}
\boldsymbol{P}_{i-2}^h &= (1, 0, 0, 1) \\
\boldsymbol{P}_{i-1}^h &= (1, 1, 0, 1) \\
\boldsymbol{P}_i^h &= (0, 2, 0, 2)
\end{aligned} \qquad (3.10)$$

すなわち，制御頂点を式(3.10)で与えれば，式(3.8)で円弧を厳密に定義できる。

ところで，式(3.8)は同次座標表現である。これを通常座標表現へ変換することを考える。式(3.4)，(3.6)から分かるように，同次座標表現は通常座標表現に同次座標値 h が乗算された形になっている。つまり，同次座標における一点を(X , Y , Z , h)とし，これに対応する通常

座標を(x, y, z)とすると次式の関係がある[4]。

$$x = \frac{X}{h}$$
$$y = \frac{Y}{h} \qquad (3.11)$$
$$z = \frac{Z}{h}$$

同次座標表現でのB−splineの一般式(3.7)に対応した通常座標でのB−splineは，式(3.11)を参照して，次式のようになる[3]。

$$\boldsymbol{R}(t) = \frac{\sum_{l=i-2}^{i} B_l(t)\cdot h_l \boldsymbol{P}_l}{\sum_{l=i-2}^{i} B_l(t)\cdot h_l} \qquad (3.12)$$

式(3.12)と式(3.8)，(3.10)から，第一象限にある中心角90°の円弧は，B−splineを用いて次式で与えられる　図3.1参照)。

$$\boldsymbol{R}(t) = \frac{(1-t)^2\boldsymbol{P}_{i-2} + 2t(1-t)\boldsymbol{P}_{i-1} + 2t^2\boldsymbol{P}_i}{(1-t)^2 + 2t(1-t) + 2t^2} \qquad (3.13)$$

式(3.12)，(3.13)は有理式であるから，有理式B−splineとよばれる[3]。

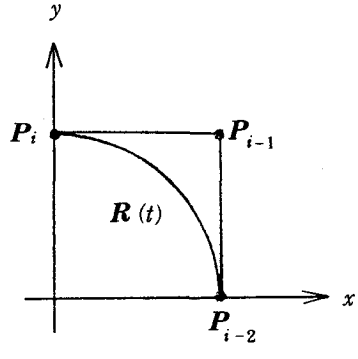


図3.1　有理式B−splineを用いた円弧の表現

3.2.2　多項式表現

前項で述べたように，多項式では円弧を厳密に表現できない。そのため，有理式B−splineを用いて，厳密な形状定義を行うシステムも開発されている[3]。

しかし，有理式スプラインを用いると次に示す問題がある。

(1) 等間隔パラメータ（式（3.13）における t）に対応する曲線上の点の間隔が場所により大きく異なる（図3.2参照）。

(2) 同次座標値 h_i というパラメータが一つ増える。

(3) 曲面も有理式表現となる。多項式スプラインに比べ曲面の相貫線計算など，各種の形状処理が複雑になる。

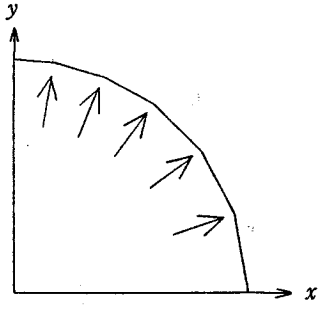


図3.2　等間隔パラメータに対応する曲線上の点（↑で示す）

そのため，多項式スプラインの方が扱いやすいといえる。ところで，曲面を扱うシステムへの応用を考えた場合，円弧は実用上の要求から決まる精度で近似できればよい。本項では，多項式スプラインを用いて円弧を近似することを考える。使用するスプライン式としては，Bézier の式を用いる。その理由は，後述するように曲線の両端点における接線ベクトルと制御頂点の関係が単純なためである。

Bézier の式は，Bernstein基底関数 $\Psi_{i,n}(t)$ を用いて次のように表される[5]。

$$\boldsymbol{C}(t) = \sum_{i=0}^{n} \Psi_{i,n}(t) \cdot \boldsymbol{P}_i \quad , \quad 0 \leqq t \leqq 1 \qquad (3.14)$$

ここで，n は曲線の次数であり，

$$\Psi_{i,n}(t) = \binom{n}{i} (1-t)^{n-i} \cdot t^i \qquad (3.15)$$

$\boldsymbol{P}_i$ は Bézier の制御多角形の頂点の位置ベクトルである。前項では，一般に知られた円弧のパラメトリックな表現式と係数を比較する方法を用いたため，2次の Bézier 式を用いた。本項では，後述するように円弧近似で，円弧の端点における接線ベクトルを条件として用いるが，2次の Bézier 式では，曲線の両端点の接線ベクトルの方向が与えられると曲線形状はただ一つに決まるので，接線ベクトルの大きさは自由に変えることができない。そのため，接線ベクトルの方向が与えられても，大きさを自由に変えることができる3次式を用いる。3次の Bézier 式を示す（本項では，特にことわらない限り，通常座標表現である）。

$$\boldsymbol{C}(t) = \Psi_{0,3}(t)\cdot\boldsymbol{P}_0 + \Psi_{1,3}(t)\cdot\boldsymbol{P}_1 + \Psi_{2,3}(t)\cdot\boldsymbol{P}_2 + \Psi_{3,3}(t)\cdot\boldsymbol{P}_3 \qquad (3.16)$$

$$\Psi_{0,3}(t) = (1-t)^3$$

$$\Psi_{1,3}(t) = 3(1-t)^2 t$$

$$\Psi_{2,3}(t) = 3(1-t)t^2$$

$$\Psi_{3,3}(t) = t^3$$

$\overrightarrow{\boldsymbol{P}_0\boldsymbol{P}_1}$, $\overrightarrow{\boldsymbol{P}_2\boldsymbol{P}_3}$ は，曲線の両端点における接線ベクトルの方向と一致している（図3.3）。

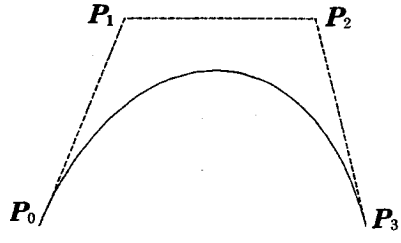


図3.3　3次の Bézier 曲線

第一象限にある中心角θ，半径1の円弧（図3.4）を考え，この円弧を式（3.16）で近似する方法を示す。

まず，Bézier 曲線の両端点は，円弧の両端点と一致しているものとし（図3.4），$t=1/2$ で円弧と Bézier 曲線が一致する条件を設定する。このとき，Bézier の制御頂点$\boldsymbol{P}_0$,$\boldsymbol{P}_1$, $\boldsymbol{P}_2$,$\boldsymbol{P}_3$ は，正の定数Kを用いて，幾何学的に次のように表せる。このとき，$\overrightarrow{\boldsymbol{P}_0\boldsymbol{P}_1}$ と$\overrightarrow{\boldsymbol{P}_2\boldsymbol{P}_3}$ の方向が円弧の接線ベクトルの方向と一致している条件も用いる。

$$\boldsymbol{P}_0 = (1,0)$$

$$\boldsymbol{P}_1 = (1,K) \qquad (3.17)$$

$$\boldsymbol{P}_2 = \boldsymbol{P}_3 + K(\sin\theta, -\cos\theta)$$

$$\boldsymbol{P}_3 = (\cos\theta, \sin\theta)$$

$t=1/2$で円弧と一致する条件から次式を得る。

$$\frac{1}{8}\cdot(1,0) + \frac{3}{8}\cdot(1,K) + \frac{3}{8}(\cos\theta + K\sin\theta, \sin\theta - K\cos\theta)$$

$$+ \frac{1}{8}(\cos\theta, \sin\theta)$$

$$= \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\right) \qquad (3.18)$$

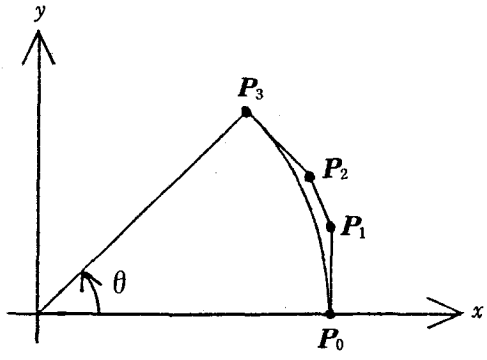


図3.4　多項式 Bézier による円弧近似

上式の左右両辺における x 座標値と y 座標値はそれぞれ等しいから，

$$1+3+3\cos\theta+3K\sin\theta+\cos\theta=8\cos\frac{\theta}{2} \qquad (3.19)$$

$$3K+3\sin\theta-3K\cos\theta+\sin\theta=8\sin\frac{\theta}{2} \qquad (3.20)$$

上の二式の各々から，同じ K の値が求まり，

$$K=\frac{4}{3}\left(\frac{2\sin\frac{\theta}{2}-\sin\theta}{1-\cos\theta}\right) \qquad (3.21)$$

式（3.17），（3.21）で定まる制御頂点を用いて，式（3.16）の B−spline 式を定めることができる。この式で半径1の円弧を近似したときの誤差を，中心角の大きさの異なるいろいろの弧について求めた結果を図3.5に示す。図3.5からわかるように，$t=0.5$ の左右に誤差の極値が一つづつある。表3.1にこの極値における誤差の値を示す。

ところで，精度の問題は，適用分野によって許容値が異なる。そこで，高精度ＮＣ加工を行うことを考え，近似式の実用性を調べる。

ＮＣ指令の最小設定単位は通常0.001㎜である。この最下位桁が近似の影響を受けない最大の円の半径を計算した結果を表3.2に示す。表3.2に示した半径以下であれば，最下位の桁に近似の影響が現われず，実用上精度の問題はない。それ以上の半径が必要な場合は，円弧を分割し，中心角を小さくすればよい。この場合，一つの円弧を二つ以上の曲線として定義することになる。

表3.1　極値における誤差

中心角	誤　差　%
40°	0.953674×10^{-4}
50°	0.762939×10^{-3}
60°	0.228882×10^{-2}
70°	0.591278×10^{-2}
80°	0.133514×10^{-1}
90°	0.270844×10^{-1}

表3.2　NC指令の最下位桁が近似の影響を受けない最大半径

中心角	半　径　mm
40°	1052.0
50°	131.0
60°	43.0
70°	16.0
80°	7.6
90°	3.7

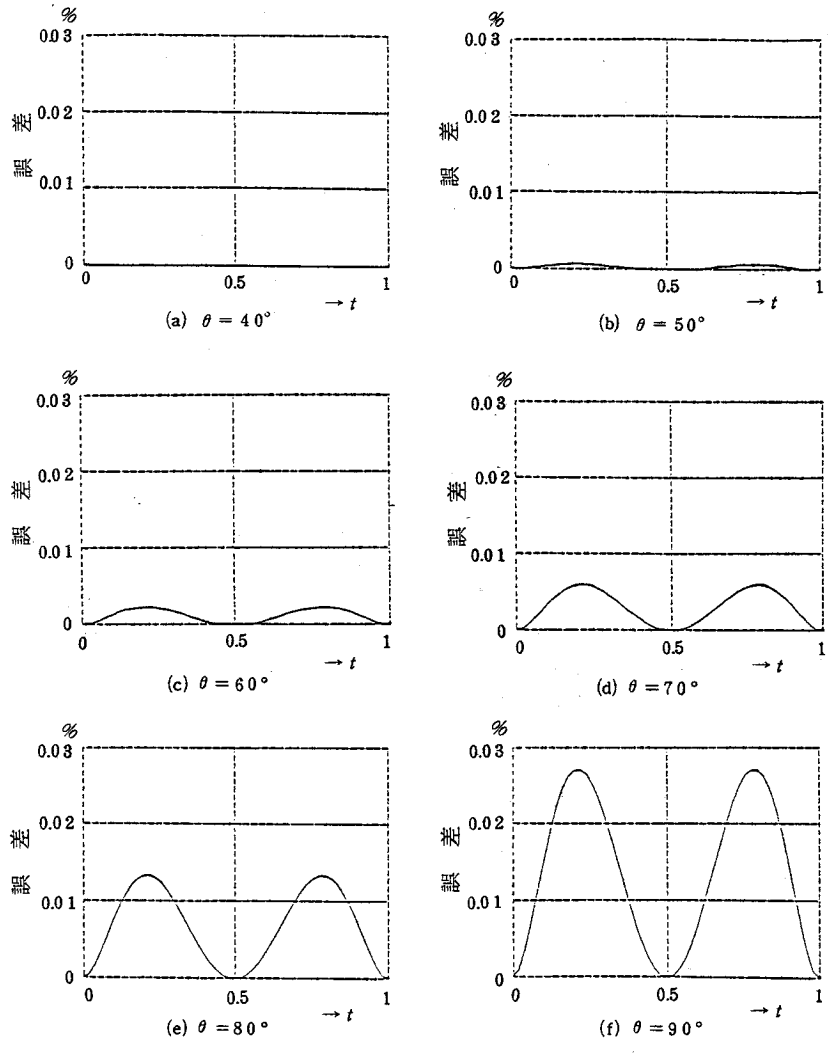


図3.5　種々の中心角の円弧をB−splineで近似したときの誤差

3.3 —セグメント境界曲線による曲面式

Bézierの式，Fergusonの式，B−splineの式など通常よく用いられる自由曲面の式は，パッチの境界曲線のセグメント数は全て1である。これらの曲面は，カルテジアン積曲面（Cartesian product surface）あるいはテンソル積曲面（tensor product surface）とよばれる表現形式を用いている[4]。

次節で述べる曲面式との比較のために，カルテジアン積曲面について検討しておく。カルテジアン積曲面の具体例として，3次のBézier式を用いる。$\boldsymbol{P}_{ij}$ を制御頂点とすれば，曲面$\boldsymbol{S}(s,t)$を次式で与えることができる[5]。

$$\boldsymbol{S}(s,t) = [\,\Psi_{0,3}(s),\ \Psi_{1,3}(s),\ \Psi_{2,3}(s),\ \Psi_{3,3}(s)\,] \times \begin{bmatrix} \boldsymbol{P}_{00} & \boldsymbol{P}_{01} & \boldsymbol{P}_{02} & \boldsymbol{P}_{03} \\ \boldsymbol{P}_{10} & \boldsymbol{P}_{11} & \boldsymbol{P}_{12} & \boldsymbol{P}_{13} \\ \boldsymbol{P}_{20} & \boldsymbol{P}_{21} & \boldsymbol{P}_{22} & \boldsymbol{P}_{23} \\ \boldsymbol{P}_{30} & \boldsymbol{P}_{31} & \boldsymbol{P}_{32} & \boldsymbol{P}_{33} \end{bmatrix} \begin{bmatrix} \Psi_{0,3}(t) \\ \Psi_{1,3}(t) \\ \Psi_{2,3}(t) \\ \Psi_{3,3}(t) \end{bmatrix} \qquad (3.22)$$

$$0 \leqq s,\ t \leqq 1$$

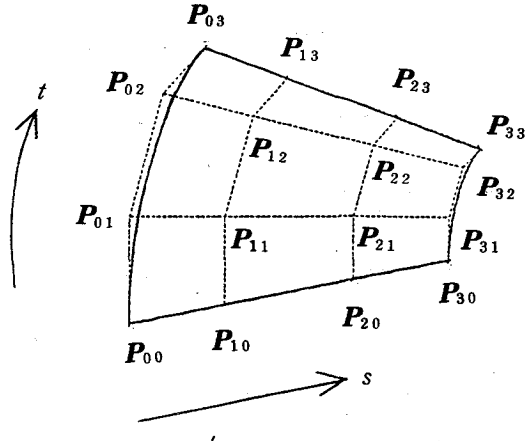


図3.6　3次のBézier 曲面における制御頂点

$\Psi_{i,n}$ は，式（3.15）で定められるBernstein 基底関数である。図3.6に曲面と $\boldsymbol{P}_{ij}$ の関係を示しておく。なお，図3.7に示すように，境界曲線を$\boldsymbol{C}_{01}$, $\boldsymbol{C}_{02}$, $\boldsymbol{C}_{10}$, $\boldsymbol{C}_{20}$ で表すと，3次のBézier 曲面では

(1) 境界曲線 $\boldsymbol{C}_{10}$は$\boldsymbol{P}_{00}$, $\boldsymbol{P}_{10}$, $\boldsymbol{P}_{20}$, $\boldsymbol{P}_{30}$,

(2) 境界曲線 $\boldsymbol{C}_{01}$は$\boldsymbol{P}_{00}$, $\boldsymbol{P}_{01}$, $\boldsymbol{P}_{02}$, $\boldsymbol{P}_{03}$,

(3) 境界曲線 C_{20} は P_{03}, P_{13}, P_{23}, P_{33} ,

(4) 境界曲線 C_{02} は P_{30}, P_{31}, P_{32}, P_{33} によって定まり,

(5) P_{11}, P_{12}, P_{21}, P_{22} は曲面の四隅におけるトゥイストベクトルを定める。

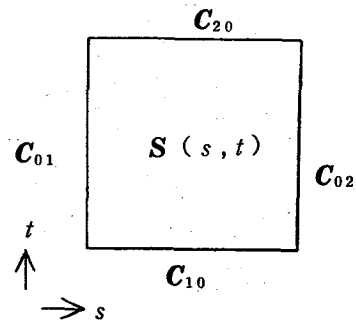


図3.7 境 界 曲 線

P_{00}, P_{03}, P_{30}, P_{33} は曲面の四隅の位置ベクトルである。また, P_{01}, P_{02}, P_{10}, P_{20}, P_{31}, P_{32}, P_{13}, P_{23} は図3.3に示したように, 曲面の四隅におけるパラメータ方向の接線ベベクトルで定められる。すなわち, 式(3.2 2)におけるすべての P_{ij} は曲面の四隅のデータで記述できるので, 曲面形状は曲面の四隅のデータで決まる。また式の形から, 境界曲線は一セグメントのみ許される。以上述べたことはFergusonの式など他の自由曲面の式をカルテジアン積表現で用いる限り共通の性質である。

仮に, 境界曲線 C_{01} が三セグメント境界曲線で与えられ, C_{02} が二セグメントで与えられたとすると, このままでは式(3.2 2)を用いてパッチの定義ができない(図3.7参照)。また, C_{01}, C_{02} 共に二セグメントの境界曲線であれば, C_{01}, C_{02} を分割して, 二つのパッチにすればよい。しかし, 分割によって生じる新たな境界曲線の定義方法に難点がある。以上のことを整理すると, C_{01}, C_{02}, C_{10}, C_{20} 境界曲線のセグメント数を各々 n_{01}, n_{02}, n_{10}, n_{20} とすると, カルテジアン積曲面は,

$$n_{01} = n_{02} \quad \text{かつ} \quad n_{10} = n_{20} \qquad (3.23)$$

のとき, $n_{01} \times n_{10}$ 個のパッチに分割し, 個々のパッチに対して曲面式を適用できる。すなわち, パッチをメッシュ状に分割する必要がある。この場合, 次に示す問題点がある。

(1) 新たに生じるパッチのコーナ点の決定手法を考える必要がある。

(2) 新たに生じた境界曲線の定義方法に難点がある。

(3) 向かい合う境界曲線のセグメント数が異なる場合, 一方の境界曲線を分割し, セグメント数を多い方に一致させる処理が必要になる。

(4) 境界曲線の一セグメントあたりの曲線長が不ぞろいになり，図3.8に示すように，等間隔パラメータに対応する s 曲線，t 曲線の密度がパッチ上の領域により大きく違う場合がある。このため，パラメータ方向の接線ベクトル方向が不規則になり，曲面の性質が悪くなる。

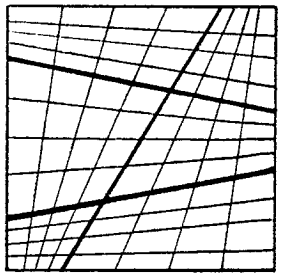


図3.8　パッチの分割（6つのパッチに分割した例）

上述した問題点を実用上の観点から検討すると次のようになる。

(1) 実際の曲面設計では，曲面の要所要所の断面曲線が円弧と線分で厳密に指定される。指定された形状を許容誤差内で近似するには，曲線のセグメント数を増やす必要がある（第3.2節）。曲面の境界曲線のセグメント数は与えられるデータから自動的に決まり，セグメント数が1と決まっているカルテジアン積曲面では，曲面形状の定義が難しくなる場合がある。

(2) 曲面形状定義後の形状確認における図形処理の手法では，一定幅のパラメータによって小領域を計算し，その領域を塗りつぶす手法がよく用いられる。この場合，パラメータ方向の曲線で小領域を作成すると，小領域の面積の大きさが不均一になり，例えばシェーディングでは，光のコントラストが不連続になるという問題を生じる。

(3) 上記(2)と関連するが，曲面のＮＣ加工を行う場合，工具の送り量，ピックフィード量を一定にする計算が複雑になり，曲面の加工精度の保証が難しくなる。

3.4　複数セグメント境界曲線を許す曲面式

工業的な応用において，曲面の各境界曲線のセグメント数が多くなる原因は，データとして与えられる個々の断面曲線が影響を与える曲面の表面積が大きいため，境界曲線が複雑になり一つの式で表現しきれなくなるからである。このような曲面を，境界曲線が全て一セグメントしか許されない曲面式で定義する場合の問題は，第3.3節で述べた。この問題に対処するため，境界曲線が複数セグメントであっても曲面を創成できる手法を示す。

3.4.1 理 論 式

現在の曲面式の基礎であるCoonsの式について考える。一般Coonsの式は次式で与えられる[7]。

$$\boldsymbol{S}(s,t) = \sum_{i=0}^{1}\sum_{k=0}^{n} H_{k,i}(s)\cdot\boldsymbol{C}^{k,0}(i,t) + \sum_{j=0}^{1}\sum_{l=0}^{n} H_{l,j}(t)\cdot\boldsymbol{C}^{0,l}(s,j) - \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{n}\sum_{l=0}^{n} H_{k,i}(s)\cdot H_{l,j}(t)\cdot\boldsymbol{C}^{k,l}(i,j) \qquad (3.24)$$

$$0 \leqq s,\ t \leqq 1$$

ただし，

$$\boldsymbol{C}^{a,b}(i,j) = \frac{\partial^{a+b}}{\partial s^{a}\,\partial t^{b}}\ \boldsymbol{C}(s,t)\Bigg|_{\substack{s=i\\ t=j}}$$

であり，$H_{k,i}(s)$, $H_{l,j}(t)$ はCoonsのブレンディング関数である[7]。

式(3.24)は，四辺形のパッチが，四つの境界曲線と境界曲線に沿う境界条件を用いて，定義できることを表している。ところで，式(3.24)が実用的に用いられるのは，$n=1$ の場合であって，$n=1$ とおいても，Coonsの式の一般的な特徴は変わらない。$n=1$ とおき，式(3.24)をマトリックス形式で表すと，

$$\boldsymbol{S}(s,t) = \begin{bmatrix} H_{0,0}(s), H_{0,1}(s), H_{1,0}(s), H_{1,1}(s) \end{bmatrix} \begin{bmatrix} \boldsymbol{C}(0,t) \\ \boldsymbol{C}(1,t) \\ \boldsymbol{C}_s(0,t) \\ \boldsymbol{C}_s(1,t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}(s,0), \boldsymbol{C}(s,1), \boldsymbol{C}_t(s,0), \boldsymbol{C}_t(s,1) \end{bmatrix} \begin{bmatrix} H_{0,0}(t) \\ H_{0,1}(t) \\ H_{1,0}(t) \\ H_{1,1}(t) \end{bmatrix} - \begin{bmatrix} H_{0,0}(s), H_{0,1}(s), H_{1,0}(s), H_{1,1}(s) \end{bmatrix}$$

$$\times \begin{bmatrix} \boldsymbol{C}(0,0) & \boldsymbol{C}(0,1) & \boldsymbol{C}_t(0,0) & \boldsymbol{C}_t(0,1) \\ \boldsymbol{C}(1,0) & \boldsymbol{C}(1,1) & \boldsymbol{C}_t(1,0) & \boldsymbol{C}_t(1,1) \\ \boldsymbol{C}_s(0,0) & \boldsymbol{C}_s(0,1) & \boldsymbol{C}_{st}(0,0) & \boldsymbol{C}_{st}(0,1) \\ \boldsymbol{C}_s(1,0) & \boldsymbol{C}_s(1,1) & \boldsymbol{C}_{st}(1,0) & \boldsymbol{C}_{st}(1,1) \end{bmatrix} \begin{bmatrix} H_{0,0}(t) \\ H_{0,1}(t) \\ H_{1,0}(t) \\ H_{1,1}(t) \end{bmatrix}$$

$$0 \leqq s,\ t \leqq 1$$

$$\boldsymbol{C}_s \equiv \frac{\partial}{\partial s}\boldsymbol{C},\quad \boldsymbol{C}_t \equiv \frac{\partial}{\partial t}\boldsymbol{C},\ \boldsymbol{C}_{st} \equiv \frac{\partial^2}{\partial s\,\partial t}\boldsymbol{C} \qquad (3.25)$$

となる。式(3.2 5)から曲面は，四つの境界曲線 $\boldsymbol{C}(0,t)$，$\boldsymbol{C}(1,t)$，$\boldsymbol{C}(s,0)$，$\boldsymbol{C}(s,1)$，それらの曲線を横切る方向の接線ベクトル $\boldsymbol{C}_s(0,t)$，$\boldsymbol{C}_s(1,t)$，$\boldsymbol{C}_t(s,0)$，$\boldsymbol{C}_t(s,1)$，トゥイストベクトル $\boldsymbol{C}_{st}(0,0)$，$\boldsymbol{C}_{st}(0,1)$，$\boldsymbol{C}_{st}(1,0)$，$\boldsymbol{C}_{st}(1,1)$ で定義できることがわかる。ところで，いかなる境界曲線でも変域が0から1であれば，式(3.2 5)中で用いることができる。つまり，複数セグメントで定義されていても，変域を0から1に変換すれば，一般 Coons の式(3.2 5)を用いて，曲面を創成できる。しかし，このままでは，トゥイストベクトルという取扱いの難しい諸量をデータとして与える必要があるため，式の改良を考える。

式(3.2 5)から，Coons の式は三つの曲面の重ね合せとなっていることが見出せる[8]。すなわち，

$$\boldsymbol{S}(s,t) = \boldsymbol{S}_{LS}(s,t) + \boldsymbol{S}_{Lt}(s,t) - \boldsymbol{S}_c(s,t) \qquad (3.26)$$

である。この三つの曲面の中で $\boldsymbol{S}_c(s,t)$ は，式の形より第3.3節で述べたカルテジアン積曲面であることが分かる。

$\boldsymbol{S}_{LS}(s,t)$，$\boldsymbol{S}_{Lt}(s,t)$ は，互いに向かい合う境界曲線と境界曲線を横切る接線ベクトルで定義されていることが分かる。この二つの曲面はロフト曲面(Loft surface)とよばれている[8](図3.9)。

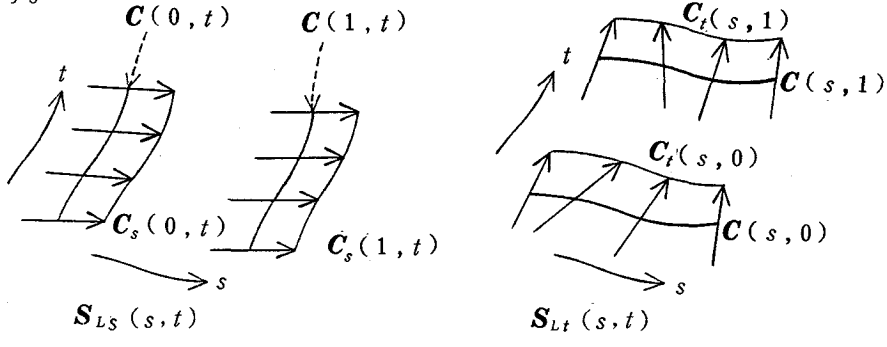


図3.9　ロフト曲面

カルテジアン積曲面は，二つのロフト曲面の和によって生じる意味のない境界曲線や，境界曲線に沿った境界条件を取り除く役目を持つ。例えば，ロフト曲面 $\boldsymbol{S}_{LS}(s,t)$ における $\boldsymbol{C}(0,t)$，$\boldsymbol{C}(1,t)$ の境界曲線はデータとして与えたものであり，創成する曲面 $\boldsymbol{S}(s,t)$ の境界曲線でもあるが，$\boldsymbol{S}_{LS}(s,0)$，$\boldsymbol{S}_{LS}(s,1)$ という余分な境界曲線が含まれる。この s 方向の境界曲線は，もう一つのロフト曲面 $\boldsymbol{S}_{Lt}(s,t)$ で与えられるから，$\boldsymbol{S}_{LS}(s,0)$，$\boldsymbol{S}_{LS}(s,1)$ の影響をなくす必要がある。この曲線データは，カルテジアン積曲面を引き差ることで取り除かれる。すなわち，カルテジアン積曲面の代わりに，二つのロフト曲面の和によって生じる余分な境界曲線などの影響をなくす方法を考えると，トゥイストベクトルを持つカルテジアン積曲面を用いなくてすむ。

そこで，上述した意味のない境界曲線や境界曲線に沿った境界条件を二つのロフト曲面の凸結合によって除去することを考え[9)]，

$$\boldsymbol{S}(s,t)=\alpha(s,t)\cdot\boldsymbol{S}_{Lt}(s,t)+\beta(s,t)\cdot\boldsymbol{S}_{LS}(s,t) \qquad (3.27)$$

$$0\leqq s,\ t\leqq 1$$

と書く。ここで

$$\alpha(s,t)=\frac{s^2(1-s)^2}{s^2(1-s)^2+t^2(1-t)^2}$$

$$\beta(s,t)=\frac{t^2(1-t)^2}{s^2(1-s)^2+t^2(1-t)^2} \qquad (3.28)$$

である。$\alpha(s,t)$，$\beta(s,t)$ は Brown が twist compatibility condition という別の目的で用いた重み関数である[10)]。図3.10に $\alpha(s,t)$ と $\beta(s,t)$ を示す。図からも分かるように，$\alpha(s,t)+\beta(s,t)=1$ であり，境界曲線上では，向かい合う辺を対として，その値が1か0であるため，前述の余分な境界曲線は除去し，しかも，与えた境界曲線はそのまま曲面形状に反映することができる。

以上から，$n=1$ の一般 Coons 曲面である式(3.25)の第一項，第二項のロフト曲面を用いて，式(3.27)で曲面を定義すれば，複数セグメントの境界曲線であっても，曲面の創成ができる。また，トゥイストベクトルをデータとして与える必要はない。

なお，前述したように境界曲線は第3.2.2項で示した手法で定義するが，図3.11に示すように変域の変更が必要となる。

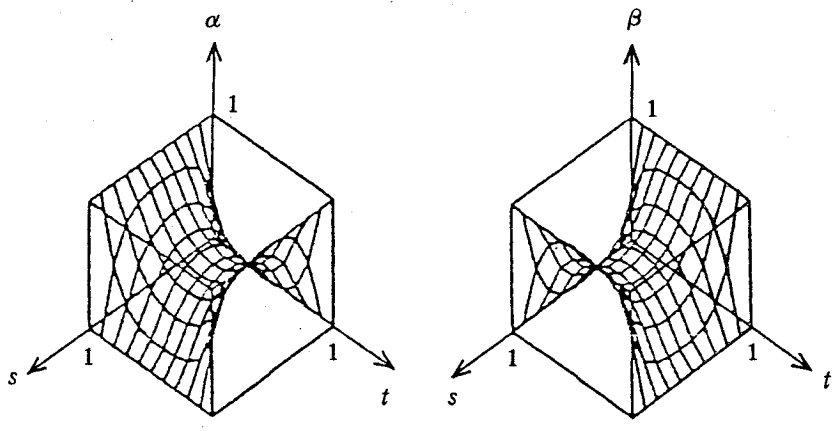


図3.10　$\alpha(s,t)$と$\beta(s,t)$

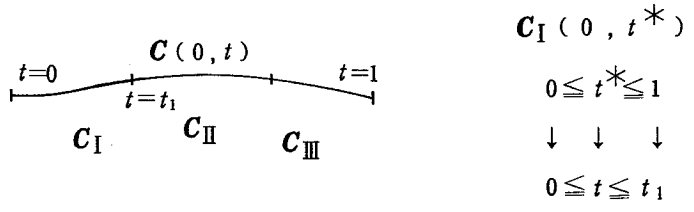


図3.11　$C(0,t)$を例にした変域の変更

3.4.2　曲面表現能力

複数セグメントの境界曲線を扱えるパッチ式（3.27）を用いると，応用上次に示す曲面が取扱えるようになる[9]。

(i)　式（3.27）をそのまま用いる四辺形曲面

(ii)　$\alpha(s,t)=1$，$\beta(s,t)=0$または，$\alpha(s,t)=0$，$\beta(s,t)=1$のロフト曲面

(iii)　境界曲線の一つを縮退させた三角形曲面

(iv)　境界曲線上の尖点（cusp）を曲面のコーナ点と考えるn角形曲面（$n\geqq 5$）

以下において，三角形曲面と，n角形曲面について説明を補足しておく。

(1)　三角形曲面

工業製品によく含まれる形状のコーナー部の曲面は，通常，三角形となる。そのため，三角形のパッチ専用の曲面式の研究も行われた[11]。境界曲線の一つを縮退させることにより三角形パッチを表現する場合，パッチ式としては四辺形のものをそのまま用いる[12]。この手法は，従来の四辺形パッチ式でも適用でき，式（3.27）に限ったものではない。縮退によって三角形曲面を表現する場合，縮退点における法線ベクトルの連続性などの問題を生じる可能性があるが，以下に

示す特徴がある。

(i) 四辺形曲面と同様に扱えるため，各種形状処理アルゴリズムも，専用に開発する必要がない。

(ii) 隣接する n 角形（ $n \geqq 4$ ）曲面との接続も容易で，応用上連続性の問題も生じない。

図 3.1 2 に三角形曲面の創成例を示す。

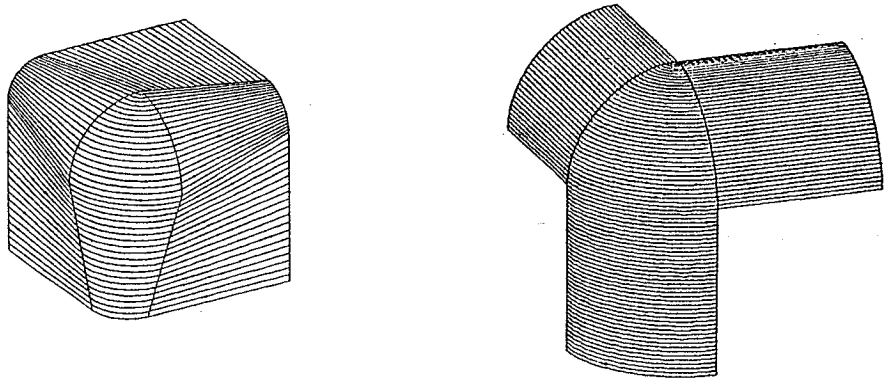


図 3.1 2　三角形曲面創成例

(2) n 角形曲面

境界曲線のセグメント数が，二以上でも許されるので，一本の境界曲線中に尖点を含めることができる。すなわち，セグメントの端点では，隣接セグメントの端点と接続だけされていればよいのであって，接続ベクトルは連続である必要はない。したがって，セグメントの端点を尖点に設定することができる。

セグメントの端点が尖点の場合，この端点は見かけ上曲面のコーナ点と考えられるので[13]，式（ 3.2 7 ）の四角形曲面の式で，n 角形曲面の表現ができる。図 3.1 3 に五角形曲面の創成例を示す。

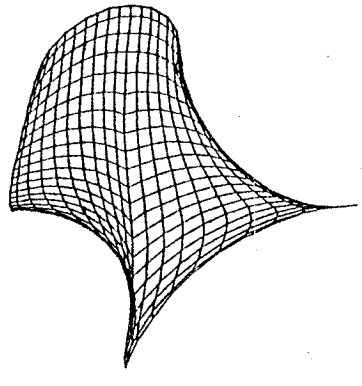


図 3.1 3　五角形曲面創成例

なお，尖点は一セグメントのBézier 式など他の一セグメントの式でも創成できるが，曲線の形状が指定されている場合は，一セグメントの式で曲面のコーナ点と創成した尖点を一致させ，しかも与えられた曲線形状を満たすことは難しい。

3.5 結 言

本章では，図面などによって要所要所の断面形状が指定される場合の自由曲面の表現式について検討を行った。この断面形状は創成する曲面の境界曲線として選択されるが，円弧と線分で構成された曲線が多いため，円弧のスプライン式による表現方法が問題となる。また，工業的な応用においては，与えられた境界曲線によって定められる曲面の面積が大きいため，境界曲線が複雑な形状となる。そのため，境界曲線を一セグメントで定義するのは難しく，カルテジアン積による曲面式では，曲面の定義が難しい。そこで，複数セグメント境界曲線をもつ自由曲面式の一手法を提案した。以下に結果を要約する。

(1) 通常座標表現での3次のBézier 式で，円弧を近似する場合の制御頂点を示し，その近似精度を示した。得られた結果は，中心角40°の円弧で誤差は最大で0.95×10^{-4}%であり，実用上十分な精度が得られる。

(2) Brownの重み関数を用いて，$n=1$の一般Coons 曲面に含まれる二つのロフト曲面の凸結合を行い，曲面を創成する方法を示した。この表現式を用いると，複数セグメントをもつ自由曲面の創成が可能となる。

(3) ここで示した曲面式は応用上，三角形曲面や五角形曲面の定義も可能である。

参 考 文 献

1) 山口富士夫： 形状処理工学〔Ⅱ〕，日刊工業新聞社(1982)．

2) K. J. Versprille: Computer－Aided Design Applications of the Rational B－spline Approximation Form, Ph. D. dissertation, Syracuse Univ., Syracuse. N.Y., Feb.(1975).

3) W. Tiller: Rational B－splines for curve and surface Representation, IEEE Computer Graphics and Applications, 3(1983) 61.

4) D. F. Rogers and J. A. Adams: Mathematical elements for Computer Graphics, McGraw－Hill (1976).

5) P. Bézier : Numerical Control－Mathematics and Applications, John Wiley & Sons Inc., London (1972).

6) G. J. Peters: Interactive Computer Graphics Application of the Parametric Bi－Cubic Surface to Engineering Design Problem, Computer Aided Geometric Design, Academic Press(1974)259.

7) S. A. Coons: Surface for Computer－Aided Design of Space Forms, M. I. T. Project MAC, MAC－TR－41, June (1967).

8) A. R. Forrest: On Coons and other Methods for the Representation of curved Surfaces, Computer Graphics and Image Processing, 1 (1972).

9) 塩谷景一 : 複数セグメント境界曲線をもつ自由曲面式の一手法, 昭和61年度精密工学会学術講演論文集(1986) 407.

10) R. E. Barnhill and J. H. Brown: A New Twist in Computer Aided Geometric Design, Computer Graphics and Image processing, 8 (1978).

11) R. E. Barnhill: Smooth Interpolation over Triangles, Computer Aided Geometric Design, Academic Press(1974)45.

12) 山口富士夫 : 形状処理工学〔Ⅰ〕, 日刊工業新聞社(1982).

13) S. A. Coons: Surface Patches and B－spline Curves, Computer Aided Geometric Design, Academic Press(1974)1.

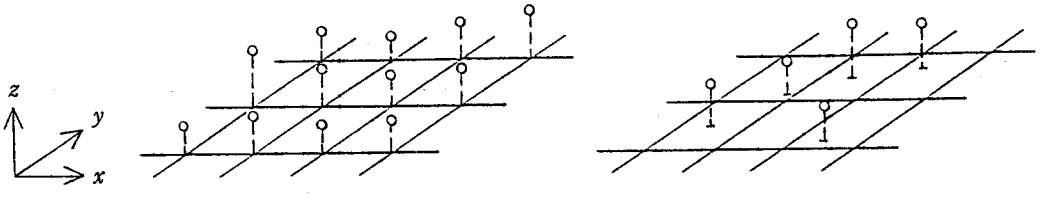
第4章　点群からの自由曲面の創成

4.1　緒　言

第2章，第3章では，クレイモデルなどの実体モデルを用いないで，直接，自由曲面を創成する手法を示した。この場合，自由曲面を定義するために与えられるデータは，数本の曲線である。本章では，例えば，3次元測定機を用いて実体モデルを測定することによって得られる点群を入力データとし，これらの点を通過する自由曲面を創成する方法を示す。

点群データから曲面を創成する手法の必要性は，増加する傾向にある。例えば，計算機を用いて，空気抵抗の少ないプロペラや飛しょう体などの設計を行ったり，マイクロ波アンテナ形状を設計する場合に，計算結果が点群としてのみ得られる。製品として完成させるためには，曲面の定義が必要となり，点群データから曲面を創成する手法が必要になる。曲面の定義が必要なのは，これらの形状が，ＮＣフライスによる切削加工で作成されているからであり，また，性能評価でも曲面のデータを用いるためである。ところで，Coons，Bézier などの曲面式を用いて自由曲面を創成する場合には，与えられた各点の $x\ y$ 平面への投影点は，基本的に格子状に並んでいる必要がある。すなわち，四辺形パッチが自由曲面を創成するための基本要素である（図4.1(a)参照）。したがって，図4.1(b)に示すように，与えられた各点の $x\ y$ 平面への投影点が $x\ y$ 平面上で任意に分布する（以下簡単に「任意に分布する点」と呼ぶ）場合には，これら各点の座標値をそのまま用いてCoons などの曲面式を直ちに決定することはできない。

Barnhill らは[1),2)]，三角形パッチを曲面の基本要素とする手法を提案している。しかし，三角形パッチは曲面式の種類が少く，曲面創成後の各種形状処理も特有なものになる。さらに，与えた点をパッチ頂点とするため，パッチごとに大きさが不規則になるなどの問題が存在する。そこで，曲面創成後の処理や，他の曲面形状定義システムとの結合も考慮し，任意に分布した点群を近似的に等価な格子状の点群におきかえて曲面を創成することを考える。本手法は，点群を補間する空間曲線を創成し，平面あるいは線分との交点を求めるプロセスの繰り返しで構成されている。そのため，高速処理が可能で，また，点群を補間する曲線式として種々の理論式を用いることができ，創成する曲面の特性に応じた使い方ができる。



(a)　格子状の点群　　(b)　任意分布点群

図4.1　分布点群と四辺形パッチ

4.2 任意に分布した点群から格子状の点群を求める手法[3)]

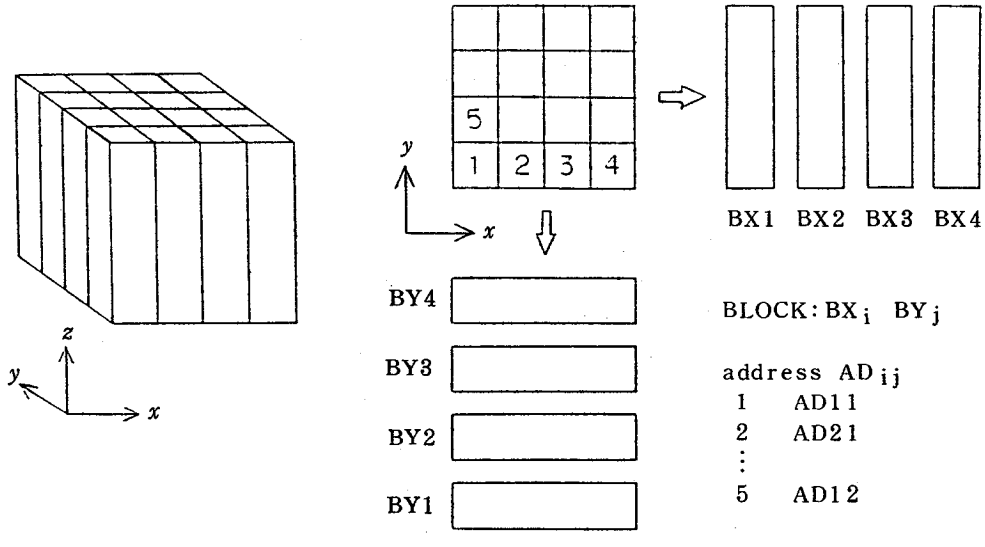
提案する曲面創成法は次の二段階で構成される。

(1) 任意に分布した点群を用いて，近似的に等価な格子状の点群を求める。

(2) この格子状の点群のデータを用い，一般の曲面理論式に基づいて，四辺形パッチ内部を補間する。

段階(2)では，Coons，Bézier，B-splineなどの理論式を使用する。段階(1)の手法については，5×5の25格子点群を求める計算例を用いながら以下に説明する。

まず，任意に分布した点群を内部に含む直方体を考える。$x-y-z$の座標系は，直方体の稜線方向に設定する。次に，x＝一定の平面（yz平面に平行な平面），y＝一定の平面（zx平面に平行な平面）で直方体を適当に分割する（図4.2(a)では，x＝一定の平面で4分割，y＝一定の平面で4分割の計16分割である）。すなわち，この直方体をBX_i ブロックとBY_jブロック（i，j＝1，2，…）の領域に分割する。この直方体をxy平面に投影すると図4.2(b)のような格子状パターンが得られる。ここで，x＝一定，y＝一定の平面で囲まれた小領域をアドレスとよぶことにし，AD_{ij}とアドレスの識別名をつける。図4.2(b)の例では，BX_iブロックはi＝1～4，BY_jブロックもj＝1～4，アドレスの数は16である。格子状に並んだ点群を次の手順で求める。個々のブロック内において，任意に分布する点群を通過する空間曲線をそれぞれ求める。例えば，図4.3に示すように，BX_i ブロックが4個，BY_j ブロックも4個の場合，y方向，x方向ともに4本ずつの空間曲線を求める。



(a) 任意に分布した点群を内部に含む直方体

(b) 直方体をxy平面に投影して得られる格子状のパターン

図4.2 点群を内部に含む直方体

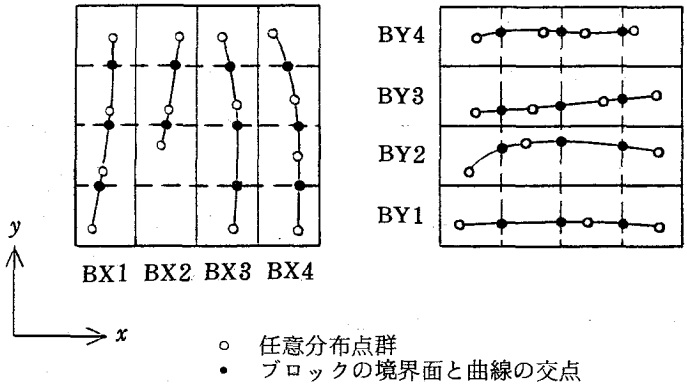


図4.3　任意に分布する点群を通過する空間曲線

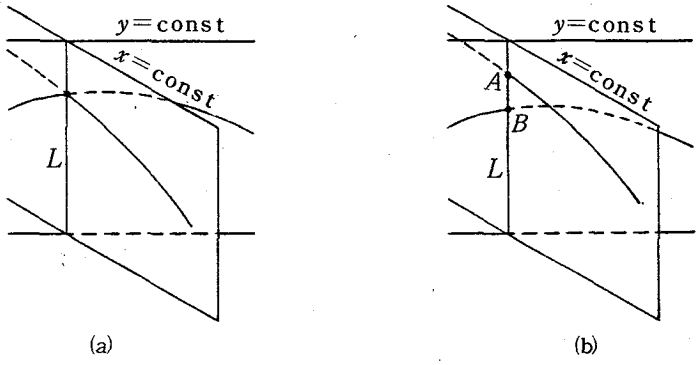


図4.4　格子状に並ぶ点群を求める手法

この結果，BX_i ブロック内に創成された空間曲線は BY_j ブロックの境界面（すなわち y ＝一定の平面）と交わり，BY_j ブロック内に創成された空間曲線は BX_i ブロックの境界面（x ＝一定の平面）と交わる（図4.3における黒丸は交点，点線は他方向ブロックの境界面を表している）。

次に，各ブロックの境界面上に得られた交点（図4.3の黒丸）を補間点とする補間曲線を個々の境界面上で求める。すなわち，図4.2(a)の直方体を分割する平面上に曲線が得られたが，これらの曲線を創成したい自由曲面の近似的な断面曲線と見なすことになる。

この結果，図4.4に示すように断面曲線と線分 L（図4.2における x ＝一定の平面と y ＝一定の平面との交線）との交点が求められる。実際には，x ＝一定の平面上の断面の断面曲線と線分 L との交点と，y ＝一定の平面上の断面曲線と線分 L との交点とは，図4.4(b)の点 A，B に示すように一般には一致しない。したがって，点 A，B の座標値の平均をとり図4.4(a)のように交点を求める。この点は，近似的ではあるが，格子状に並ぶ曲面上の点として得られたものである。

4.3 本手法の検討

まず，絶対誤差を次の手順で求める。

(1) x y 平面上に任意に分布する点群を，乱数を用いて定める。ただし，ここでは

$$0 \leqq x \ , \ y \leqq 10$$

とする。表4.1は，テストに用いた$F1$～$F4$の4種類の曲面（関数$z=f(x, y)$）を示しているが，これらの式によって各点のz座標値をそれぞれ求める。図4.5は，テストに用いた4つの曲面を図示している。

(2) (1)の手順で得られた任意分布の点群から，格子状に並ぶ点群の座標値を，前章に述べた方法で計算する。

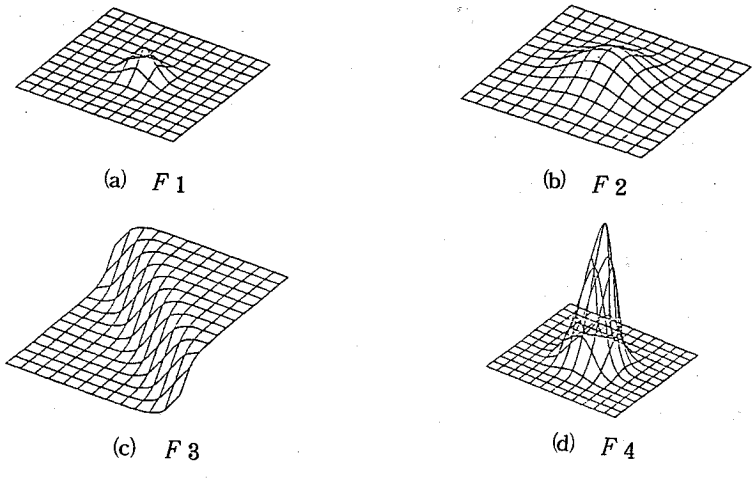
(3) 各格子点における正しいz座標値は，表4.1の曲面の式についてそれぞれ計算する。

(4) (3)の手順で得られた格子点のz座標値と，(2)の手順で推定された格子点のz座標値との差の絶対値を求める。

(5) 手順(4)を，すべての格子点に対して行い，得られた値の平均値を求める。以下，これを絶対誤差と呼ぶ。

表4.1 テストに用いた曲面の式

$F1$	$z = \exp(-((x-5)^2+(y-5)^2))$
$F2$	$z = \exp(-((x-5)^2+(y-5)^2)/4)$
$F3$	$z = \tanh(x+y-11)$
$F4$	$z = 10\exp(-((x-5)^2+(y-5)^2)/2)$



(a) $F1$　(b) $F2$　(c) $F3$　(d) $F4$

図4.5 テストに用いた曲面

次に，本手法による計算結果を検討するため，

(1) 格子点の数を一定にして，任意分布の点の個数を変化させ，絶対誤差の変化を調べる。

(2) 任意分布の点の個数を一定にして，格子点の個数を変化させ，絶対誤差の変化を調べる。

4.3.1 空間曲線の創成

第4.2節で述べたように，点群を通過する空間曲線を創成する（すなわち，点群を補間する）ことが，本手法の中心となっている。以下の数値例では，線形補間，B−spline，Fergusonの三式を用いて点群を補間し，絶対誤差を比較する。

以下に説明する空間曲線$\boldsymbol{C}_i(t)$は，式（4.1）を示すものである。

$$\boldsymbol{C}_i(t)=(x_i(t),y_i(t),z_i(t)) \qquad (4.1)$$
$$i=0,1,2,\cdots \quad 0\leqq t\leqq 1$$

1） 線形補間の式

隣り合う2個の点を表す位置ベクトルを$\boldsymbol{a}_i$，$\boldsymbol{a}_{i+1}$とする。このとき，$\boldsymbol{a}_i$と$\boldsymbol{a}_{i+1}$の間の線形補間式は，次式で表される。

$$\boldsymbol{C}_i(t)=\boldsymbol{a}_{i+1}\cdot t+\boldsymbol{a}_i\cdot(1-t) \qquad (4.2)$$
$$i=0,1,2,\cdots \quad 0\leqq t\leqq 1$$

2） C^2級B−splineの式[4) 5)]

形状制御多角形（Polygon）の頂点を表す位置ベクトルを$\boldsymbol{P}_i$とすると

$$\boldsymbol{C}_i(t)=\boldsymbol{P}_{i-3}\cdot B_{-3}(t)+\boldsymbol{P}_{i-2}\cdot B_{-2}(t)+\boldsymbol{P}_{i-1}\cdot B_{-1}(t)+\boldsymbol{P}_i\cdot B_0(t)$$
$$i=0,1,2,\cdots \quad 0\leqq t\leqq 1 \qquad (4.3)$$

$$B_{-3}(t)=-\frac{1}{6}t^3+\frac{1}{2}t^2-\frac{1}{2}t+\frac{1}{6}$$
$$B_{-2}(t)=\frac{1}{2}t^3-t^2+\frac{2}{3}$$
$$B_{-1}(t)=-\frac{1}{2}t^3+\frac{1}{2}t^2+\frac{1}{2}t+\frac{1}{6}$$
$$B_0(t)=\frac{1}{6}t^3$$

$\boldsymbol{P}_i$は，補間する点群の位置ベクトルを$\boldsymbol{a}_i$とすると，次式で計算される。

$$BV=6A$$

ここで，B，V，Aはマトリクスである。

$$B = \begin{bmatrix} 5 & 1 & & & & \\ 1 & 4 & 1 & & 0 & \\ & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 \\ & 0 & & & 1 & 5 \end{bmatrix}$$

$$V = \begin{bmatrix} \boldsymbol{P}_{-2} \\ \boldsymbol{P}_{-1} \\ \vdots \\ \boldsymbol{P}_{n-2} \end{bmatrix} \quad ; \quad A = \begin{bmatrix} \boldsymbol{a}_0 \\ \boldsymbol{a}_{-1} \\ \vdots \\ \boldsymbol{a}_n \end{bmatrix}$$

3） Fergusonの式[6)]

隣り合う2個の点を表す位置ベクトルを$\boldsymbol{a}_i$，$\boldsymbol{a}_{i+1}$，これらの2点における接線ベクトルをそれぞれ$\boldsymbol{T}_i$，$\boldsymbol{T}_{i+1}$とおくと，

$$\boldsymbol{C}_i(t) = \boldsymbol{a}_i \cdot H_1(t) + \boldsymbol{a}_{i+1} \cdot H_2(t) + \boldsymbol{T}_i \cdot H_3(t) + \boldsymbol{T}_{i+1} \cdot H_4(t) \qquad (4.4)$$

$$i = 0, 1, 2, \cdots \qquad 0 \leqq t \leqq 1$$

$$H_1(t) = 2t^3 - 3t^2 + 1$$

$$H_2(t) = -2t^3 + 3t^2$$

$$H_3(t) = t^3 - 2t^2 + t$$

$$H_4(t) = t^3 - t^2$$

となる。

4.3.2 数値例

数値例1（曲面の種類＊を変化させた場合）

図4.6に曲面の種類と絶対誤差の関係を示す。F1，F2などは，表4.1に示したように，曲面

＊ ここでいう「曲面の種類」は，「創成したい曲面」のことである。四辺形パッチに対する補間式Coons，B−splineなどの曲面式を示すのではない。

の種類を表す記号である。任意に分布した点の個数を３００，格子点の個数を１３×１３とし，任意に分布する３００個の点群を五種類用意して，各々の点群から絶対誤差を求めた。図4.6の縦軸は，求めた５種類の絶対誤差の平均値を示している。

Fergusonの式を補間式に用いた場合，もっとも絶対誤差は小さいことが図4.6から分かる。図4.5に示すように，Ｆ１はなだらかな曲面で，Ｆ４はかなり急激な変化を示す曲面である。そのため，Ｆ１は線形補間式を用いても，Fergusonの式を補間式として用いた場合と比較して，誤差の増大は少ない。Ｆ４では誤差の増大がかなり見られる。

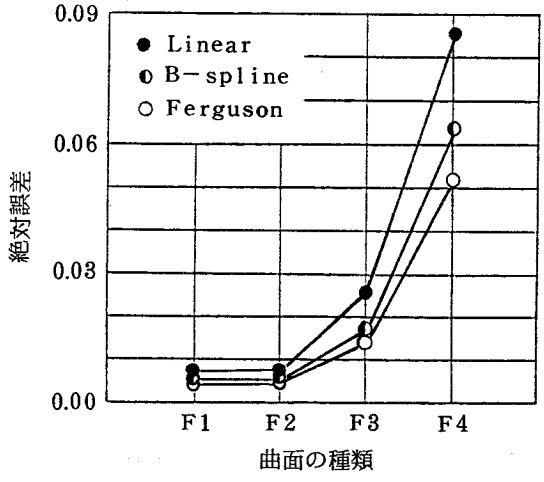


図4.6　曲面の種類と絶対誤差の関係

数値例２（格子点の個数を一定にした場合）

図4.2に示したアドレス数で１４４，ブロック数でｉ＝１２，ｊ＝１２である格子点の個数を一定数１３×１３にしたときの，任意に分布した点の個数と絶対誤差の関係を図4.7に示す。また，任意分布の点の個数と実行に要した計算時間の関係を図4.8に示す。使用した計算機は13MIPSの処理能力をもつ。各データは，五種類の分布点群を用いて，五回計算を行った平均値である。テストに用いた曲面は，表4.1のF3である。

格子点の個数を一定にした場合には，任意分布の点の個数がある値以上になると，絶対誤差の減少は期待できないことが図4.7から分かる。格子点の個数を１３×１３とした本実例では，任意分布の点の個数を５００以上増加しても絶対誤差の値はほぼ一定である。

計算時間は，線形補間の式を用いた場合に最も短く，任意に分布した点の個数に比例することが図4.8から明らかである。

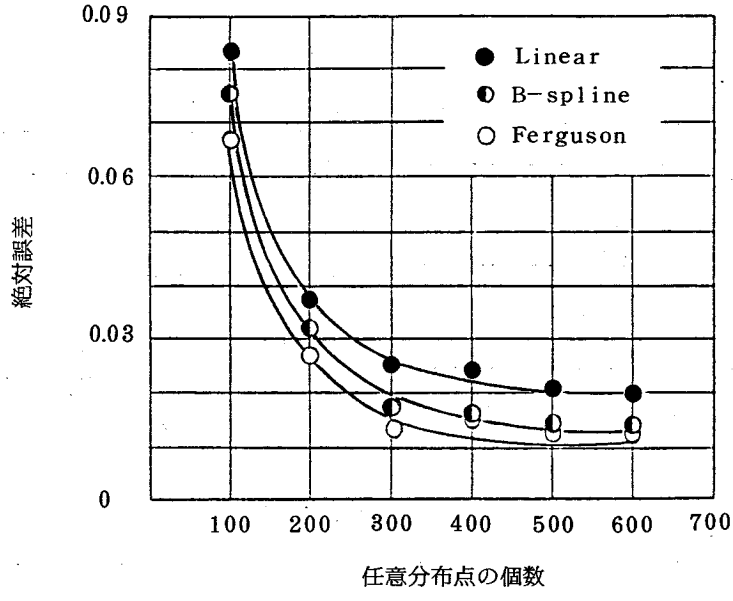


図 4.7　任意に分布した点の個数と絶対誤差の関係

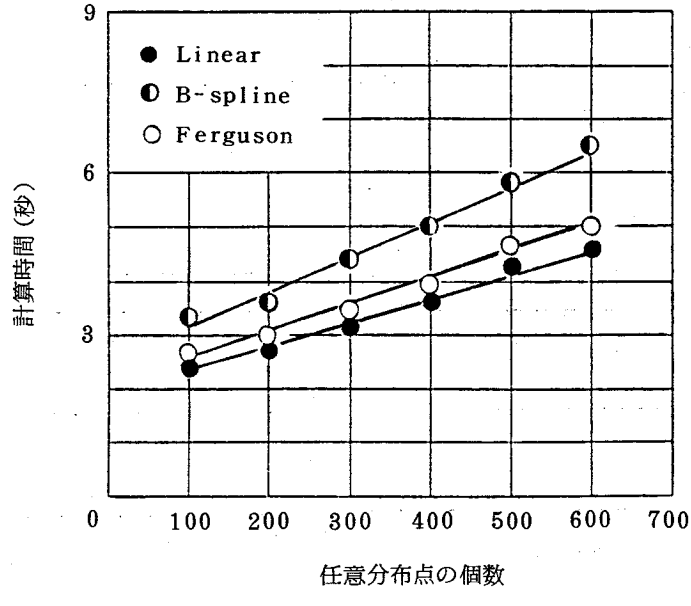


図 4.8　任意に分布した点の個数と計算時間の関係

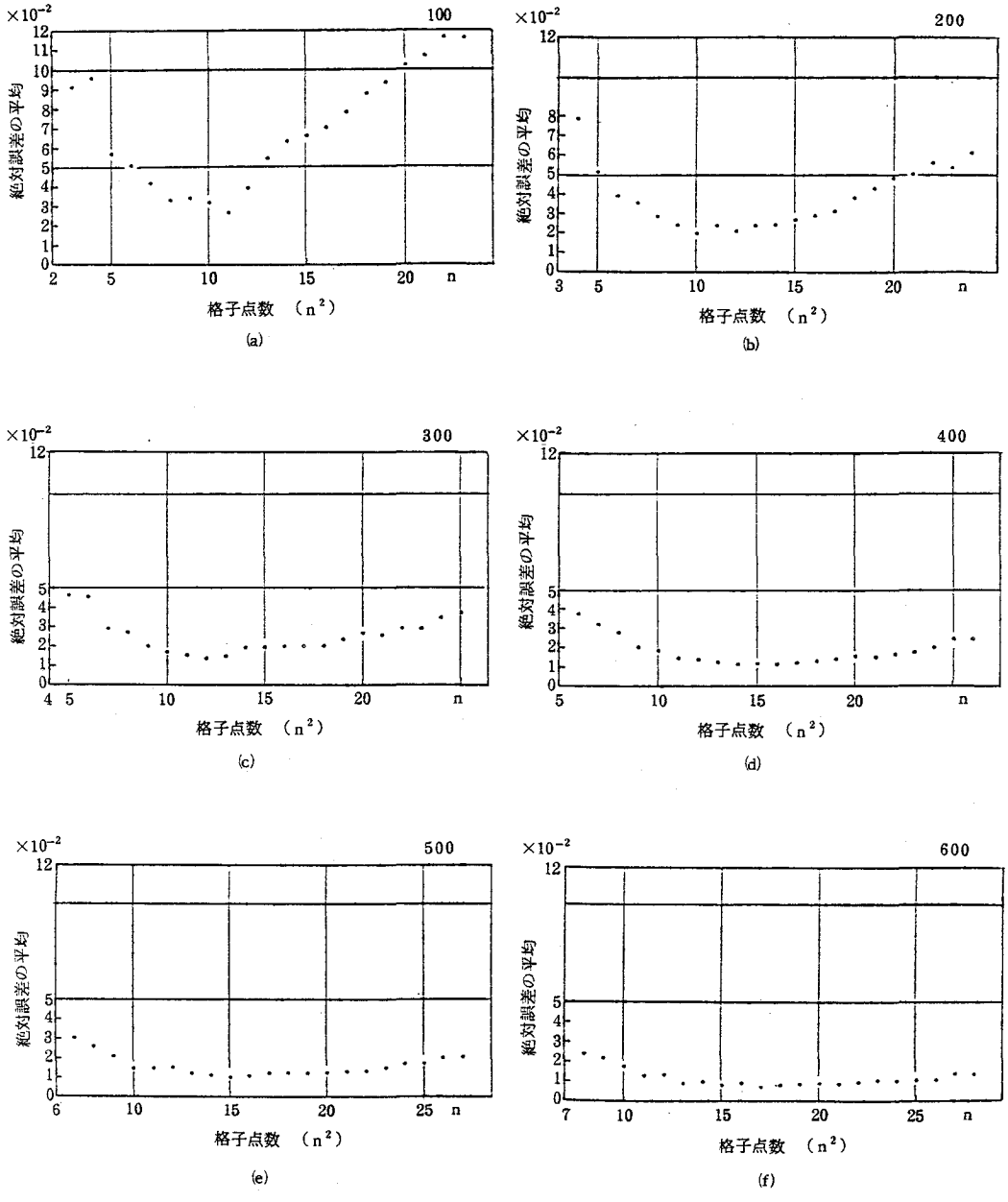


図4.9　任意分布の点の個数を一定にしたときの格子点の個数と絶対誤差の平均

数値例3（任意分布の点の個数を一定にした場合）

任意分布の点の個数を一定（＝１００個，２００個，３００個，・・・，６００個）にしたときの，格子点の個数（アドレス数，ブロック数）と絶対誤差の関係を図4.9に示す。各データは，ほかの数値例と同様に，五種類の任意分布の点群を用いて絶対誤差を求め，それらの平均をとったものである。テストに用いた曲面は表4.1のＦ３であり，補間式としてはFergusonの式を用いた。

図4.9より，任意に分布した点の個数が一定の場合，格子点の個数が少なくても多くても，絶対誤差の増加が見られる。格子点の個数が少ない場合に誤差が増大することは，一般に考えられることである。格子点の個数が多い場合には，推定したい格子点の個数に対して，データとして与えられる分布点の個数が少なすぎるために，絶対誤差が増加するものと考えられる。

4.4 結　言

任意に分布した点群を近似的に等価な格子状の点群におきかえ，曲面を創成する一手法を示し，次の結果が得られた。

(1) 本手法は，点群を補間する空間曲線を創成し，平面あるいは，線分との交点を計算するプロセスの繰返しである。そのため，13MIPSの計算機で，６００個の任意分布点群から１３×１３個の格子点を推定するための計算時間は４～７秒である。

(2) 本手法のプログラムは，FORTRANで作成されており，大きさは３４kWである。計算手法の簡便さも考慮に入れると，いわゆるパーソナルコンピュータでも本手法を使用することが可能である。

(3) 点群を補間する曲線式として，種々の理論式を用いることが可能である。したがって，補間に使用する式を変えることによって，許容される誤差を満足し，かつできるだけ短い計算時間で，任意に分布した点群から近似的に等価な格子状の点群を求めることができる。

前述したように，点群データから曲面を創成する手法の必要性は増加する傾向にあるが，曲面の特性や，曲面創成アルゴリズムの精度問題などの研究課題が残されている。

参 考 文 献

1) R.E.Barnhill, G.Birkhoff & W.J.Gordon: Smooth Interpolation in Triangles, J.Approx. Theory, 8 (1973) 114.

2) R.E.Barnhill: Smooth Interpolation over Triangles, Computer Aided Geometric Design, Academic Press, New York (1974)45.

3) 塩谷景一, 牧之内三郎, 山縣敬一 : 任意分布点群による自由曲面の一創成法, 精密機械, 49.7 (1983) 860.

4) R.F.Riesenfeld: Applications of B-spline Approximation to Geometric Problems of Computer Aided Design, Univ. of Utah UTEC-CSc-73-126(1973).

5) W.J.Gordon & R.F.Riesenfeld: B-spline Curves and Surfaces, Computer Aided Geometric Design, Academic Press, New York (1974) 95.

6) P.Bézier : Numerical Control-Mathematics and Applications, John Wiley & Sons, Inc. London (1972).

第5章　立体モデリング手法

5.1　緒　　言

製品の設計・生産では，形状に関する情報や，材質・熱的特性などの技術情報など種々のデータが必要になる。近年，製品性能が高機能化し，かつ軽薄短少が要求され，上述した情報の処理を計算機の支援下で行う必要性が高くなっている。本章では，これら種々なデータの中で，製品の形状に関する情報を計算機で取扱う手法について述べる。

形状としては，円柱・直方体・球など単純な部分形状の組み合せで表現できるものと，2次元の断面形状の移動によって定義できるものを扱う。前章までに述べた自由曲面では，曲面創成手法が研究課題の中心であったが，ここでは，機能的な要求によってきまる立体を対象とする。この場合，計算機上に作成された形状データから，有限要素法など設計で用いる技術計算プログラムへの入力データが作成できるデータ構造を考える必要がある。つまり，単に，グラフイックディスプレイやプロッタなどに作画できるだけでは不充分で，体積・立体を構成する面の数・面の法線方向などの立体に関する種々なデータが抽出でき，かつ，立体の内と外を識別できなければならない。ただし，立体に関する全ての情報が容易に抽出できるデータ構造の開発は難しい。そのため目的に応じて，ほぼ直接に抽出できるデータとして何を選択するかを決め，そのデータの取扱いが容易なデータ構造が開発されることが多い。

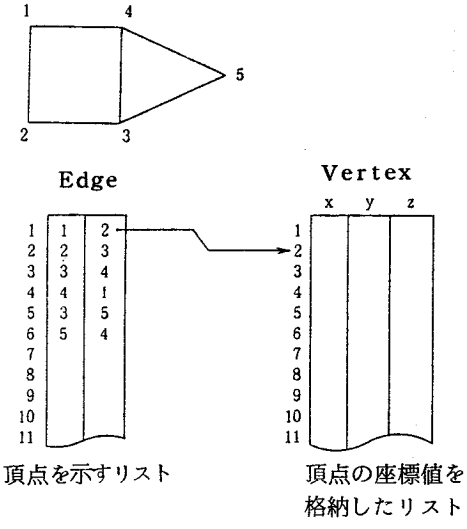
本章では，まず立体モデリング手法の基礎技術について述べ，工業的に頻出する形状の分析に基づいて新たに提示する立体モデリング手法について検討を行う。本手法は，一般に工業製品の形状が，適当に平面によって切断し，それぞれの部分について適当な局所座標系を用意すると，xy 平面上の一価関数によって部分形状の上界が定義される場合が多いことに着目している。つまり，この一価関数で表される上界の面と定義平面ではさまれた領域として表現できる立体セグメントを基礎において形状モデリングを行うものである。

5.2　立体モデリングの基礎技術

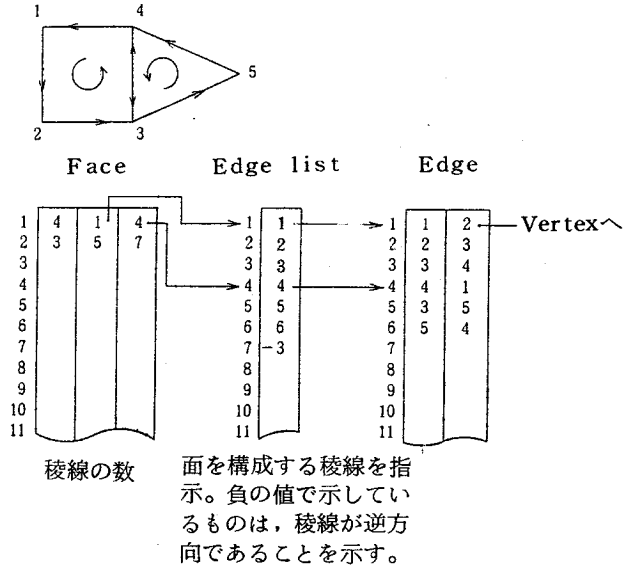
本節では，次節以降の比較対照のため，従来の立体モデリング手法を分析し基礎技術を明確にする。

5.2.1　形状定義[1)]

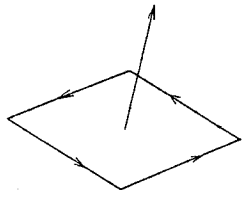
形状の表現を，頂点・稜線・面によって行う場合，形状データをどこまで正確に計算機内部に持つかによって，ワイヤーフレームモデル，サーフェスモデル，ソリッドモデルの三種類に分類される。



(a) ワイヤーフレームモデル



(b) サーフェスモデル



サーフェスモデルにおいて，面を定義する境界線が閉じていて，方向性を持つことを利用し，境界線を一周する方向が右ねじを進める方向とし，右ねじが進む方向をもって法線ベクトルの向きとする。実体は，法線方向にあるものとする。

(c) ソリッドモデル

図 5.1　形状の計算機内部での表現

(1) ワイヤーフレームモデル

形状を稜線のみで表現する方法である。面は定義されない。

データ構造は，稜線を定義するための端点の座標値の対を並べたマトリックスを用いる（図5.1(a)）。

(2) サーフェスモデル

ワイヤーフレームの間に面を張ることによって実現される。各面の構造化は行わず，各面は独立して定義する。一般にデータ構造は，ワイヤーフレームモデルのデータ構造に対して，面の境界を構成する稜線を一順するようにポインターを付与したものを用いる（図5.1(b)）。

(3) ソリッドモデル

サーフェスモデルでは，面は独立して定義したが，ソリッドモデルでは各面を構造化し全体として閉じた空間を定義する。サーフェスモデルとの相違は，面のいずれ側に実体が存在するかをデータとして付与するか，しないかである（図5.1(c)）。

ワイヤーフレームモデル，サーフェスモデル，ソリッドモデルを計算機内部に表現する方法は種々に考えられる。図5.1は方法の一例である。この立場で形状表現を行うと，自由曲面も1つの“サーフェス”であるから，実体が存在する方向を示すデータを付与することによって，ソリッドモデルとしての取扱いが可能である。

上記の説明は，頂点・稜線・面によって形状表現を行うという立場から，各モデル間の関係を述べたが，ソリッドモデルをワイヤーフレームモデルの延長線上にあるものと考えずに，ボリュームをもった基本形状の組合せによって表現する方法がある。

(4) モデルのレベル

図5.1に示したデータ構造をみると，ワイヤーフレームモデルの場合，長方形が，頂点1→2→3→4→1で構成されているという情報は保有していないことが分かる。ワイヤーフレームモデルの場合でも，人は表示図形をみて図形が長方形と三角形で構成されていることを認識できる。しかし，ワイヤーフレームモデルの場合，長方形，三角形という面の情報は含まれていない。つまり，計算機内部で面を取扱えない。

注意すべきことは，計算機の出力図をみて，ワイヤーフレームモデルであるとか，サーフェスモデル，ソリッドモデルであるとかいうのは間違いで，モデルを論ずる場合あくまで内部表現の問題であり，出力図には無関係である。

3次元CADシステムに用いるモデルのレベルが種々に論ぜられるが，システムの利用目的に応じて，モデルを選択する必要がある。ただし，ワイヤーフレームモデルの場合，断面図の作成，隠れ線消去などの図形処理を行うことが困難である。サーフェスモデルでは，図形処理はほとんど可能となる。しかし，マスプロパティ（mass properties）計算は困難で，NCテープの作

成も容易ではない。

しかし，モデルのレベルと実際の設計で要求されるデータとは一致しないことが多く，問題が指摘されている[2]。

5.2.2 ソリッドの理論[3]

H.B. Voelkerはソリッドモデルの基礎技術を，ＣＳＧ（Constructive Solid Geometry）とＢ－Reps（Boundary Representations）の二種類に分類している。ＣＳＧはソリッドモデルのセットオペレーションによる定義法を基礎にして，計算機のメモリ上にソリッドモデルのデータを構築する。Ｂ－Repsとは，面間の結合関係を与えるトポロジー構造によって，ソリッドモデルをメモリ上にデータとして表現したものである。

図5.2に，ＣＳＧとＢ－Repsの概念図を示す。

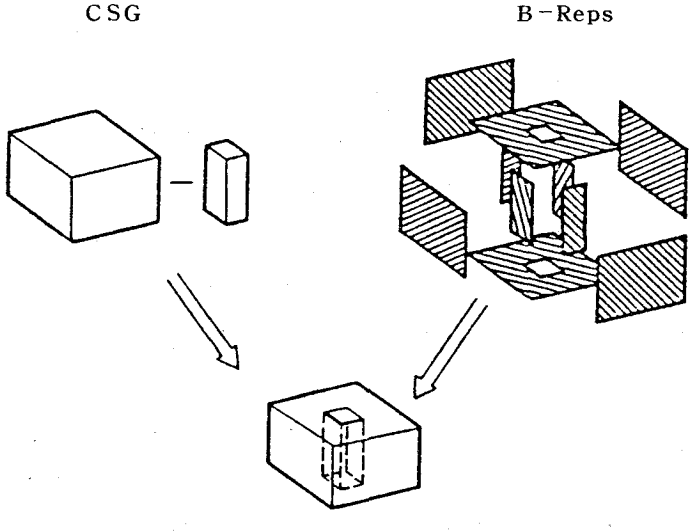


図5.2　ＣＳＧとＢ－Repsの概念図

(1) ＣＳＧについて

ＣＳＧでは，形状を3次元空間における集合として数学的に取り扱えることに基づいて，集合演算をベースにして形状定義を行う。すなわち，形状を集合Sと考え，Sをいくつかの部分集合S_j（$j=1.2.\cdots$，m）に分割すると，

$$S=S_1 \cup S_2 \cup \cdots\cdots \cup S_m=\bigcup_{j=1}^{m} S_j \qquad (5.1)$$

さらに

$$S_j = S_{1j} \cap S_{2j} \cap \cdots\cdots \cap S_{nj} = \bigcap_{i=1}^{n} S_{ij} \qquad (5.2)$$

なる S_{ij} が存在する。よって

$$S = \bigcup_{j=1}^{m} \left\{ \bigcap_{i=1}^{n} S_{ij} \right\} \qquad (5.3)$$

Sが与えられた任意の形状であるとき，S_{ij} の決め方には，広範な自由度がある。このS_{ij}を初等幾何学で表現できる基本形状とするとき，この形状をプリミティブとよぶ。式（5.1）～（5.3）に従って部品形状を定義する場合，プリミティブの結合には，セットオペレータの和，積が用いられることになる。

データ構造はバイナリトリー（binary tree）を用いる。このトリーはＣＳＧトリーともいわれ，プリミティブリーフ（primitive leaf），セットオペレータノード（set-operator node），モーションノード（motion node）によって構成される。ＣＳＧトリーの例を図5.3に示す[4]。

ここで，Ｐ1，Ｐ2はプリミティブでありＵはセットオペレータ，ΔＸはプリミティブのモーションを表している。

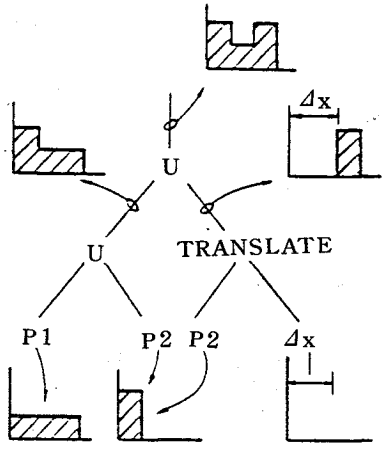


図5.3　　ＣＳＧトリーの例

(2) B－Repsについて

基本的に，面・稜線・頂点のトポロジーをベースにしたデータ構造である。前述した図5.1は，B－Repsの典型的な例である。この場合，面のデータが核となり，稜線・頂点が表現される。一方，Baumgartは，図5.4に示すように，稜線に注目しその左右の面を含む図を描き，ウィングドエッジ（Winged Edge）と名づけた[5]。ウィングドエッジをデータ構造の核とすると，すべての稜線において，図5.4の型は同じである。そのため，データの取扱いが容易になり，ウィングドエッジを収容するためのメモリサイズは常に一定となる。

なお，面・稜線・頂点で表現するデータ構造の場合，実際に存在しない形状を作ってしまう恐れがあるので，形状の定義をチェックするために拡張オイラーの式（5.4）を用いる。式（5.4）を満たすことが形状が完全であるための必要条件であるからである。

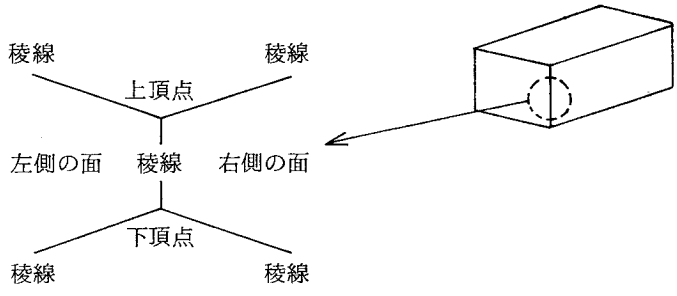


図5.4　ウィングドエッジによる表現

$$V - e + f - L + 2p - 2b = 0 \qquad (5.4)$$

V：頂点の数　　e：稜線の数

f：面の数　　L：ループの数

p：パスの数　　b：ボディーの数

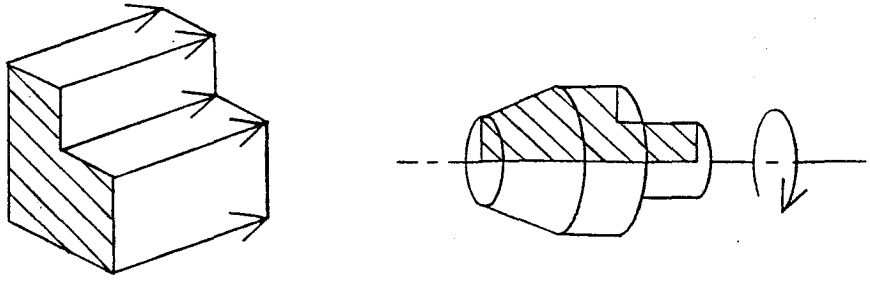
ここで，ループは面の中に面がある場合で，パスは貫通穴である。

(3) システムへの形状入力について

ＣＳＧ構造はプリミティブと和と積などのセットオペレータで表現されるので，形状入力としては，プリミティブの種類とその空間位置，セットオペレータを与える。

B－Reps構造の場合も，形状の入力手段として，プリミティブのセットオペレーションが用いられる。ただし，この場合のプリミティブはそれ自身，面・稜線・頂点からなるトポロジーを持つが，計算機内部ではセットオペレータのデータ，プリミティブの種類とその空間位置に関す

る情報は失われ，単にオブジェクト（object）の大規模なトポロジーが表現されているのみである。すなわち，B－Reps は境界を伴う面によってデータが作成されているが，これを直接入力することは困難である。そのため，ＣＳＧ的入力が用いられる。B－Reps を直接入力する方法として，２次元プリミティブのスイープ（sweep），回転による方法がある。図5.5にその例を示す。



スイープ　　回　転

図5.5　スイープと回転による形状定義

5.3　立体セグメントの作成[6)]

5.3.1　基本的な考え方

一般に工業製品の形状は，適当に平面によって切断し，それぞれの部分について適当な局所座標系を用意すると，$x\ y$ 平面上の一価関数によって部分形状の上界が定義される場合が多い。これは生産の過程に関係しており，加工を容易にするため，あるいは組立て作業を一方向から統一的に行う目的に沿わせるためであり，実際に計算機内部に作られたモデルを適当に分割して製造工程に適応させることはしばしば行われる。

本研究で述べるシステムは，部分形状ごとに局所座標系を用意した時，それらの形状が一価関数で表される上界の面と定義平面ではさまれた領域として表現できる立体セグメントを基礎において，形状モデリングを行うものである。それぞれの部分形状を上の条件を満たす集合演算で作成し，最後に物体全体を表現するために，基礎平面あるいは基礎多面体上にこれらの部分形状を配置，または接合して全体のモデル（オブジェクト）を構成する。

このシステムは開発を行い実際的応用に使われつつある。理論的に一般のソリッドモデルを扱うシステムに比べるとサブセットを扱っているに過ぎないが，上の条件を付加することによって，形

状処理の計算量を減らすことができる。しかも，工業製品の大部分を扱うことができるので，比較的小規模の計算機でも十分な実用性を得ることができる。この点が，本章で述べるシステムの大きな特徴になっている。

本章においては，一般のソリッドモデリングの場合と異なって，一定の制約条件のついた部分形状を扱うので，まず立体セグメントの定義を与え，続いて立体セグメント同志の集合演算を定義する。

5.3.2 立体セグメントの表現[7)]

図5.6の局所座標系において（これはそれぞれの立体セグメントごとに用意する），x y平面上に定義域Eを定める。これをz軸方向に掃引して得られるx y平面に垂直な柱状立体の表面をG，その内部の半無限柱領域をΣとする。これに，定義域Eにおいて一価関数として表される曲面Fを配置し，領域ΣにおいてFとx y平面にはさまれた領域を立体セグメントSとする。一価関数であるという条件を別にすればEやFの形状に制限はなく，また柱状立体の表面Gは必ずしもSの境界に使われるとは限らない。半球や円錐はここでいう立体セグメントに属する。

また曲面Fを定義する関数は，部分的に定義して接続して構わない。次節で述べるように立体セグメント同志の集合演算によって複雑な形状が定義できる。例を図5.7に示す。

このような立体セグメントの形状処理が，一般の3次元形状処理に比べて簡単になることは容易に推察できる。例えば，与えられた任意の一点がSの内部にあることの判定は，局所座標系において，

(1) 与えられた点のx y座標値が領域Eにあること

(2) z座標値がx y平面とFの間にあること

の2段階で行うことができる。扱っているのは3次元形状であるが，形状処理は2次元処理と1次元処理に分解されてしまうのである。

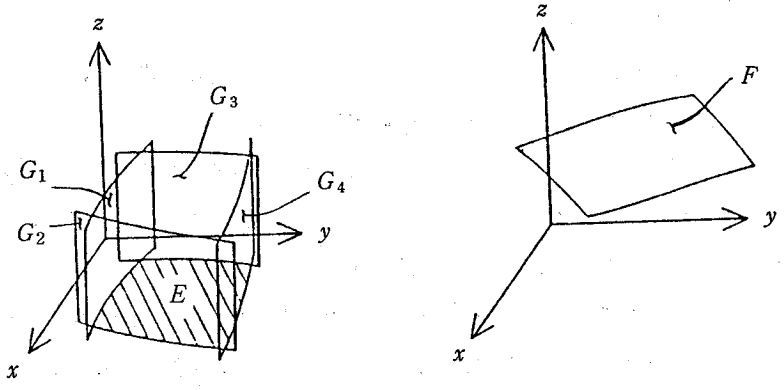


図5.6 立体セグメントの定義

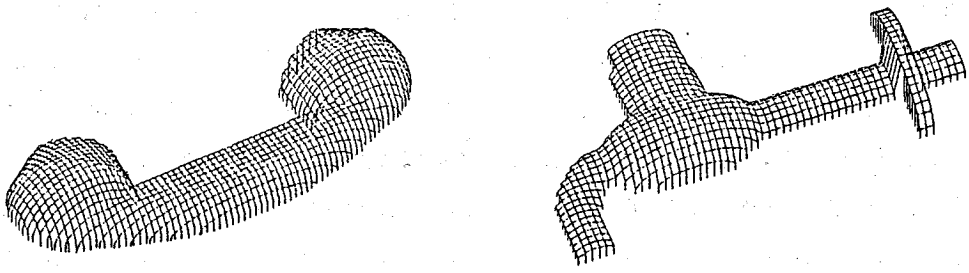


図5.7　立体セグメントの例

なお，図5.6には立体セグメントがz軸の正の側にある場合を示したが，負の側に定義してもよい。後に述べるように物体全体を構成する時，後者はくぼみの形状を定義する。両者を区別するときには，S^+，S^-という記号を使う。一つの立体セグメントで正の側，負の側の両方にまたがるものを定義することは数学的には可能であるが，形状処理の簡便さのためには適当ではないので用いていない。物体全体を構成するときに組み合わせる。これについては後述する。

5.3.3　立体セグメントの集合演算[7)]

立体セグメントは，プリミティブから出発してこれに集合演算を施すことによって作成する。プリミティブとしては，円柱，直方体，円錐，半球などがあるが，本システムにおいては工業用製品にしばしば現われる部分形状はシステムに登録されており，プリミティブと同様に扱うことができる。プリミティブ自身ここで言う立体セグメントとしての性質を持っている。

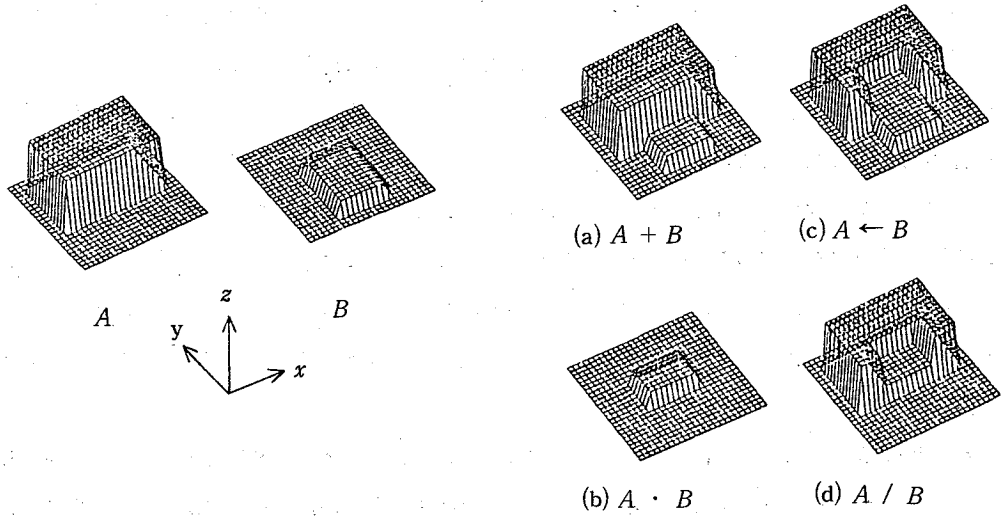


図5.8　集　合　演　算

ここで扱っている立体セグメントは，前節で述べたように制約のついた3次元形状であるため，集合演算は通常のソリッドモデルを扱う場合とは若干性質の異なったものとなる。図5.8において，A，Bを立体セグメントするとき，集合演算を行うに当って，まずA，Bの定義平面（$x-y$平面）を同一平面上に配置することを前提とする。この前提を置くことによって，図5.8(a)と(b)に示すように，和と積については通常の集合演算が行える。定義域が拡大されたり，縮少されたりすることは認める。この際，“一価関数で定義される形状”という性質は保存される。一方，差の集合演算は定義されない。明らかに，AからBを差し引いた場合，上の性質は必ずしも保存されないからである。

和と積の他に，図5.8(c)，(d)に示す二つの集合演算を新しく導入する。これらは以下のように定義される。

$$A \leftarrow B : (A \cap \overline{\Sigma}_B) \cup B \qquad (5.5)$$

$$A / B : (A \cap \overline{\Sigma}_B) \cup (A \cap B) \qquad (5.6)$$

ここにΣ_BはBの定義域をz軸方向に掃引した柱状領域を表し，$\overline{\Sigma}_B$はその補集合である。

これらの演算は，A，Bの定義域E_A，E_Bの共通部分において，Bの曲面F_Bを用いて形状を定めることを意味している。第1オペランドと第2オペランドの順序は重要である。これらの演算は集合演算と言うよりは，立体セグメントの性質に注目してオペランドの曲面，F_AとF_Bについての操作をするものであり，実際に利用してみて実用価値の高いことが確かめられている。

5.4 物体全体の構築法[6)7)]

立体セグメントを組み合わせて，物体を構成するのには、核となるような凸多面体を用意する。以下ではこれを基礎多面体と呼びΓであらわす。ただし，特別な場合として，多面体が縮退した場合の基礎平面も扱えるようにする。この場合は，基礎平面の輪郭は任意であって，必ずしも凸領域を定義するものでなくてよい。

立体セグメントは局所座標系のx y平面上に定義されているので，適当に座標交換を行ってΓ上に配置し，各面に接合することによって物体を構成することができる。ただし，Γ上の一平面において立体セグメントが定義されていない部分（定義域Eの外部）については，Γの平面がそのまま物体の境界となる。換言すると立体セグメントの定義域外の領域では，局所的なz座標値は零であるとみなす。

物体構成の過程では，一般的な集合演算は実施しないが，第5.3節でS^+とS^-の区別を導入してある。この例を図5.9に示してあるが，S^+の場合は単にΓへの接合を表すが，S^-の場合は，集合演算の差を表すことになってΓの一部がくり抜かれることになる。

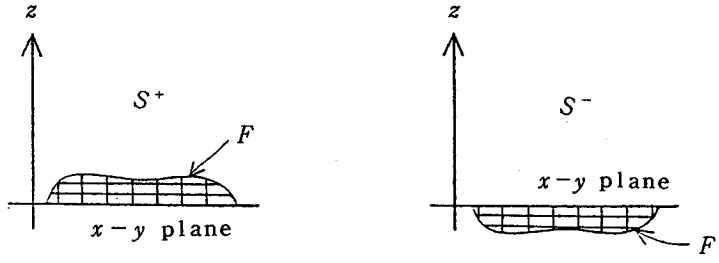


図5.9　立体セグメントの種類

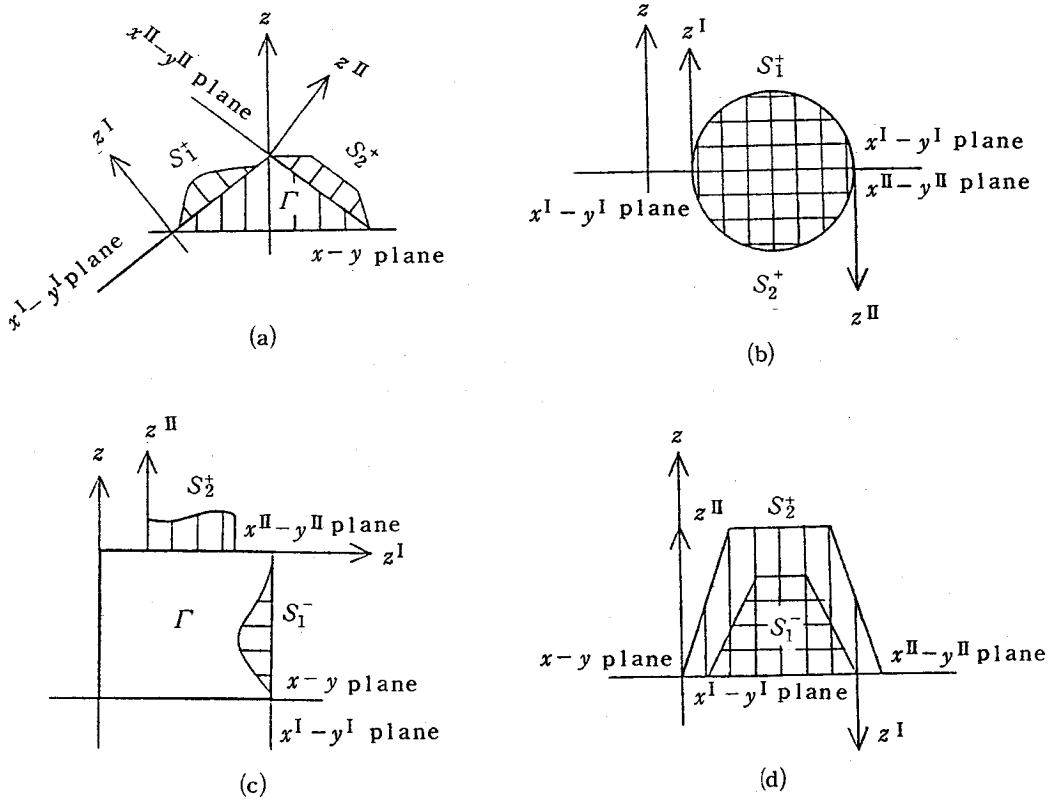


図5.10　物　体　構　成

図5.10に物体構成の典型的な例を示す。以下では，構成される物体全体のことをオブジェクトと呼ぶ。図5.10(a)は，Γはオブジェクトの座標系$x-y-z$において定義され，Γ上に局所座標系を配置して立体セグメントを接合する様子を表している。図5.10(b)は球の定義であって二つの半球を接合して表現する。この場合Γは2次元の円に縮退した基礎平面を用いている。図5.10(c)は，S^+とS^-の混在する例である。とくに図5.10(d)では，図は2次元的に示してあるが，S^+とS^-が混在していてなおかつΓが平面に縮退している場合で，S_2^+の内部がS_1^-によってくり抜かれている例である。

一般にオブジェクトをVとし，立体セグメントの集まりを$\{S_1^+, S_2^+ \cdots\}$並びに$\{S_1^-, S_2^-, \cdots\}$とすると，オブジェクトを構成するには，適当な座標変換の後，次の集合演算を行うことになる。

$$V=\Gamma\cup(\bigcup_i S_i^+)-\bigcup_j S_j^- \qquad (5.7)$$

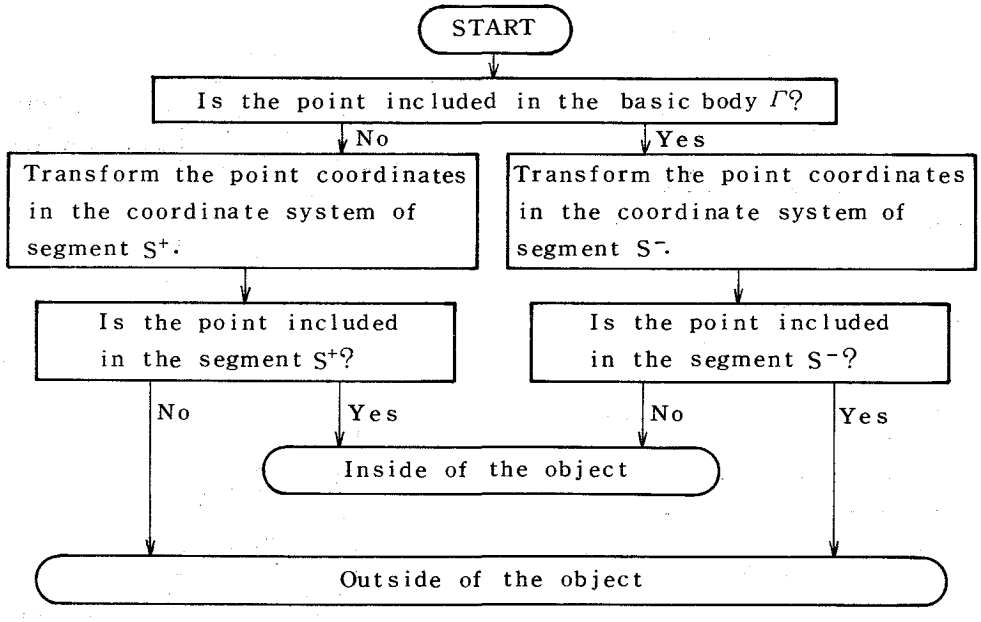


図5.11　任意に与えられた一点がオブジェクトの内部にあるか否かを判定する処理

任意に与えられた一点がオブジェクトの内部にあるか否かを判定する手順を図5.11に示す。基本的には，以下の3段階の計算が必要になる。

(1) 与えられた点がΓの内部にあるか否かを判定する。

(2) 各立体セグメントの局所座標系へ与えられた点の座標を変換する。

(3) 立体セグメントS^+またはS^-の内部に含まれるか否かを判定する。

基礎立体Γの内部に含まれるか否かの判定は，3次元の処理であるが，凸多面体であるので，一般の3次元形状処理に比べるとそれ程計算量は多くない。それぞれの立体セグメントについての判定は第5.3節で述べたように簡単であるから，ここで取り上げたオブジェクトの構成法は，各種形状処理における計算量の軽減につながることが期待できる。

5.5 適 用 例

第5.3節および第5.4節で述べた手法を応用したシステムについて述べる。図5.12は、本システムを用いて工業上よく現われる形状を作成した例である。図5.12の(a), (b), (c)は，2つの局所座標系を持ち，Γは基礎平面を用いている。特に，図5.12の(a), (b)は，基礎平面に対して対称な2つの立体セグメントで構成した例である。

図5.12の(d), (e)は部品とその金型作成例である。一般に金型を用いて作成する部品は，その製造法から一つの局所座標系におけるS^+として定義でき，図5.12の(a), (b), (c)のように局所座標系を二つ設定すれば，ほとんどの場合作成できる。金型の作成は，図5.12の(e)の場合，直方体の

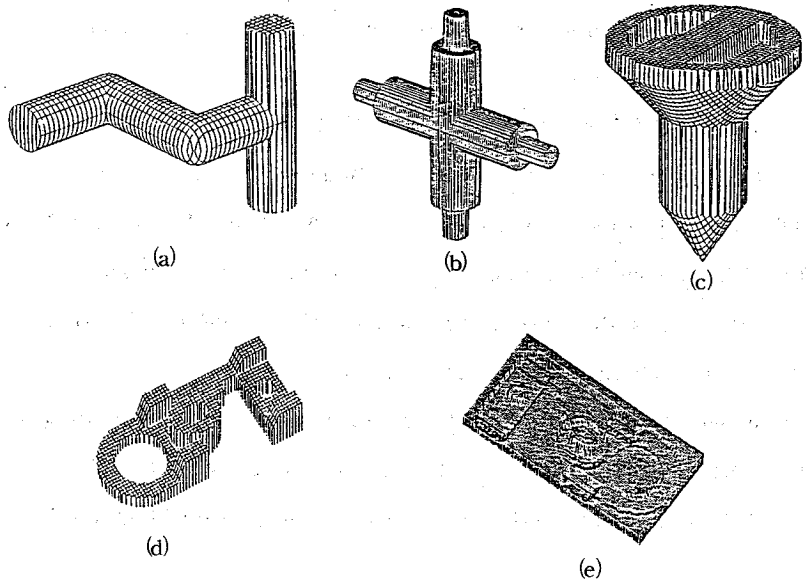


図5.12　工業上よく現れる形状の作成例

基礎多面体Γを設定し，部品であるS^+ をS^- へ変換して接合することにより実現している。この変換はシステムが行う。本システムはこのように，工業製品で現れる大部分の形状を扱える。また，金型の自動作成などへの応用も可能である。

5.6 結　言

本章では，機能的な要求によって決まる立体をモデリングする手法を示した。理論としては，従来のB-Reps，ＣＳＧなどのソリッドモデリング手法と比べるとサブセットを扱っているに過ぎないが，応用を考えた場合，いくつかの特徴がある。本章で得られた結果の要約を以下に示す。

(1) 生産の過程においてしばしば処理の対象となる部分形状を立体セグメントとして積極的に取扱える手法を開発した。

(2) 任意の一点が形状の内部にあるか外部にあるかの判定を，２次元処理と１次元処理で行う手法を開発して各種形状処理における計算量を軽減することができた。

(3) 形状モデリングに際して用いるオペレーションは，形状の表面として用いたい面を基礎としており，加工面など生産における形状操作の概念に近い。

本章では応用例として，金型形状を取りあげたが，金型はプラスティクメルト解析など，割れない金型を作成するために種々の技術計算が必要である。これらの技術計算プログラムを含めた，総合的な金型用ＣＡＥ／ＣＡＤ／ＣＡＭシステムの開発が今後進むものと考えられる。

参　考　文　献

1） A.A.G.Requicha and H.B.Voelcker:Solid Modeling:A Historical Summary and Contemporary Assessment,IEEE Computer Graphics and Applications,2.2(1982).

2） 木村文彦：ＦＡにおける情報処理技術の役割，情報処理，25.4(1984)283.

3） H.B.Voelcker and A. A. G. Requicha:Boundary Evaluation Procedures for Objects Defined via Constructive Solid Geometry, Tech. Memo. 26. Production Automation project, Univ. Rochester (1980).

4） A.A.G. Requicha:Representations for Rigid Solids,Theory,Methods and Systems, Computing Surveys, 12.4(1980).

5） B.G.Baumgart:Geometric Modeling for Computer Vision, Rep. STN-CS-74-463. Stanford Artificial Intelligence Lab.(1974).

6） 塩谷景一：CAD/CAMにおける形状モデリングの一手法，精密工学会誌，52.11(1986)1922.

7） K. Shiotani and K.Hironaka:Restricted-Three-Dimensional-Solid Modeling, Proc. of 5th ICPE(1984)744.

第6章　金型用3次元CAD/CAMシステム

6.1　緒　　言

本章では，モールド金型など3次元の金型製作を支援するシステムの開発について述べる。

近年，製品開発期間の短縮等の理由から，金型製作へのCAD/CAMシステムの導入が急務となっている。このため，種々なシステムが開発され，実用化への展開が試みられている。しかし，一般にこれらのシステムでは，取扱える3次元形状が簡単な関数で表せる曲面に限られたり，逆に自由曲面が扱える場合には，図面に表れていない点の座標を計算して入力しなければならない場合がある。また，任意の工具経路を作成することが難しく，加工条件の指定も詳細にできないことが多いとも言える。一方，NC自動プログラミングシステムでは，形状の定義をせず工具の動きを逐一定義するため，複雑な曲面になるとプログラムがきわめて複雑になる傾向がある。

そこで，これらの問題を解決するために，ブロック構造の特徴のあるプログラム言語を開発し，この言語を用いて金型形状の記述から加工手順，加工条件の指定を行うシステムを実現した。本手法では，曲面を構成する曲線間の相互関係により，曲面補間法を15種類に分類し，曲面要素と名付けている。まず，図面に表された断面曲線を定義した後，曲線の本数と曲線間の相互関係から，適切な曲面要素の種類を選択する。次に，曲面要素を接続したり，重ね合わせることにより，複雑な曲面を容易に定義できる。また，曲面のNC加工の場合データ量が多くNCテープ一巻では対応できないので，NCテープのかわりにフロッピーディスクを用いることが可能なシステムとなっている。

6.2　システムの概要

システムのハードウェア構成と基本的な処理を以下に示す。

まず，処理の流れを図6.1に従って説明する。

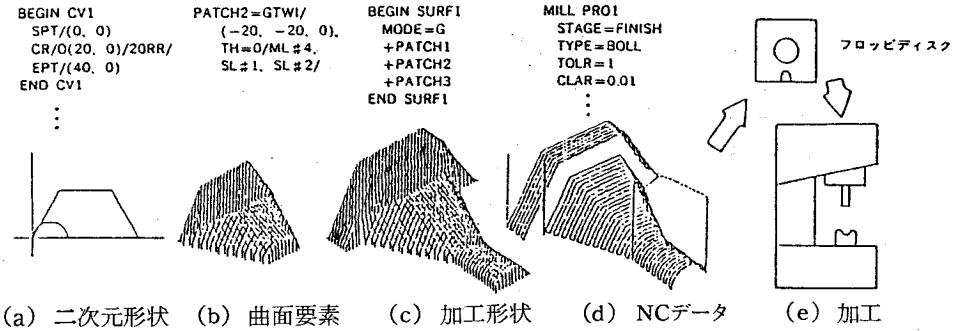


図6.1　処理の流れ

(1) まず，金型図面に表された断面曲線や輪郭曲線を，円弧や直線のパラメータを入力することによって定義する。

(2) 次に，これらの曲線間をなめらかにつなぐ曲面要素の種類を選択する。曲面要素は１５種類備えており，データとして与える複数の断面曲線の相互関係によって，適切なものを選定する。

(3) さらに，これらの曲面要素を合成することにより，金型形状を定義する。

(4) 形状定義を行ったのち，加工条件，つまり工具の送り量，ピックフィード量，工具径および加工パターンを与えると，自動的に工具経路が計算される。

(5) こうして求められたＮＣ指令のデータは，フロッピーディスク上に出力される。このフロッピーディスクをＮＣ装置側に接続したパーソナルコンピュータにセットすれば，紙テープイメージでＮＣ装置へデータが転送され加工が行われる。

図6.2にシステム構成を示す。上述したように，ＮＣ指令のデータは，フロッピーディスク（またはハードディスク）を用いてＮＣ装置へ供給される。データの編集，変更はＯＡ（Ｏｆｆｉｃｅ Ａｕｔｏｍａｔｉｏｎ）用に開発されたユーティリティプログラムを用いて簡単に行える。

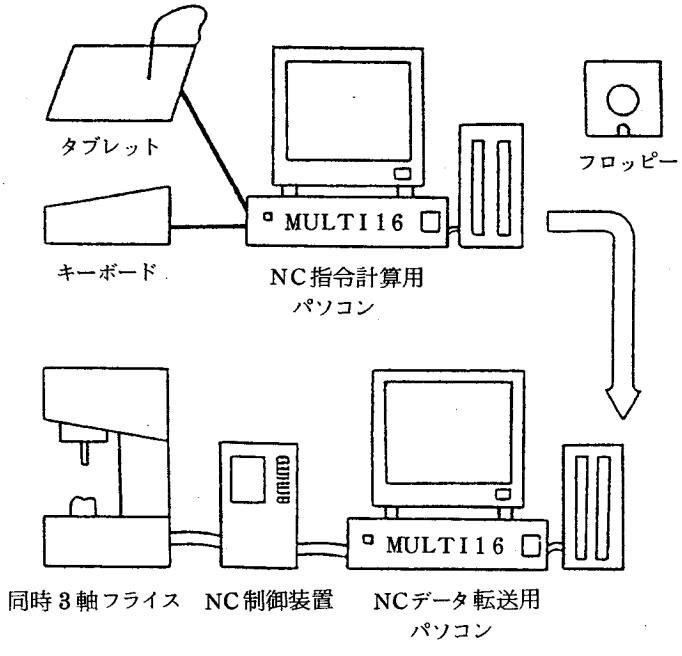


図6.2　ハードウェア構成

6.3 システムの機能

6.3.1 プログラム言語概要

本システムにおける金型形状の記述，加工手順，加工条件の指定などは，専用言語によるプログラミングによって行う。従来の加工システムでは，ＡＰＴ系の言語が用いられた。そこで，まず，ＡＰＴ系の言語について簡単に説明する[1]。

ＡＰＴ系言語では予め図形要素を記号名をつけて定義しておき，それらで規制される面を基準として工具動作を記述していく。

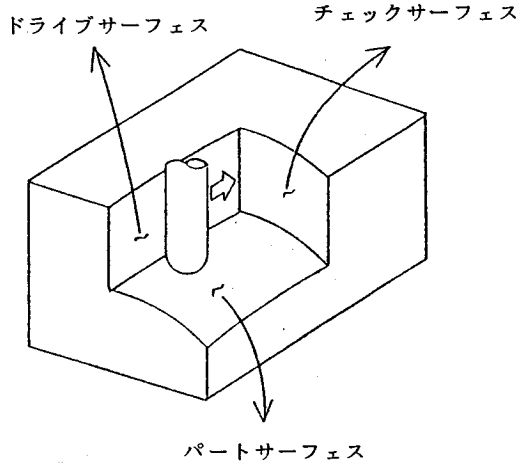


図6.3　ＡＰＴ系の言語における工具動作の制御面

図6.3に示すように，一つの工具動作の記述にパート・サーフェス，ドライブ・サーフェス，チェック・サーフェスの三つの面が必要になる。これらの制御面はパート・サーフェスは通常平面であるが，ドライブ・サーフェス，チェック・サーフェスは平面，円筒面，一般２次曲面などが許される。通常の工具動作では，工具はパート・サーフェスとドライブ・サーフェスに沿ってチェック・サーフェスまで動き，次は今までのチェック・サーフェスがドライブ・サーフェスとなって新らしいチェック・サーフェスまで動く。ＡＰＴプログラムでは予め定義された制御面の記号を用いた工具運動文により逐一工具動作を記述して行くことが必要である。

ＡＰＴ系の言語は本質的に加工動作の定義と形状定義を同時に行うため，加工工程，加工条件の変更に際して，プログラムの大幅な修正が必要である。また，プログラムにおける変数の有効範囲が不明確であるという問題もある。

そこで，ALGOLで用いられたブロック構造[2]の特徴を持つ，金型３次元CAD／CAM システム

用の専用言語を開発した。

開発した言語の特徴を示す。

(1) 局所的変数と非局所的変数の定義が明確である。

(2) プログラム数行で一つの機能を表現する場合，これらの行を一つのブロックとすることで，後での参照が容易である。

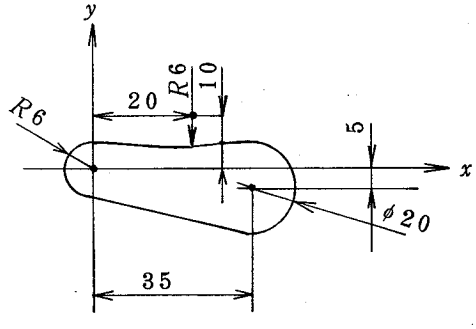
(3) 加工単位を一つのブロックとして定義でき，加工条件，加工順序の指定ができる。

(4) 形状定義から加工までプログラムできるため，プログラムリストが技術資料として役だつ。

6.3.2 2次元形状定義機能[3)]

第3章でも述べたが，図面上での曲面の定義は，要所要所の断面曲線を用いて行う。この断面曲線は，円弧と線分の組み合わせで構成されている。以下に，この断面曲線など，2次元形状を定義する機能を説明する。

図6.4(a)はAPT系の言語の一つであるMEDIAPT[4)]を用いて形状定義した例である。1行めから7行めのプログラムは補助図形の定義で，9行めから12行めにおいて引用される。このように，APT系の言語で2次元形状を定義するには，あらかじめ補助図形を定義しておく必要がある。その他，前述したように，変数の有効範囲が不明確である。



APT系の言語

```
 1  P1=X-6/Y0
 2  C1=-X0/Y0/R6
 3  C2=X20/Y10/R6
 4  C3=X35/Y-5/R10
 5  L1=C1, U/C2, D
 6  L2=C2, D/C3, U
 7  L3=C3, D/C1, D
 8  #
 9  ROUT/E1, P1
10  C1, CW; L1 ; C2, CC; L2
11  C3, CW; L3 ; P1
12  REND
```

(a)

提案する言語

```
 1  BIGIN CV02
 2  SPT/(-6, 0)
 3  *
 4  CR/O(0, 0)/6RR/
 5  LN
 6  CR/O(20, 10)/6LR/C
 7  LN
 8  CR/O(35, -5)/10RR/C
 9  LN
10  M#4/C
11  *
12  EPT/(-6, 0)
13  END CV02
```

(b)

図6.4 形状定義機能の比較

図6.4(b)は本章で提案する言語で形状を定義した例である。各行についての簡単な説明を次に示す。

1行　曲線番号を2番と定め，定義を始める。

2行　曲線の始点を座標値（−6 ,0 ）の点とする。

3行　注釈行

4行　中心座標が（ 0 , 0 ），半径6の時計まわりの円弧を定義する（曲線を構成する線分は，有向線分とし，円弧も時計まわりか反時計まわりのデータを持たせる）。

5行　この線分は，4行で定義した円と連続につながっており，次の図形が定義されないと，ただ一つに定まらない。この場合，図形が線分で一つ前の図形と連続であることのみを指定しておく。

6行　中心（ 20, 10 ）で半径6，反時計まわりで，一つ前の図形と連続につながる（／cで指定）円を定義。

7行　5行目と同じ定義方法

8行　6行目と同じ定義方法

9行　5行目と同じ定義方法

10行　以前に定義した図形と同じ図形の場合，M＃行番号で定義できる。一つ前の図形と連続であることを指定している。

11行　注釈行

12行　曲線の終点

13行　曲線No.2の定義を終了する。

このように，曲線は，BEGINとEND で囲まれたブロック内で，順次曲線に沿って円弧，線分を指定することで定義できる。

次に，開発した言語の文法を示す。

(1)　プログラム単位

文は複数の行から構成される。文は80カラム内で記述される。文には，他の文がその文を引用することができるように，文番号を，1～5カラムにつける。プログラム単位内で，同じ文番号を2つ以上の文につけてはならない。一つのプログラム単位はSTARTで始まり，STOPで終わる。（図6.5 ）。

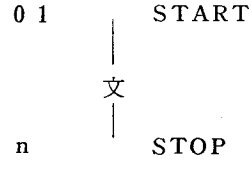


図6.5　プログラム単位

(2) 2次元形状定義言語の文法

文法の説明で用いる記号の意味を，表6.1に，文法を表6.2に示す。

ＡＰＴ系の言語（主にMEDIAPT）と比較し，次の特徴がある。

(i) 曲線に沿って順次形状定義できる。つまり，円弧は全円，線分は直線としてあらかじめ定義しておく必要がない。

(ii) 曲線を順次定義していく過程で，円や直線のパラメータが全て分からなくても，不定のまま定義できる。すなわち曲線全体が定義されて決まるパラメータは，＊を用いることで未知数として指定できる。

(iii) 曲線全体を一つの形状要素として扱うことができ，曲線どうしの接続ができる。

(iv) 補助図形を前もって定義する必要がない。プログラム中において，補助図形は自由に定義できる。補助図形が必要な行の次の行で定義するというような，後参照も可能である。

(v) プログラムを構造化できるので，理解しやすいプログラムを作成できる。

特に，上述した(iii)の特徴は，2次元形状定義言語で作成した曲線から，曲面を定義する場合に重要となる。

表6.1　文法の説明で用いる記号

#ℓ	ℓ行の図形要素を用いる。
(,)	座標値
n	図形番号
□	数値を入れる。
〔〕	一つパラメータを選択する。
{}	省略するか，一つパラメータを選択する。
＊	形状要素のパラメータで，図面には数値が指定されていないが，前後あるいは，2つ前，後ろの図形が定義されると一義的に決まるパラメータを示す。

表6.2　文　　法（1／4）

モード	ステートメント	内　　容
モデル作成モード	BEGIN CVn END CVn	ブロック内で1つの曲線を定義する。
連続曲線モードI	SPT/ ⎧P#ℓ⎫ 　　⎩(,)⎭	連続曲線の始点を定義する。
	EPT/ ⎧P#ℓ⎫ 　　⎩(,)⎭	連続曲線の終点を定義する。
	CR/ ⎧O#ℓ⎫ / ⎧□RR⎫ / ⎧C⎫ 　　⎪O(,)⎪　　⎪□LR⎪　　⎨S⎬ 　　⎩ ＊ ⎭　　⎩ ＊ ⎭　　⎩F⎭ 省略形 RR≡CR/＊/□RR/C LR≡CR/＊/□LR/C	円弧の定義 CR／円の中心／円の半径／ □　円の半径 RR　時計まわり。 LR　反時計まわり。 ＊　モーダル　前後の定義文によって決まるパラメータ。 C　指定すると前の図形と接線まで連続につながることを示す。 S, F　一つ前の図形と2つの交点をもつ場合に，図形要素の始点としてどちらを用いるかを示す。 （ポイント作成モード参照）
	LN/ ⎧P#ℓ⎫ / ⎧□PA⎫ / ⎧C⎫ 　　⎪(,)⎪　　⎪□NA⎪　　⎨S⎬ 　　⎩ ＊ ⎭　　⎩ ＊ ⎭　　⎩F⎭ 省略形 LN≡LN/＊/＊/C	線分の定義 LN／通過点／傾き／ □　線分の傾き（角度） PA　xの正の方向を向く線分 NA　xの負の方向を向く線分 傾き90度の場合，PA，NAはyの方向の正負を区別する。

表6.2　文　　法（２／４）

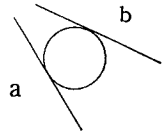
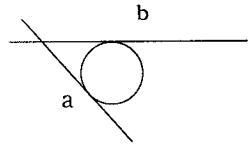
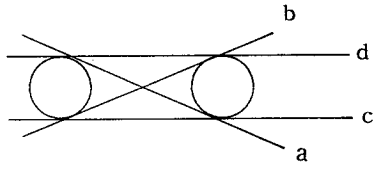
モード	ステートメント	内　　容
連続曲線モードⅡ	LN/C#ℓ/C	C＃ℓで指定した円に接する線分。 前の図形とは必ず連続につながる。  C＃ℓで示した円に，仮に接線まで連続につながるとして，a，bどちらをとるかが決まる。
	LN/C#ℓ_1, $\left\{\begin{matrix}P\#\ell_2\\(\ ,\ )\end{matrix}\right\}$ / $\left\{\begin{matrix}S\\F\end{matrix}\right\}$	一点を通りC＃ℓ_1に接する線分。  a，bの決定は，C＃ℓ_1の円の定義方向で決まる。
	LN/C#ℓ_1, C#ℓ_2/ $\left\{\begin{matrix}S\\F\end{matrix}\right\}$	2つの円に接する部分。  C＃ℓ_1，C＃ℓ_2と接続まで連続につながると仮定して，a，bあるいはc，dをとるかが決まる。
	CR/L#ℓ/ $\left\{\begin{matrix}\square RR\\\square LR\end{matrix}\right\}$ /C	L＃ℓに示した直線に接する円弧。 前の図形とは必ず連続。
	CR/ $\left\{\begin{matrix}P\#\ell\\(\ ,\ )\end{matrix}\right\}$ / $\left\{\begin{matrix}\square RR\\\square LR\end{matrix}\right\}$ /C	一点を通過する円弧の定義。 前の図形とは必ず連続。
	M/#ℓ $\left\{\begin{matrix}/C\\/S\\/F\end{matrix}\right\}$	＃ℓで定義した図形と同じ図形を定義する。 ただし，前の図との接続は新たに指定する。

表6.2　文　　法（3／4）

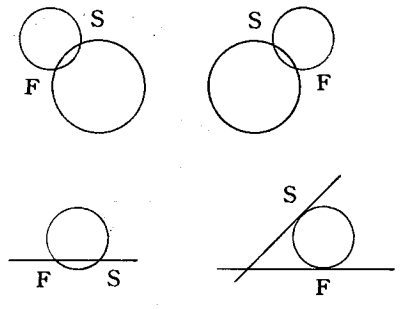
モード	ステートメント	内　　容
直接定義モード	LN/ $\begin{bmatrix}P\#\ell_1\\(\ ,\ )\end{bmatrix}$, $\begin{bmatrix}P\#\ell_2\\(\ ,\ )\end{bmatrix}$	与えられた2点を両端点とする線分。
	CR/ $\begin{bmatrix}P\#\ell_1\\(\ ,\ )\end{bmatrix}$, $\begin{bmatrix}P\#\ell_2\\(\ ,\ )\end{bmatrix}$ / $\begin{bmatrix}\square RR\\\square LR\end{bmatrix}$	与えられた2点を両端点とする円弧。
	RD/$\#\ell_1$, $\#\ell_2$ $\begin{bmatrix}\square RR\\\square LR\end{bmatrix}$	2つの図形要素の間に連続な円弧（丸みづけ）を挿入する。
補　助	*	コメント行である。
	#	BEGIN　END内で，他のステートメントから引用してのみ用いられる図形要素を定義する。連続曲線の一要素とはみなされない。
ポイント作成モード	PT/(,) PT/RA(,) PT/L$\#\ell_1$, L$\#\ell_2$ PT/C$\#\ell_1$, C$\#\ell_2$/ $\begin{bmatrix}S\\F\end{bmatrix}$ PT/C$\#\ell_1$, L$\#\ell_2$/ $\begin{bmatrix}S\\F\end{bmatrix}$ PT/C$\#\ell_1$, $\begin{bmatrix}P\#\ell_2\\(\ ,\ )\end{bmatrix}$ / $\begin{bmatrix}S\\F\end{bmatrix}$	 S，Fの識別は2つある交点のうち，y座標の大きい方をSとする。y座標が同じ場合，x座標が大きい方がS。
全円作成モード	CR/ $\begin{bmatrix}P\#\ell_1\\(\ ,\ )\end{bmatrix}$, $\begin{bmatrix}P\#\ell_2\\(\ ,\ )\end{bmatrix}$, $\begin{bmatrix}P\#\ell_3\\(\ ,\ )\end{bmatrix}$	

表6.2　文　　法（4／4）

モード	ステートメント	内　　容
図形変換モード	MOVE CVn □X,□Y	ＣＶｎを並進移動した曲線を定義する。
	ROTA CVn □A	ＣＶｎを座標原点を中心に回転移動させた曲線を定義する。
	SCAL CVn □B	拡大，縮小した図形を定義する。
	SYME CVn L#ℓ	指定した直線に対称な図形を創成する。
曲線の接続	CONTINUE CVn_j COPY＋CVn_i COPY－CVn_{i+m} CEND CVn_j	CVn_jとして定義した曲線をつなげていき曲線をCVn_jとして登録する。 －はCVn_{i+m}の定義方向を逆にすることを表す。
	CONTINUE CVn_j LOOP(m)□DX,□DY,□DA TRAN CVn_i LEND CEND CVn_j	CVn_iをDX，DY，DA同時刻みで移動させながらコピーし，連続曲線とする。 mは繰り返し数。 （CVn_iを1回目にする）
	CONTINUE CVn_j LN/ ｝直接定義モード CR/ ｝ CEND CVn_j	直接定義モードのLN/,CR/をCONTINUEブロック内で利用できる。

6.3.3　曲面形状定義機能[3)]

ＡＰＴ系の言語では，図6.3で示したように，工具の動きを三つの制御面で指定し，工具の軌跡が創成する曲面となる。簡単な解析曲面であれば，この手法でも形状定義は容易であるが，モールド金型に含まれるような複雑な曲面に対しては，その定義は難しい。また，工具の動きを逐一指定するのも難しい。

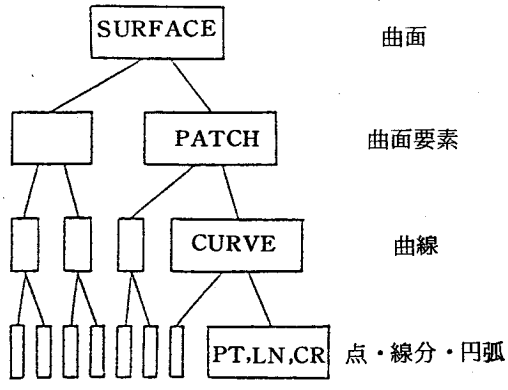
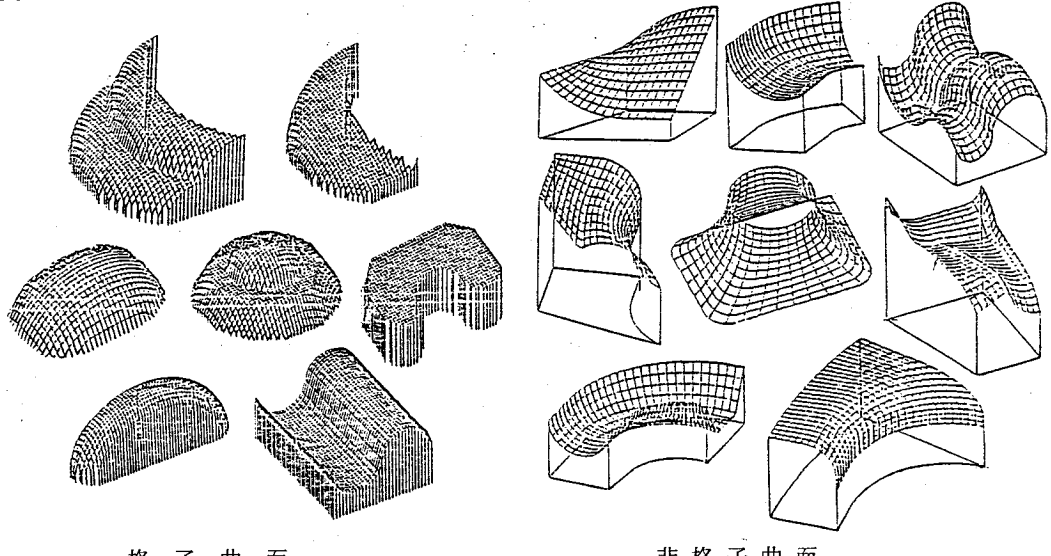


図6.6　曲面形状のトリー構造

そこで，本システムにおける曲面形状定義言語では，図面に表された曲面の断面曲線の定義と曲面補間法の選択だけで曲面形状が簡単に定義できるように工夫している。

曲面は図6.6に示すトリー状の構造で定義する。この構造の最下層は，曲線が点・線分・円弧で構成されていることを表している。曲線は第6.3.2項で示した2次元形状定義言語を用いて定義する。



格子曲面　　非格子曲面

図6.7　曲面要素

曲面要素は，曲面の特性によって１５種類に分類してある。図面に表された断面曲線を定義した後，断面曲線の本数と断面曲線間の相互の位置関係から，曲面の特性を判断し適切な曲面要素を選択する。１５種類の曲面要素は，図6.7に示す格子曲面と名付けたものと，非格子曲面と名付けたものの二つに分類され，モデリング技術は第５章で述べた立体セグメントの応用である。以下において，曲面要素のモデリング手法を説明する。

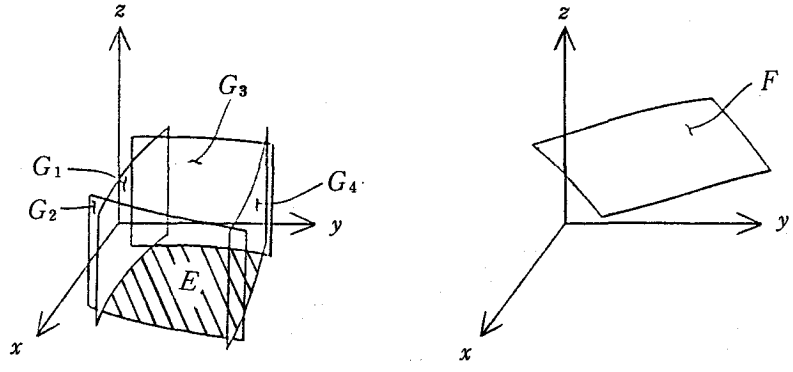


図6.8　曲面要素の表現

図6.8の座標系において，x y平面上に定義域Eを定める。これをz軸方向に掃引して得られるx y平面に垂直な柱状立体の表面をG，その内部の半無限柱状領域をΣとする。これに，定義域Eにおいて一価関数として表される曲面Fを配置し，領域ΣにおいてFとx y平面にはさまれた領域を立体セグメントSとする。

格子曲面は，立体セグメントSによって形状表現し，非格子曲面は曲面Fで形状表現したものである。ただし，非格子曲面は格子曲面のサブセットではなく，曲面Fを格子曲面の場合，x，yをパラメータとした$z=f(x, y)$で表現するのに対し，曲面に沿った二つのパラメータで表現している。このパラメータは，刻みを等間隔にすると対応する曲面の境界曲線上の位置が，一定幅の曲線長をもつ。そのため，加工に際して有効である。

曲面（SURFACE）は，格子曲面の場合，第5.3.3項で述べた集合演算を用いて，曲面要素（PATCH）の合成，部分除去を行い定義する。非格子曲面では，曲面要素（PATCH）間の境界曲線を境に接続していくことで，曲面（SURFACE）を定義する。格子曲面と非格子曲面の混在は，非格子曲面と，この非格子曲面の定義域E内にある格子曲面間においてのみ，どちらの曲面上の値を優先させるかの指定が可能である。図6.9に曲面（SURFACE）定義の三つの方法を示す。

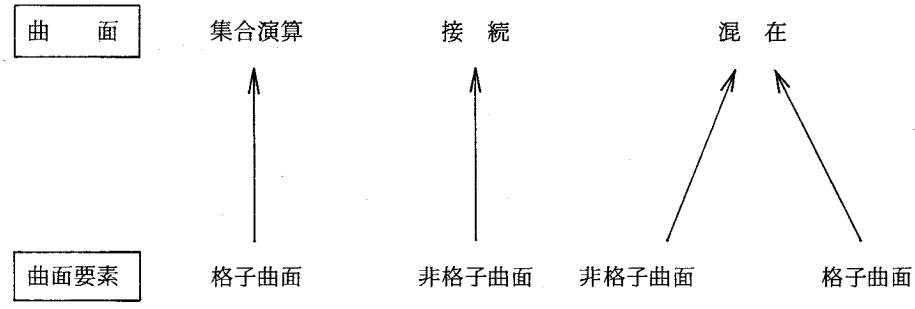


図6.9　曲面定義の方法

曲面形状定義文を表6.3に示す。このBEGIN～ENDブロックにおいて，曲面要素（PATCH）の接続，合成を表6.4に示すオペレータを用いて行う。

曲面要素定義文を表6.5に示す。Gで始まる名前をもつ曲面要素が前述の格子曲面で，Fで始まるものが非格子曲面である。定義文中で用いた記号の意味を表6.6に示す。また，図6.10に格子曲面の機能を，図6.11に非格子曲面の機能を示す。これらの図では，曲面要素の定義に必要な断面曲線の本数と相互の位置関係が示してある。曲面形状定義言語の特徴を示す。

(1)　曲面形状の定義と加工動作定義を独立して行える。

(2)　曲面要素の接続，重ね合せによって複雑な曲面（SURFACE）が定義できる。

(3)　曲面要素の重ね合せに際して，相貫線データを必ずしも必要としない。

(4)　曲面の構成要素（曲面要素，曲線）を明確に記述できるので，後の工具経路製作において，部分加工指定が容易になる。

表6.3　曲面形状定義文

BEGIN　SURFn MODE＝[G / F / GF] {BASE＝} [＋ / ＊ / ／ / 0] PATCHm END　SURFn	G; PATCH　nameがGで始まる曲面間の合成を行う。 F; PATCH　nameがFで始まる曲面をML＃断面曲線方向に連続化する。 GF; PATCH　nameがFで始まる曲面とGで始まる曲面の混在。ただし，Fで始まる曲面はBASE＝を用いて定義する。 G；＋，＊，／，0 F；＋ GF；＋，／　} のオペレーション使用可能。

表 6.4　オペレータ

＋	MODE＝G，GF の場合，曲面間の合成は和集合となる。 MODE＝Fの場合，曲面要素間の境界曲線を境に接続する。
＊	曲面要素の合成は積集合となる。
／	既に定義されてある曲面形状であっても，本オペレータ指定を持つ曲面全体が，合成にあたって，新たな形状になる。
0	既に定義された曲面形状と重なる部分は積集合で合成され，重ならない部分は和集合で合成される。

表 6.5　曲面要素定義文

一般形

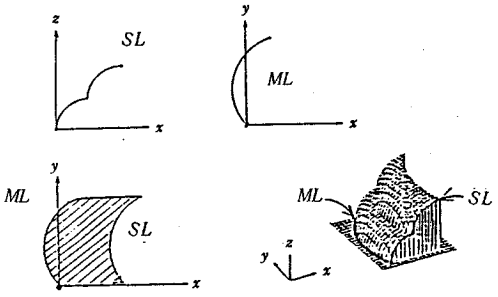
PATCHn＝name／(x，y，z)，TH＝0／ML＃，SL＃／(，)

name
GDRW／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，SL＃n／
GTWI／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，SL＃n，SL＃n／
GDOM／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，SL＃n，SL＃n／
GSPN／(x_0，y_0，z_0)，TH＝$\theta°$／SL＃n／PH＝0
GTUN／(x_0，y_0，z_0)，TH＝$\theta°$／SL＃n／T＝±1
GSTR／(x_0，y_0，z_0)，TH＝$\theta°$／SL＃n／LS＝ℓ_S
GLET／(x_0，y_0，z_0)，TH＝$\theta°$／SL＃n／LS＝ℓ_S
FROT／(x_0，y_0，z_0)，TH＝$\theta°$／SL＃n，SL＃n／(x_2，y_2，z_2)
FSID／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，SL＃n／
FSAL／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，SL＃n／
FVID／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，SL＃n，SL＃n／
FVAL／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，SL＃n，SL＃n／
FPIN／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，ML＃n，SL＃n／(x_2，y_2)
FCIR／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，ML＃n，SL＃n／
FPAT／(x_0，y_0，z_0)，TH＝$\theta°$／ML＃n，ML＃n，SL＃n，SL＃n／(x_2，y_2)

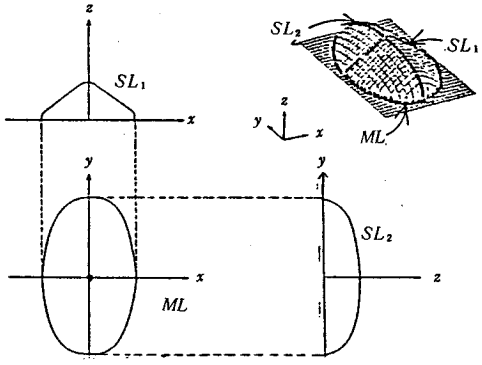
表 6.6　曲面要素定義文で用いた記号の意味

(x_0，y_0，z_0)	基準座標系に対するパッチ座標系の並進変位量を表わす。
TH ＝ $\theta°$	基準座標に対するパッチ座標系の回転変位量を表わす。
ML＃，SL＃	断面曲線NOを指定する。

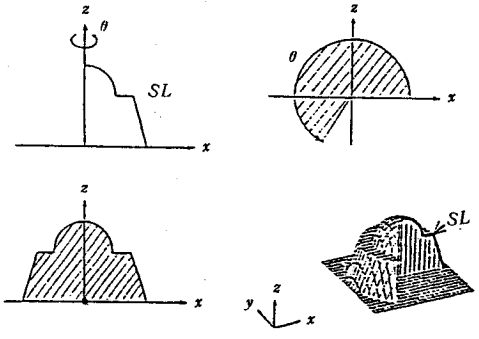
GDRW/(x_0 , y_0 , z_0) , TH=θ°/ML#n , SL#n/



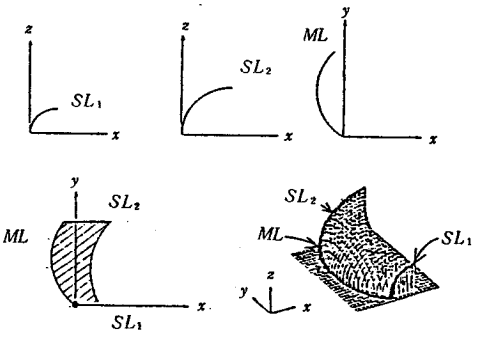
GDOM/(x_0 , y_0 , z_0) , TH=θ°/ML#n , SL#n , SL#n/



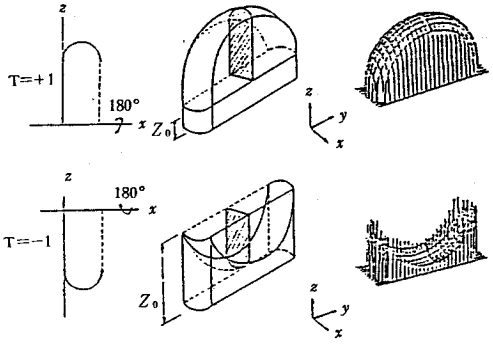
GSPN/(x_0 , y_0 , z_0) , TH=θ°/SL#n/PH=θ°



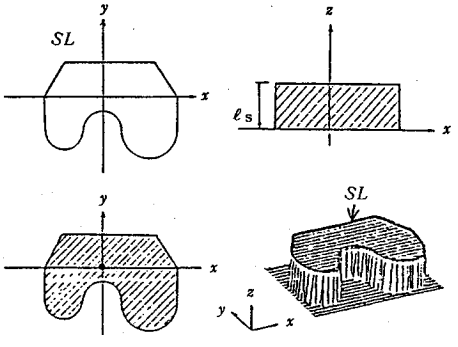
GTWI/(x_0 , y_0 , z_0) , TH=θ° /ML#n , SL#n , SL#n/



GTUN/(x_0 , y_0 , z_0) , TH=θ°/SL#n/T=±1



GLFT/(x_0 , y_0 , z_0) , TH=θ°/SL#n/LS=ℓ_s



GSTR/(x_0 , y_0 , z_0) , TH=θ°/SL#n/LS=ℓ_s

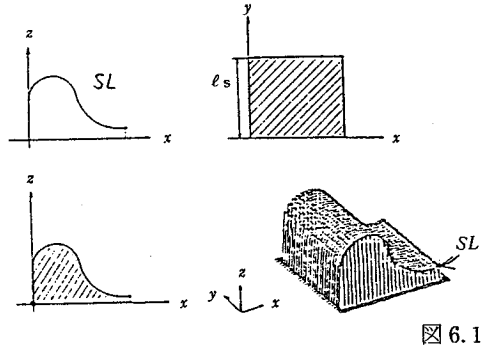
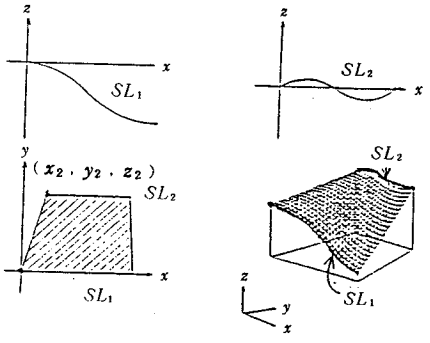
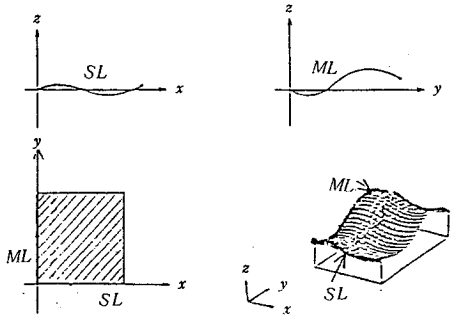


図6.10　格子曲面の機能

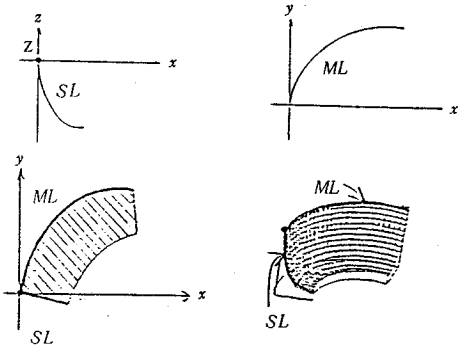
FROT／(x_0, y_0, z_0), TH=θ°／SL#n,
SL#n／(x_2, y_2, z_2)



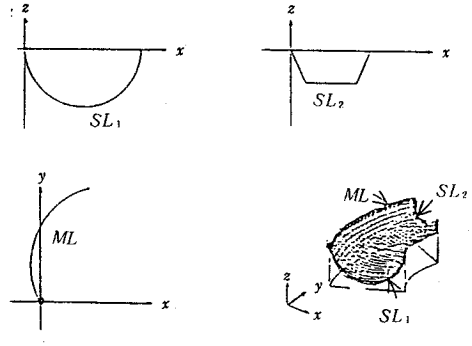
FSAL／(x_0, y_0, z_0), TH=θ°／ML#n, SL#n／



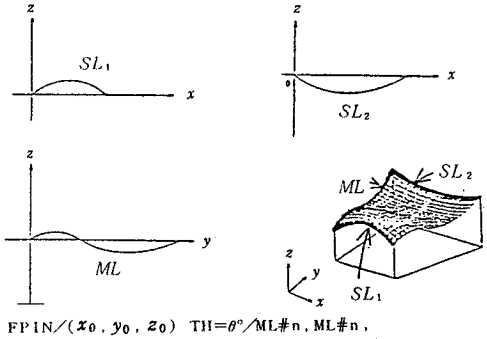
FSID／(x_0, y_0, z_0) TH=θ°／ML#n, SL#n／



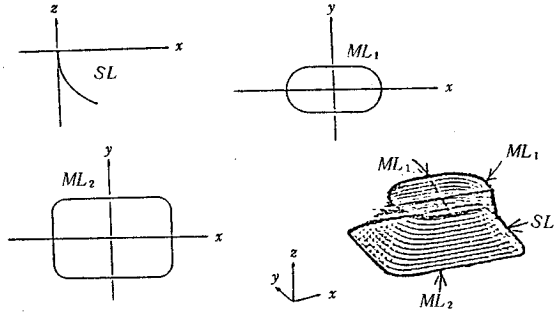
FVID／(x_0, y_0, z_0), TH=θ°／ML#n, SL#n, SL#n／



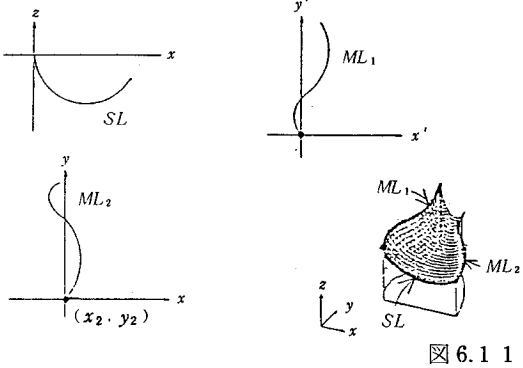
FVAL／(x_0, y_0, z_0) TH=θ°／ML#n, SL#n, SL#n／



FCIR／(x_0, y_0, z_0), TH=θ°／ML#n, ML#n, SL#n／



FPIN／(x_0, y_0, z_0) TH=θ°／ML#n, ML#n,
SL#n／(x_2, y_2)



FPAT／(x_0, y_0, z_0), TH=θ°／ML#n, ML#n, SL#n,
SL#n／(x_2, y_2)

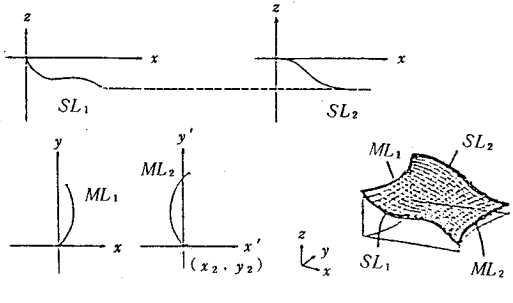


図 6.1 1　非格子曲面の機能

6.3.4　加工動作定義機能[5)]

ＮＣ指令の作成は，個々の曲面要素（PATCH）ごとにも，曲面全体（SURFACE）でもできる。加工機能は，図6.12に示すように，ＰＲＯとよぶ単位と，ＣＵＴとよぶ単位がある。ＰＲＯはＭＩＬＬ～ＭＥＮＤ で囲まれたブロックで定義する。

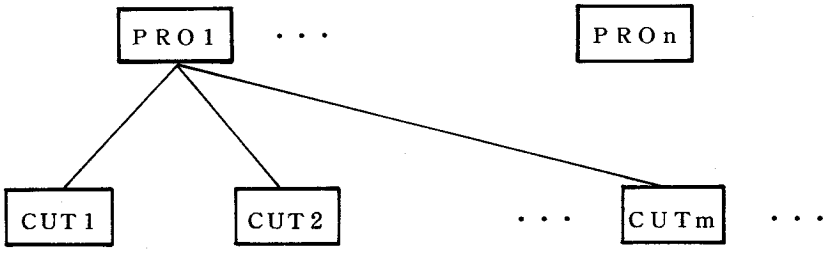


図 6.12　加工で設定できる単位

PRO，CUT の単位を加工工程とどのように対応づけるかは，利用者の自由であるが，標準的と思われる対応づけを以下に示す。

PRO を一つの加工工程と考える。工具の変更，変換，ワークのチャック位置変更等，先へ進めるには人が介在する必要のある加工の範囲を PRO として設定する。全ての工程に人が介在する必要がなければ PRO は一つ設定すればよい。工具経路パターン，送り量，ピックフィード量等の加工条件が，同一である単位を CUT として設定する。加工動作定義文を表6.7に示し，加工条件定義パラメータを表6.8に示す。

このように，加工方法の指定と加工順序の指定を PRO と名づけたブロックと，ブロック内の局所変数 CUT によって行えるという特徴がある。

工具半径分の曲面のオフセットは，システムが自動的に計算する。この計算は，曲面要素個々にオフセットを実施し，オフセットを行った曲面要素をもう一度接続，重ね合せることにより，曲面全体（ SURFACE ）のオフセットを得る手法である。

加工の機能としては，細井[6)]が有効性を示しているドリルによる荒加工用ＮＣ指令の自動生成（図6.13），平面上に作成した模様を任意の方向から曲面上に投影し曲面上に模様を削り込むＮＣ指令の生成（図6.14）もできる。また，治具などの工具侵入禁止領域を格子曲面として定義すると，形状の内と外が識別できるので，工具干渉の自動回避をシステムに行わせることができる。

表6.7　加工動作定義文

文	文の意味
MILL PROn	第nプロセスの加工
STAGE= [ROUPH / FINISH / DIRECT]	POUPH → 荒加工 FINISH → 仕上げ加工 DIRECT → 工具経路直接定義
CUTn= [PACHn / SURFn]	同一パラメータで加工するPACH，SURFを指定する。
パラメータ設定	CUTnで指定した順に加工される。
MEND PROn	パラメータは，表6.8を参照

表6.8　加工条件定義パラメータ

文	文の意味
TYPE= [BOLL / FLATT / DRILL]	工具の種類 BOLL；ボールエンドミル FLATT；フラットエンドミル DRILL；ドリル
TOLR= □	工具半径指定 □ mm
CLAR= □	仕上げ代（クリアランス）指定 □ mm
STEPm= □	送り量指定 □ mm
PICKm= □	ピックフィード量指定 □ mm
ORGN=(, ,)	工具原点座標の設定

表6.8　加工条件定義パラメータ　（つづき）

文	文の意味
PATNm=[LP / HZ / FZ]	ピックフィードするときの工具経路の設定
DIRCm=[XX / YY / OF / SP / ST]	工具経路のパターンの指定
DOWN= □	フラットエンドミルを用いた荒加工の一回当たりの彫り込み量 mm
ICVR=[GO / EP / HI]	他の曲面が送り方向に存在する場合の処理
APRC=[DOWN / LN / CR / (,,)]	工具原点から切削開始点への工具軌跡の定義
ESCP=[UP / LN / CR / (,,)]	切削終了点から工具原点への工具軌跡の定義
DOMAIN=X_{min}, X_{max}, Y_{min}, Y_{max}	加工の全領域（直方体）の設定
AREAm=CVn／[IN / OUT]	部分加工する場合の領域限定 領域境界曲線をＣＶｎ（閉曲線に限る）で指定し，曲線で囲まれた領域の内側（IN）か外側（OUT）を指定
WORK= □	ワークの高さ

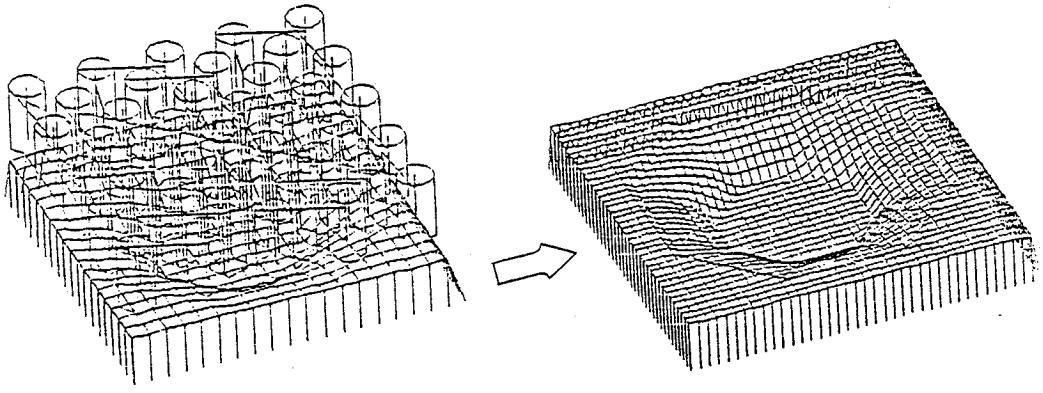


図6.1 3　ドリルによる荒加工の例

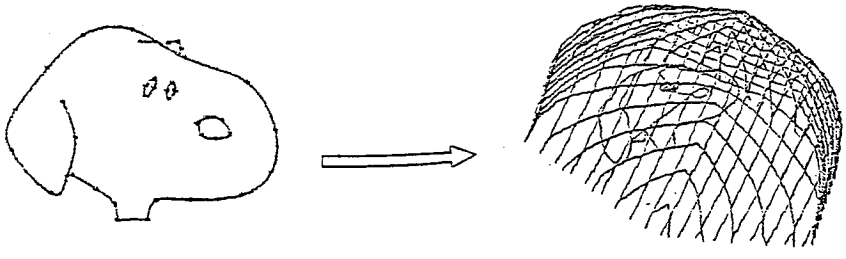
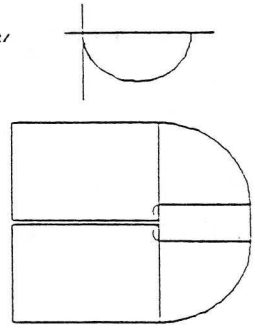


図6.1 4　模様の作成

6.4　適用例

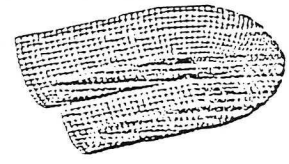
本システムの適用例を示す。図6.1 5は，簡単な形状の定義からＮＣ指令の作成までのプログラム例である。このように，図6.6に示したトリー構造の下から上へ順次定義していけば曲面の定義ができる。また，ブロック構造を導入したことで，簡潔なプログラムとなっていることがわかる。図6.1 6に比較的複雑な曲面加工における工具経路シミュレーション例を示す。

```
 10  BEGIN CV1
 20   SPT/(0.0)
 30   CR/O(8.35.0)/8.3SLR/
 40   EPT/(16.7.0)
 50  END CV1
 60 •
 70  BEGIN CV2
 80   SPT/(0.0)
 90   CR/O(7.0)/7LR/
100   EPT/(14.0)
110  END CV2
120 •
130  BEGIN CV3
140   SPT/(0.0)
150   LN/(0.0),(22.5.0)
160   EPT/(22.5.0)
170  END CV3
180 •
190  BEGIN CV4
200   SPT/(0.0)
210   CR/O(0.-14)/14RR/
220   EPT/(14.-14)
230  END CV4
```



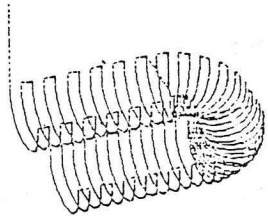
2次元形状定義

```
 10  PATCH1=FV(ID/(-22.5.17.05.38).TH=0/ML#3.SL#1.SL#1/
 20  PATCH2=FV(ID/(0.17.05.38).TH=0/ML#4.SL#1.SL#2/
 30  PATCH3=FV(ID/(14.5.05.38).TH=0/ML#5.SL#2.SL#2/
 40  PATCH4=FV(ID/(14.-8.05.38).TH=0/ML#6.SL#2.SL#1/
 50  PATCH5=FV(ID/(0.-17.05.38).TH=0/ML#7.SL#1.SL#1/
 60 •
 70  BEGIN SURF1
 80   MODE=F
 90   =PATCH1
100   =PATCH2
110   =PATCH3
120   =PATCH4
130   =PATCH5
140  END SURF1
```

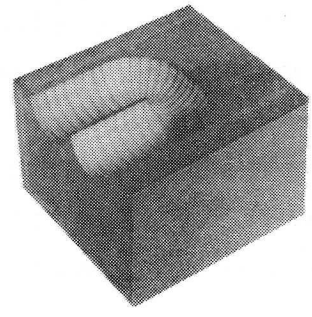


加工形状の定義

```
 10  MILL PRO1
 20   STAGE=FINISH
 30   TYPE=BALL
 40   TOLR=3
 50   CLAR=0.3
 60   WORK=15
 70   ORGN=(300.0.50)
 80   APRC=DOWN
100   CUT1=PACH1
102   CUT2=PACH2
103   CUT3=PACH3
104   CUT4=PACH4
105   CUT5=PACH5
110    STEP0=0.5
120    PICK0=5
130    PATN0=NZ
140    DIRC0=ST
150   ESCP=UP
160  MEND PRO1
```



工具経路計算



加工結果

図6.15　適　用　例

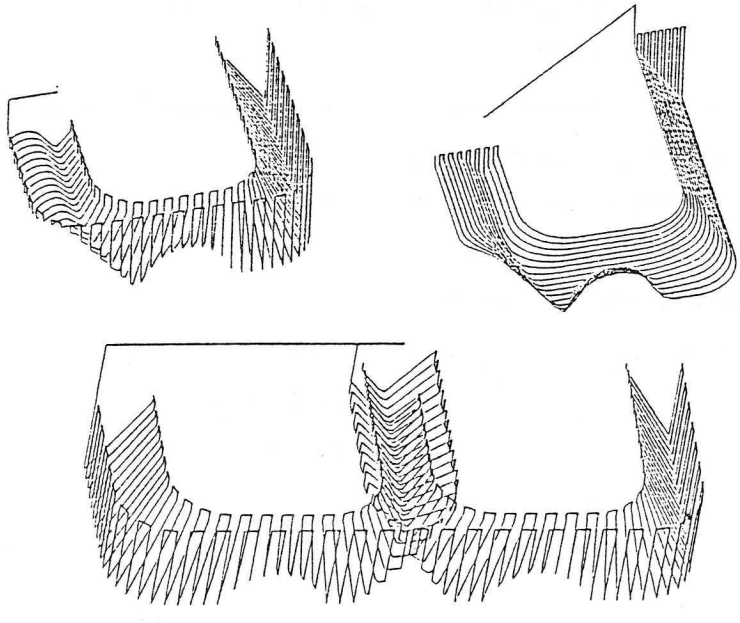


図6.16　工具経路シミュレーション例

6.5 結　言

本章で示したシステムは，意匠性に富んだ曲面を多く持つ家電製品のモールド金型の製作に利用している。システムの特徴をまとめる。

(1) 金型図面に表されている寸法などのデータに従って断面形状や曲面要素を言語を用いて定義し，これらを組み合せて複雑な曲面を容易に定義できる。

(2) 定義された曲面の加工法を指定することにより，荒加工から仕上げ加工までのＮＣ指令を自動的に作成できる。

(3) タブレットなどを用いて入力した模様や文字などの曲線を曲面に彫り込むＮＣ指令を作成できる。

(4) フロッピーディスクを用いてＮＣ指令を転送すると，紙テープ１０巻分のデータを連続的にＮＣ装置へ供給することができ長時間無人加工ができる。また，ＮＣ指令の編集，変更，管理が容易である。

今後の展望として，計算機やプログラム言語の知識をあまりもたない意匠設計者，ＮＣ工作機械のオペレータが，簡単に使えるシステムにする必要がある。そのため，曲面要素の選択と曲面要素の重ね合せによる曲面定義を手助けする機能の開発が望まれる。

参　考　文　献

1） S.Hori(ed.): APT Part Programming, IIT Research Institute, McGraw-Hill(1967).

2） P.Naur (ed.): Revised Report on the Algorithmic Language ALGOL 60, Computer Journal, 5 (1963) 349.

3） 塩谷景一，山田祐子： パーソナルCAD／CAMシステムの開発，昭和61年度精密工学会春季大会学術講演論文集（1986）239.

4） MEDIAPT，三菱電機㈱(1983).

5） 山田祐子，塩谷景一： 3次元金型パソコンCAMシステムの開発，第2回フレキシブル・オートメーションシンポジウム講演論文集(1985) 133.

6） 細井俊明： 新しい金型型面加工法，日刊工業新聞社(1983).

第7章　人工衛星用3次元ＣＡＤシステム

7.1　緒　言

3次元ＣＡＤ／ＣＡＭシステムは，製品設計や製品の生産プロセスを支援する手段として期待がよせられている。第5章でも述べたモデリング手法は，システムの核となる理論として，種々な研究が行われている。

ところで，人工衛星にはスピン衛星や三軸衛星などの種類があり，それらの大きさ，重量は種々であるが，基本的に人工衛星は通信系・姿勢およびアンテナ制御系・電源系・二次推進系・熱制御系などのサブシステムから成っている。また，各サブシステムを構成する機器は精密製品であり，かつ，その数は多い。しかも，発射時から各段階のモータ燃焼時における環境条件に耐える強度と剛性が必要である。このように，極めて複雑なシステムを苛酷な環境条件で使用するため，設計とモックアップと呼ばれる実物大モデルによる試験を繰り返し行なう。何十人もの設計者が，二年以上もかけて開発を行なうことも珍しくはない。そのため，3次元ＣＡＤシステムによる設計の支援が必要となる。

本章では，人工衛星を対象とした3次元ＣＡＤシステムの開発例について述べる。特に，人工衛星の形状モデリング手法について詳しく説明する。ところで，第5章で説明したＣＳＧ，B-Repsのモデリング手法は，データの特性を対比するとそれぞれ一長一短があるため，双方のデータを持つdual representationという手法も提案されている[1)]。しかし，工業的応用を考えた場合データ特性の問題だけでなく，これらのモデリング手法における形状創成プロセスが実際の設計プロセスと一致しないという問題が残されている。

そのため，従来のモデリング手法と比べるとサブセットではあるが次に示す特徴のあるモデリング手法を開発し，これを本システムに採用している。

(1)　人工衛星の設計で頻出する形状をプリミティブに設定したため，プリミティブの単純な接合で全体の形状が定義でき，衛星の設計に必要な各種の技術計算モジュールに対する入力データを自動的に容易に作成できる。

(2)　応用解析モジュールで必要なデータの分析に基づいてデータ構造中に陽に表すデータを選択したためデータ量が少ない。

(3)　隠面処理などの図形処理を，主にディプスバッファを持つグラフィックディスプレイによるハードウェアで行う方式にしたため，プリミティブのデータ構造が単純になり各種処理を高速に行える。

以下において，まずシステムの概要を説明し，次に人工衛星のモデリング手法と応用解析モジュールとのインタフェースを述べ適用例を示す。

7.2 システムの構成

人工衛星は，生産台数が極めて少なく，宇宙空間という特殊環境下で使用され，小型軽量，高信頼性，耐環境性を徹底的に追求する必要がある。そのため，構想設計，基本設計に多くの人手と期間を必要としている。図7.1にシステム構成を示す。

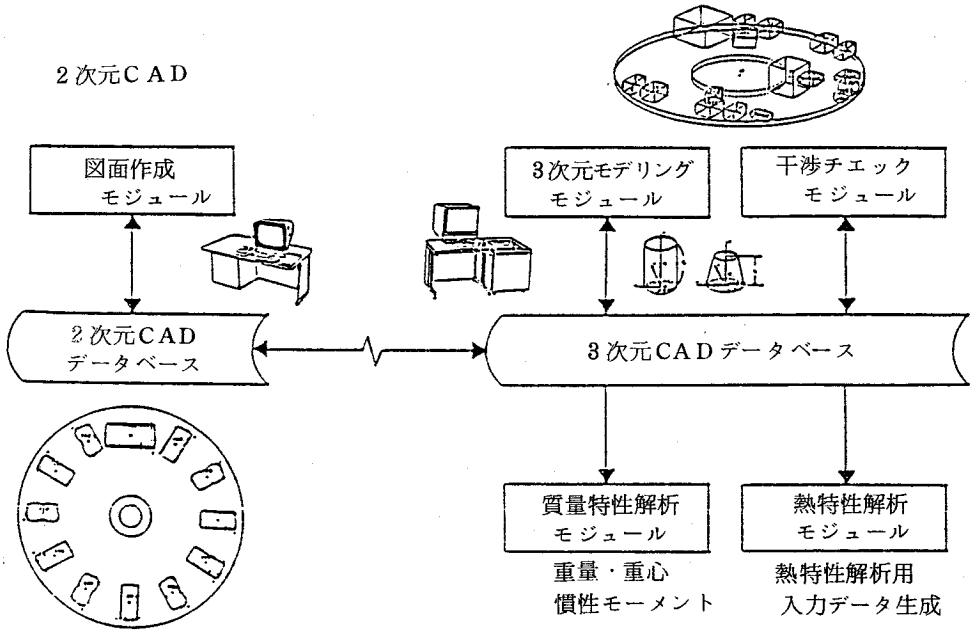


図7.1　システム構成

本システムは図面情報の取り扱いを中心とした2次元CADサブシステムを含み，設計過程に応じて，2次元図形処理機能と3次元形状処理機能を使い分ける。具体的には，生産現場への指示事項を含めた詳細設計は図面中心の2次元図形処理機能を用いて進め，従来，モックアップ(mock-up）とよばれる実物大モデルを用いて検討した機器の搭載可能性，および，質量バランスなどの各種技術的な検討は3次元形状処理機能を用いて行なう。

搭載機器の配置設計では，まず2次元CADサブシステムを用いて図面を作成し，2次元の配置を決める。この2次元の配置データは自動的に3次元CADデータベースへ引き渡されるので，利用者は高さ方向の配置データのみ入力すれば搭載機器の空間的な配置を決めることが出来る。この場合，あらかじめ搭載機器を立体として定義しておき，図面として表現された2次元データと対応させておく。一方，3次元形状処理機能を用いて空間の配置が決まれば，平面上の配置データは2次元CADサブシステムへ転送されて図面が修正される。

このように，2次元図形処理機能と3次元形状処理機能とは個別のデータを持つが，同一の形状

間はデータの対応づけが行なわれていて，一方の機能を用いて配置修正を行えば，他方のデータも自動的に修正される。

次に，応用解析モジュールの概要を説明する。

(1) 干渉チェックモジュール

人工衛星は，内部空間が狭いにもかかわらず数多くの機器を搭載する必要があるため，実装可能性が問題となる。最終的にはモックアップを作成し実物大モデルによる検討が必要であるが，モックアップを繰り返し作成するのは時間的・費用的に問題である。そこで，システムの干渉チェックモジュールを用いて，搭載機器を立体として定義できるので，干渉チェックはシステムで自動的に行なえる。

(2) 質量特性解析モジュール

人工衛星の設計には，質量中心の位置・慣性モーメントなどの質量特性データがきわめて重要である。搭載機器の配置によりこれらの数値は影響を受ける。そのため，配置変更の際には必ず本モジュールを用いて質量特性を調べる。

(3) 熱特性解析モジュール

宇宙環境下にある人工衛星の熱放射・熱伝導の特性を計算するモジュールである。搭載機器の熱伝導率，および配置密度により人工衛星全体としての熱特性が変わるため，配置変更があれば本モジュールを用いて解析を再度行なう必要がある。

(4) そ の 他

上述した三つのモジュールは，人工衛星の設計専用に開発されたプログラムである。なお，構造解析は有限要素法に基づく汎用プログラムを利用しているが，この際必要となるメッシュデータは自動的に作成している。

7.3 人工衛星のモデリング

7.3.1 専用モデリング手法の必要性

形状モデリングの表現形式では，論理的に必要最小限のデータのみ持つ形式とすると記憶容量は少なくてすむが，データ抽出に時間がかかる。一方，要求されるデータをすべて陽に持つと，データ抽出には時間を要しないが記憶容量を多く必要とし，データ変更に際してはデータ間の矛盾を引き起こさないように整合性を保って処理するために余分の計算時間を要する。人工衛星を設計するときには次に示すデータを取り扱う必要がある。

(1) 形状の頂点・稜線・面のデータ

(2) 形状の内と外を判別出来るデータ

(3) 人工衛星がどの様に組み立てられているかを示すデータ

さらに形状以外に次のデータも必要である。

(4) 質量・慣性モーメント・熱特性などの技術データ

(1)~(4)に示したデータは，人工衛星の設計に使う必要最小限のデータである。しかし，これらのデータのみでデータベースを構築すると，例えば図形処理で用いられるデータの抽出に時間を要することになる。そこで隠面処理などの図形処理をソフトウェアで行うならば，副次的ではあるが処理の高速化のために，例えばウィングドエッジ（winged−edge）に代表されるような冗長度のあるトポロジー構造をデータベースに採用する必要がある。。

形状合成は一般に，集合演算を用いてプリミティブとよばれる円柱，直方体などの基本形状を組み合わせて行なう。この場合，差の集合演算と積の集合演算では，定義される形状内部に含まれないプリミティブ間の形状操作となることがある。このことは形状内外の判定処理，形状処理，図形処理などを難しくする場合がある。また，一般に集合演算は長い計算時間を必要とする。したがって実用的なシステムの構築にあたっては，応用対象の特徴に注目し計算時間を軽減するための工夫が必要になる。

第5章と同様に物体全体をオブジェクトと呼ぶことにし，プリミティブとオブジェクトのデータ構造の関係を考える。第5章でも述べたように，ＣＳＧにおけるオブジェクトデータ構造は形状創成過程とどのプリミティブで合成されているかのデータを持つ。各種処理はプリミティブのレベルにまで戻り形状創成過程を繰り返すことで実行される。一方，一般に B−Repsでは，オブジェクトのデータ構造は，オブジェクト自身を頂点・稜線・面によって表現してあり，どのプリミティブで合成されているかのデータは基本的に存在しない。そのため各種処理はプリミティブのレベルにまで戻って行われることはない。これらから，記憶容量，形状修正に関してはＣＳＧが優れ，図形処理速度に関しては，Ｂ−Repsが優れていることが分かる。つまり，ＣＳＧあるいはB−Repsのどのデータ構造を用いるかによってシステムの特徴が大きく変わる。そのため実用的なシステムを構築する場合，記憶容量ができるだけ少なくてしかも図形処理速度が速くなるようなオブジェクトのデータ構造を用いる必要がある。

そこで，本システムのように応用対象が決まっている場合には，専用モデリング手法を開発することにより各種処理の計算時間の軽減やデータベースの小容量化が可能となり，システムの実用性が高くなることが期待できる。

以上述べた背景を考慮して，人工衛星を対象にした専用のモデリングが手法を開発した。以下において具体的に本モデリング手法について述べる。

7.3.2 オブジェクトのデータ構造

人工衛星は大規模なシステムであるからサブシステムに分解され，さらにモジュールに細分化さ

れて設計が行われる。ここで示すオブジェクトのデータ構造はこの設計単位と対応づけが出来るように構成されている（図7.2）。つまり，人工衛星全体をオブジェクト，設計上一つの機能単位として扱いたい部分形状をサブオブジェクト，個々の機器をコンポーネントとして定義する。サブオブジェクト間，コンポーネント間は各々が独立した機器であるから重なりがない。そのため，サブオブジェクトを組み合せてオブジェクトを定義する場合と，コンポーネントを組み合せてサブオブジェクトを定義する場合には単純な接合のみでよい。

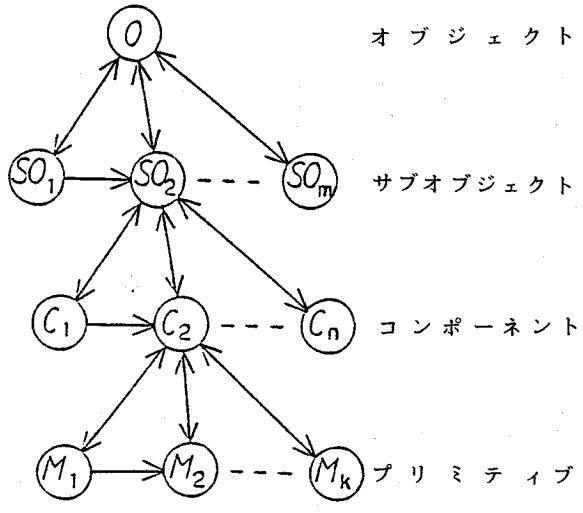


図7.2　オブジェクトデータ構造

コンポーネントはプリミティブの組み合せで定義するが，本システムの形状モデリング手法ではこの方法に特徴がある。図7.2に示したように，コンポーネントはプリミティブの単純な接合で定義される。プリミティブは形状の基本単位であって機器とは限らないから，一般にはプリミティブ間の重なりもありえる。実際に従来のモデリング手法では，例えばパイプ形状は二つの円柱を定義し，直径の大きい方の円柱から小さい方の円柱を引きさる差の集合演算によって作成した。つまり，プリミティブの単純な接合で形状定義は行えなかったのである。しかし，パイプ形状をプリミティブとすると，集合演算を用いないでパイプ形状を作成することができ内径を0にすれば円柱を作成することができる。

そこで，人工衛星の設計で頻出する形状の分析に基づいて図7.3に示す基本的な形状をプリミティブに選び，さらに中実，中空，肉厚のある中空の三つの状態を各プリミティブで設定できるようにして，プリミティブの単純な接合のみでコンポーネントの定義ができるように工夫した。

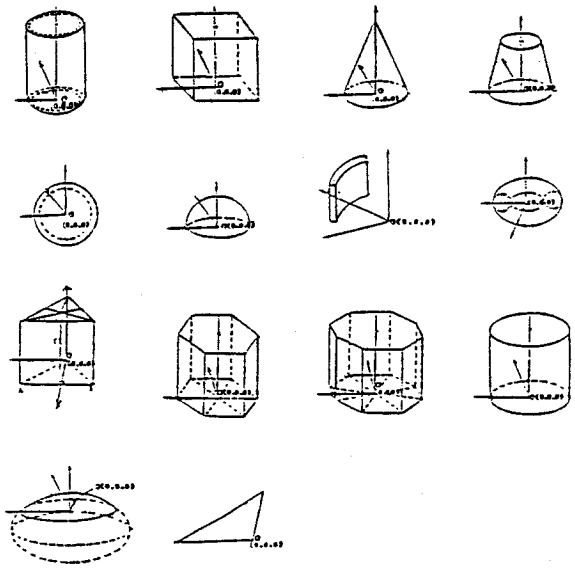


図7.3　プリミティブ

データ構造の詳細を図7.4に示す。T_1, T_2, T_3, T_4 はオブジェクト，サブオブジェクト，コンポーネント，プリミティブの関係を定義するデータテーブルである。この4つのデータテーブルで人工衛星全体の構造が定義される。T_5, T_6, T_7 は各々オブジェクト，コンポーネント，プリミティブ単位の技術データと，個々の形状が属する一つ上位の階層に対する配置データを持つ。T_8, T_9, T_{10}, T_{11} はプリミティブのトポロジーとジオメトリーのデータを持つテーブルで，これらについては次項で述べる。各データテーブルの内容を図7.5に示す。技術データはオブジェクト以外の形状の階層ごとに設定でき，質量特性解析モジュール，熱特性解析モジュールへの入力データの一つとして利用される。

以上述べたように，本手法は下位階層の単純な接合によって上位階層を定義できるから，形状定義は容易で，また，頂点・稜線・面などの形状データはプリミティブの階層のみに持たせても高速に各種処理ができる。一般に集合演算を用いると各種処理が複雑になる。これは例えば差の集合演算を用いると，引き去る側の形状は作成する形状内部には含まれないから，与えた任意の一点が形状の内部にあるか外部にあるかの判定が，単純な接合で形状定義された場合に比べて複雑になることから容易に推察できる。

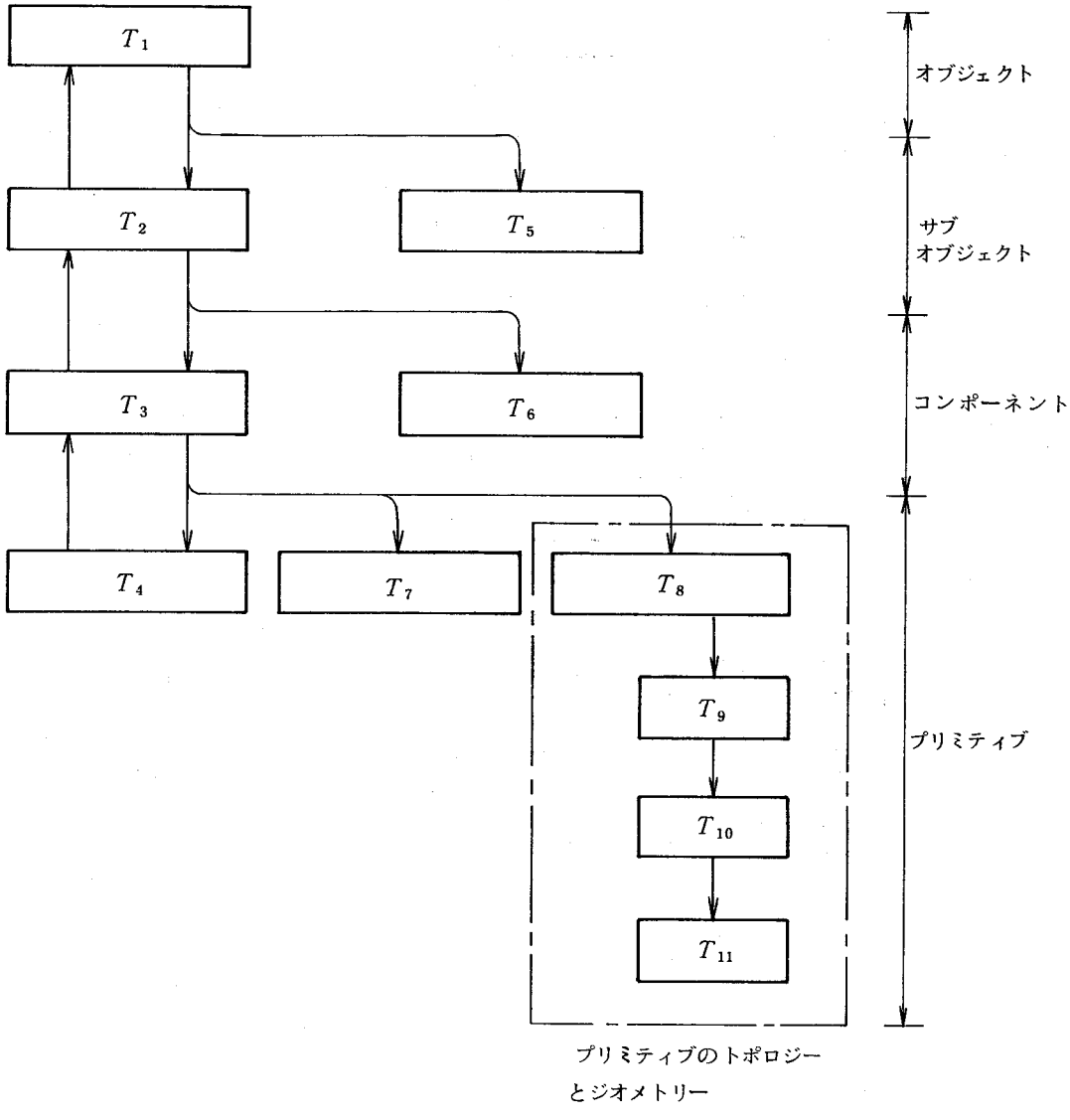


図 7.4　オブジェクトデータ構造の計算機内部表現

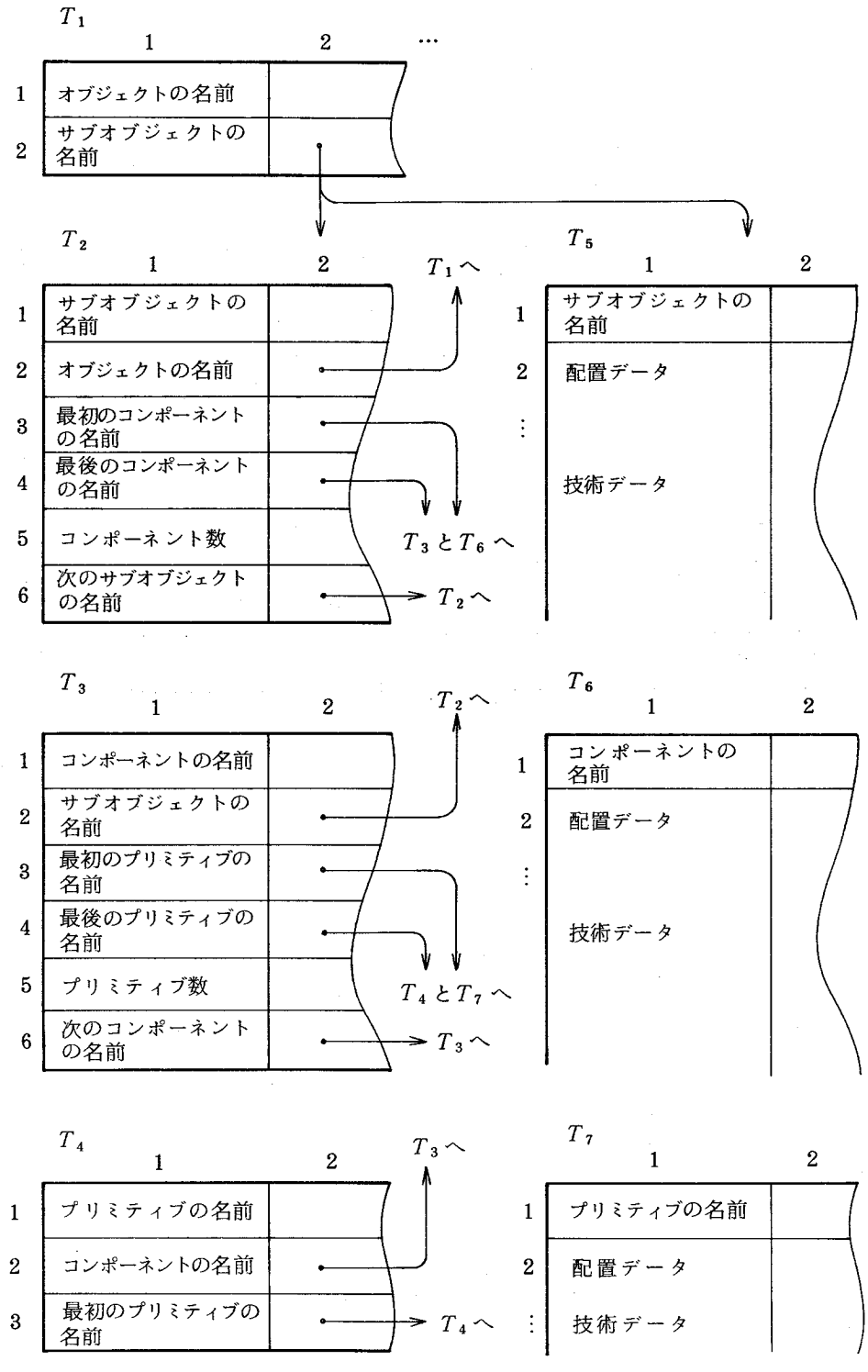


図7.5　データテーブルの内容

7.3.3　プリミティブの内部表現

本項で用いるトポロジーは Baer[5] による表記法を用いる（図7.6）。$t_1:\{t_2\}$ は t_1 は t_2 で記述されることを意味している。V は頂点，e は稜線，f は面を表す。

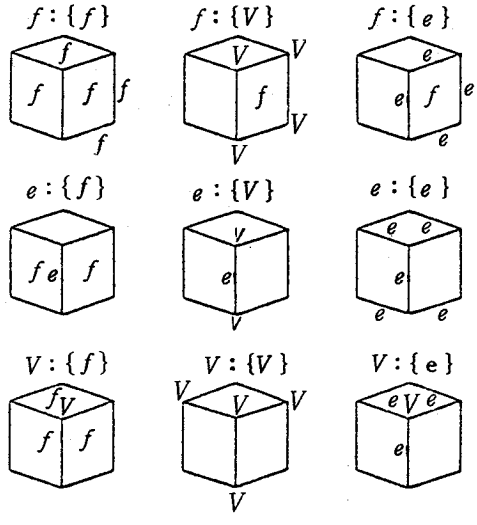


図7.6　トポロジー

本手法におけるプリミティブのデータ構造を図7.7に示す。図形処理に必要な稜線や頂点のデータの取扱いを容易にするため，全てのプリミティブをトポロジー的には多面体として表現する。一

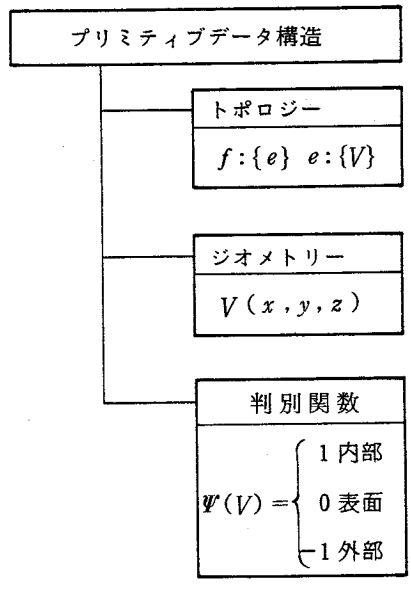
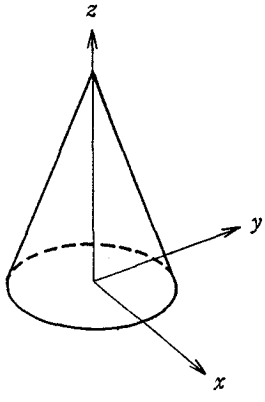


図7.7　プリミティブのデータ構造

方，干渉チェックなど形状の内と外を厳密に判定できることが要求される処理に対応するため，判別関数をデータに含めた。判別関数の意味を円錐を例にして説明する。

図7.8に円錐の判別関数を示す。ここでHは円錐の高さ，$R(z)$は$x-y$平面に平行で高さzの平面で円錐を切断した時に得られる円の半径である。表面上の点を判定するためにεを導入している。すなわち，計算誤差のためx^2+y^2と$R^2(z)$が一致することは少ないため，x^2+y^2と$R^2(z)$の差の絶対値がεより小さければx^2+y^2と$R^2(z)$が一致したと見なす。これらの不等式を用いて形状の内と外を判定する方法を説明する。空間にある任意の点（x，y，z）が，二つの不等式$0 \leqq z \leqq H$と$x^2+y^2 \leqq R^2(z)$を同時に満たす場合この点は形状内部に存在し，同時に満たさない場合は形状外部に存在するとみなす。$0 \leqq z \leqq H$と$|x^2+y^2-R^2| \leqq \varepsilon$の二つの不等式を同時に満足する場合に点（$x$，$y$，$z$）は形状表面に存在するものとみなす。



$$0 \leqq z \leqq H$$

$$x^2+y^2 \leqq R^2(z)$$

$$|x^2+y^2-R^2(z)| \leqq \varepsilon$$

図7.8　判別関数の例

このように判別関数は複数の不等式で構成され，プリミティブの種類によって個有のものである。実際のシステムでは，$\boldsymbol{P}$を空間にある任意の一点とした場合，判別関数$\Psi(\boldsymbol{P})$は，$\boldsymbol{P}$が形状内部に存在する場合1，表面に存在する場合0，形状外部に存在する場合-1の値となるように不等式の判定結果で三つに場合分けされる。

$$\Psi(\boldsymbol{P}) = \begin{cases} 1 & \text{形状内部} \\ 0 & \text{形状表面} \\ -1 & \text{形状外部} \end{cases} \qquad (7.1)$$

プリミティブのトポロジーとジオメトリーのデータは図7.9に示すように，単純な構造で表現されている。これは，稜面処理などの図形処理を主にハードウエアで行うので，図形処理を高速にするために複雑なトポロジーを持つ必要がないからである。これについては次項で述べる。

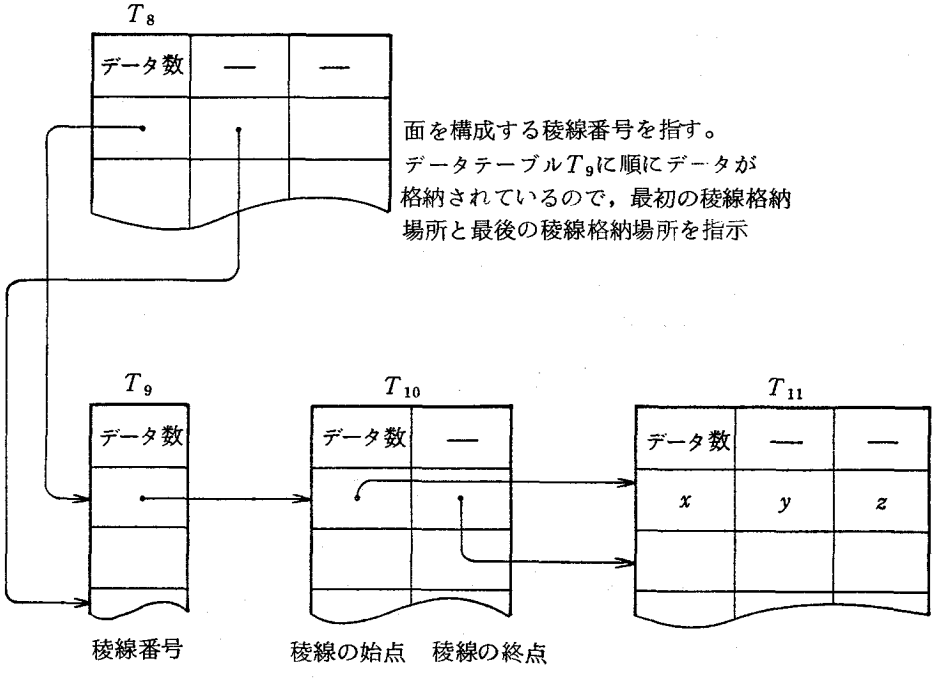


図7.9　プリミティブのトポロジーとジオメトリーの定義

従来の手法として[6]，本手法と同様にトポロジー的には多面体として表現するが（例えば円柱はn角柱），各面の表現をB−spline 式を用いて定義し，形状が空間に占める領域も厳密に定義する試みがある。円柱側面や球などもB−spline式を用いて定義する理由は，曲面と曲面の相貫線計算などの形状処理が全て，B−spline 式を用いた数値計算に統一できるためである。この場合，形状表現能力が高くなり複雑な形状も扱える。しかし，任意の一式が形状内部にあるか，形状外部にあるかの判定は式（7.1）の判別関数を用いる場合に比べると処理速度は遅い。この理由は，B−spline 式はパラメトリック表現のため，形状の内外を判定できる判別関数が複雑になるからである。本手法では，図7.4に示すプリミティブが式（7.1）の$\Psi(\boldsymbol{P})$を容易に作成できることをうまく利用したデータ構造を開発したことにより各種処理を高速に行え，さらに形状が空間に占める領域も厳密に定義できるシステムを実現している。

7.3.4　図形処理用データ構造

ＣＡＤシステムにおいて図形処理は重要な機能の一つである。定義した形状をグラフイックディスプレイに表示することによって，形状の確認や設計上の種々の検討が行える。定義した３次元形状をグラフィックディスプレイ上に２次元投影図表示するには，射影変換，透視変換，隠面処理などの図形処理が必要である。隠面処理は，ある方向から形状を見た場合隠れて見えない面の消去を行うもので，一般に計算時間を長く必要とする。そこで，本システムではディプスバッファ（depth buffer）を持つグラフィックディスプレイを用いて，隠面処理をハードウェアで行う方式にしてある。ディプスバッファは深さ情報（画面垂直方向の値）を入れるメモリーで，このデータを用いて目に近い面が後ろの面を隠す作用を比較回路で行うことができる。図7.10に図形処理用データ構造を示す。

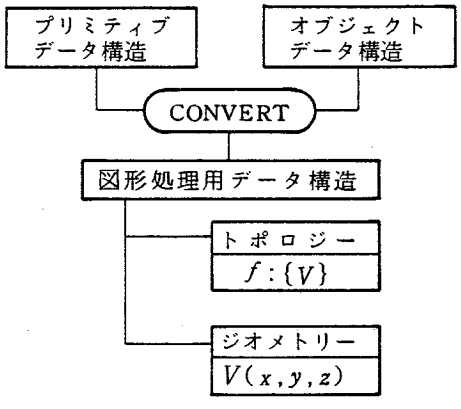
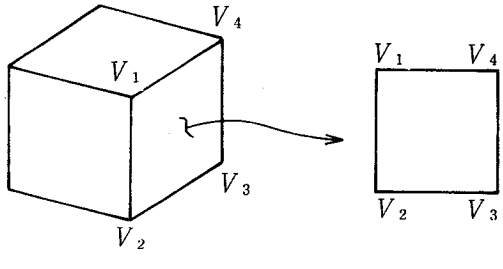


図7.10　図形処理用データ構造

グラフィックディスプレイへの入力データは$f:\{V\}$のトポロジーと$V(x,y,z)$のジオメトリーのみである（図7.11）。このデータはプリミティブデータ構造におけるトポロジー$f:\{e\}$，$e:\{V\}$とジオメトリー$V(x,y,z)$からほぼ直接的に作成できる。つまり，面のデータをグラフィックディスプレイ上に構築する。

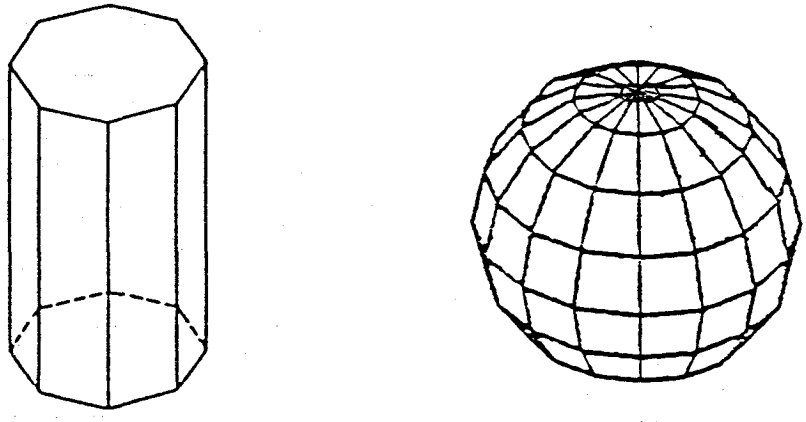


V_1	x_1	y_1	z_1
V_2	x_2	y_2	z_2
V_3	x_3	y_3	z_3
V_4	x_4	y_4	z_4

各面ごとに面の境界を一巡する順番で頂点の座標を与える。

図7.11　グラフィックディスプレイへの入力データ

取扱える面は平面に限るため，円柱，球などの曲面を持つプリミティブは，図7.12に示すように多面体で近似する。図に示すように円柱はn角柱となる。この例では円柱を八角柱の多面体で近似している。



(a) 円　柱　　(b) 球

図7.12　円柱と球の多面体による近似

このようにして，面のデータがグラフィックディスプレイ上に構築されたならば，計算機側に負荷をかけることなく図形処理を行える。つまり，第7.3.3項で示したような単純なプリミティブのデータ構造でも，種々の図形処理を高速に行える。

7.4　応用解析モジュールとのインタフェース

干渉チェックモジュールでは，3次元CADデータベースにおけるプリミティブのトポロジー，ジオメトリー，判別関数およびオブジェクトデータ構造における形状の相互関係を示すデータを利用する。干渉チェックは形状の基本単位であるプリミティブ相互の重なりがあるか否かで判断する。プリミティブ相互の重なりを調べる基本アルゴリズムを二つのプリミティブM_1，M_2を用いて簡単に説明する。プリミティブM_1とM_2が重なっているか重なっていないかの判別は，一方のプリミティブの頂点が他方のプリミティブの形状内部にあるか形状外部にあるかで行う。具体的には，式(7.1)において，プリミティブM_1の頂点を$\boldsymbol{P}$，ΨをプリミティブM_2の判別関数として$\Psi(\boldsymbol{P})$を調べることにより，一つでも$\Psi(\boldsymbol{P})$が1となる頂点があればプリミティブM_1とM_2が重なっていると判定する。次にプリミティブM_2の頂点を$\boldsymbol{P}$，ΨをプリミティブM_1の判別関数とし$\Psi(\boldsymbol{P})$を調べ一つでも$\Psi(\boldsymbol{P})$が1となる頂点があればプリミティブM_1とM_2が重なっていると判定する。　この一連の処理でΨ

($\boldsymbol{P}$)の値が全て－1の場合にプリミティブM_1とM_2は重なりがないものと見なす。このように3次元ＣＡＤデータベースから直接得られるデータだけで干渉チェックが行える。

質量特性解析モジュールと3次元ＣＡＤデータベース間のデータ入出力を図7.13に示す。T_6,

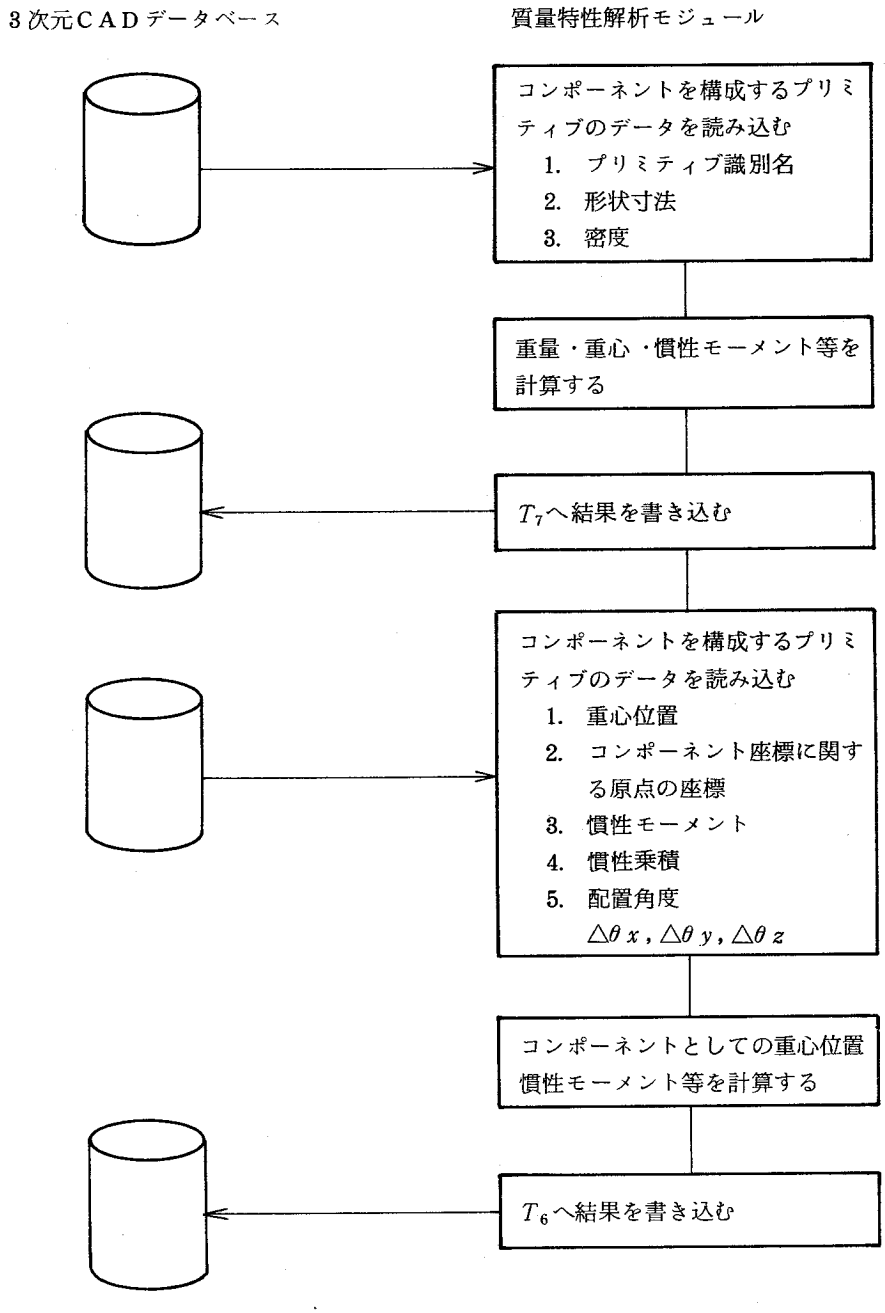


図7.13　質量特性解析モジュールと3次元ＣＡＤデータベースのデータ入出力

T_7は図7.4，図7.5に示したデータテーブルである。プリミティブ識別名は形状定義の段階でデータテーブルT_7に登録される。質量特性の計算式は円柱，球などプリミティブ対応にあらかじめ準備してある。プリミティブ識別名に従って計算式が選択され質量特性の計算が行われる。プリミティブ識別名はアルファベット一文字で，例えば中実の円柱はCで表している。第7.3節で示したように，本モデリング手法はプリミティブの単純な接合によりコンポーネントを定義するため，質量特性計算が容易である。以下に慣性モーメントを例にして説明する。

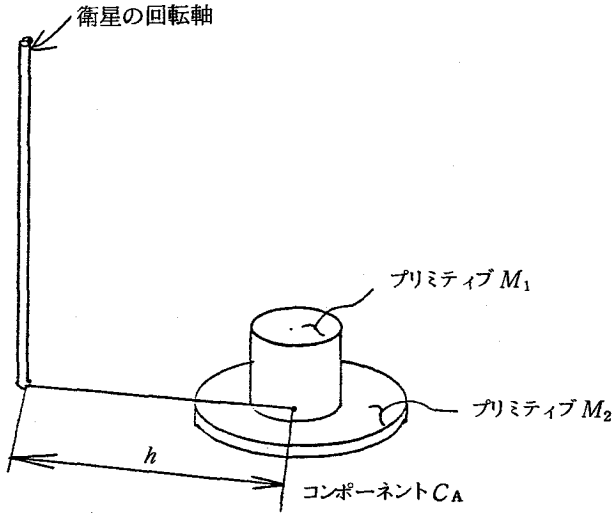


図7.14　慣性モーメントの計算

図7.14に示すようにコンポーネントC_Aは二つの円柱プリミティブM_1とM_2で定義されていると仮定する。プリミティブM_1，M_2の円柱の半径と質量を各々，r_1，r_2およびm_1，m_2とする。また，プリミティブM_1とM_2の円柱の中心軸は同一直線上にあり，かつ衛星の回転軸と平行でhの距離にあるものとする。円柱の中心軸に関する慣性モーメントIは円柱の半径をr，質量をmとするとよく知られている次式で得られる（ただし，密度一様）。

$$I = \frac{r^2}{2} \quad m \qquad (7.2)$$

プリミティブM_1，M_2の慣性モーメントをI_1，I_2とすると式（7.2）と慣性モーメントの平行軸定理から次式で計算される。

$$I_1 = \frac{r_1^2}{2} \quad m_1 + h^2 m_1 \qquad (7.3)$$

$$I_2 = \frac{r_2^2}{2} \quad m_2 + h^2 m_2 \qquad (7.4)$$

コンポーネントC_Aの慣性モーメントをI_Aとおく。コンポーネントC_Aがプリミティブの単純な接合で定義されているので，プリミティブM_1, M_2の個々の慣性モーメントが分っていると，慣性モーメントの和の定理から，次式で計算される。

$$I_A = I_1 + I_2 \qquad (7.5)$$
$$= \frac{r_1^2}{2} m_1 + h^2 m_1 + \frac{r_2^2}{2} m_2 + h^2 m_2$$

円柱の慣性モーメントの計算式 $\frac{r^2}{2} m$はあらかじめ質量特性解析モジュールに準備してある。hはプリミティブの配置データから得られ，r_1，r_2，m_1，m_2は，3次元CADデータベースから直接得られるデータである。つまり，プリミティブの単純な接合で形状全体を定義できるため，慣性モーメントの和の定理などが有効に使えることが本モデリング手法の特徴の一つである。

熱解析モジュールでは3次元CADデータベースのプリミティブのトポロジー，ジオメトリー，オブジェクトデータ構造における形状の相互関係を示すデータと熱容量などの技術データを利用する。ただし，熱解析において利用できる形状の基本単位は平板，球，円板，円筒，円錐に限るため，15種類のプリミティブをこれらの基本単位に分解あるいは近似する処理を熱解析モジュールで行う。

ところで，熱解析では図7.15に示すように二つの物体が接合していて，物体AからB，あるいは物体BからAへ熱が伝わる場合，接触面で熱的な抵抗がある。本モデリング手法では，プリミティブの単純な接合で形状の定義を行っているため，複数物体の接触面における熱伝達の問題を取扱いやすい特徴がある。

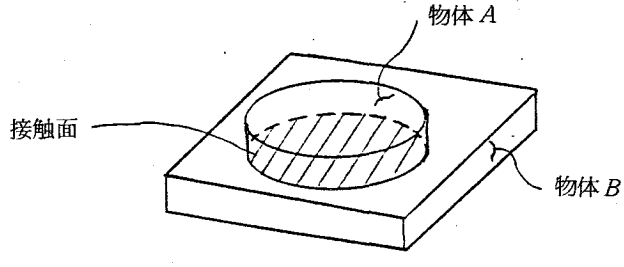


図7.15　接触面における熱的な抵抗

7.5 適用例

まず，代表的な通信衛星の概要を説明するが，人工衛星の仕様は打ち上げロケットや寿命などに関する基本的な要求事項と宇宙実積のあるハードウェアの利用などいろいろの問題点を検討した結果，仕様が個々に決まるので，以下に示すのは参考データである。

現在の通信衛星には，スピン安定型が多い。図7.16にその概観を示す。衛星本体は直径約200cm高さ約250cmの円筒形で，底部のアポジモータから通信用アンテナまでの全高は約350cmである。打上げ時の総重量は約1000kgである。衛星構体は，打上げ時における主エンジンカットオフにおいて，縦方向に約9Gの荷重がかかるので，これに耐える強度と剛性が必要である。しかし，総重量には制限があるため，衛星構体重量は軽くし，長期使用による静止衛星軌道からのずれを補正する推進系燃料を多くして，衛星寿命を長くしなければならない。したがって衛星構体の設計は難しい。また，衛星に搭載する機器の数は数百個以上もあり，それらを衛星内部に配置するための検討も極めて難しい。

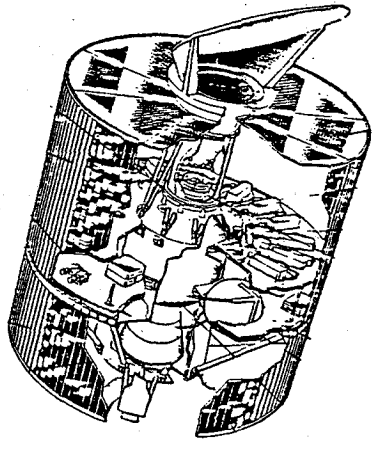


図7.16　代表的な通信衛星の概観

図7.17は人工衛星の設計プロセスの概要である。基本設計・詳細設計においては，熱構造モデル・エンジニアリングモデルなどのモックアップを作成し，種々の技術的な検討を行なう。これは，人工衛星内部に数多くの搭載機器を立体的に配置する必要があり，かつ，配置により質量特性や熱特性が変わるためである。しかし，モックアップの作成を繰り返し行なうのには限界がある。そのため，モックアップ作成による検討の一部を，計算機上の立体モデルを用いたシミュレーションにおきかえる。

まず，図7.18に示すスピン衛星を考える。スピン衛星の場合，回転により姿勢の安定を保っている。そこで，プレートと呼ばれる円板上に搭載機器を配置するにあたって本システムの干渉チェ

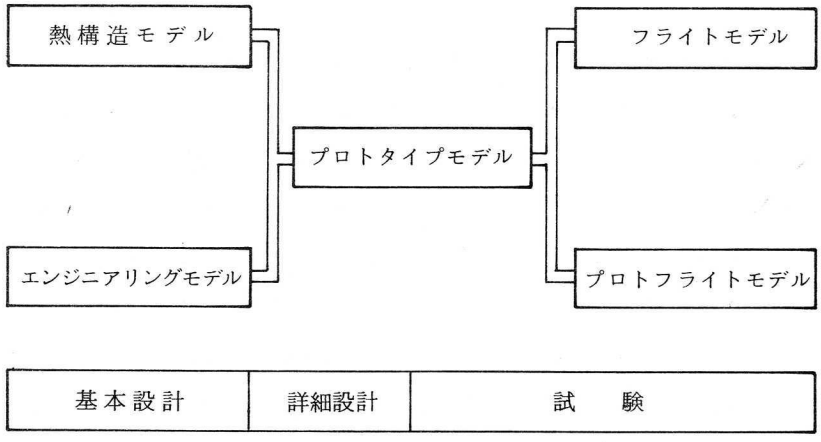


図7.17　人工衛星の設計プロセス

ック機能（図7.1参照）により配置可能性を確かめる（図7.19）と同時に質量特性解析モジュール（図7.1参照）をもちいて質量特性を調べる（図7.20）。ここで全COGは重心の位置ベクトル，MOIは慣性モーメント（kg cm²），POIは慣性乗積（kg cm²）である。さらに，熱特性解析モジュール（図7.1参照）を用いて熱特性の解析も行ない問題があれば，配置変更をおこなう。この，配置→干渉チェック→質量特性の確認→熱特性の確認→再配置のループを繰り返すことで，搭載機器の最適な配置を求める。この場合，衛星の構造体も解析の対象である。図7.21は，三軸衛星である。この場合にも，スピン衛星の設計と同じである。熱特性解析例を図7.22に示す。図7.17における，熱構造モデルのモックアップに相当する。

このように，計算機で立体モデルを扱い，種々の技術的な検討を計算機で行なうことで，全てモックアップにより検討を行なう従来方式に比べ，開発期間の短縮ができ，さらに，より多くの設計例の検討が可能となり性能向上も図れる。

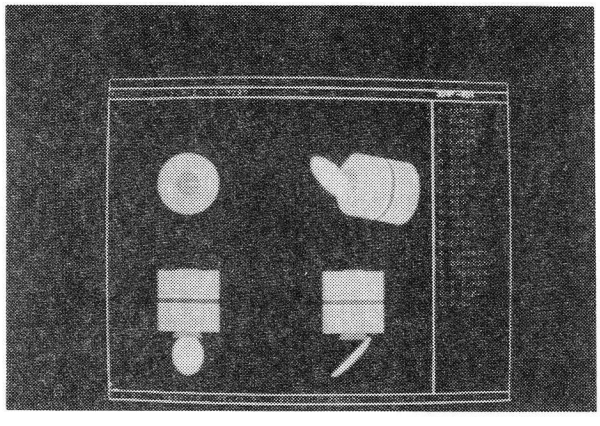


図7.18　スピン衛星

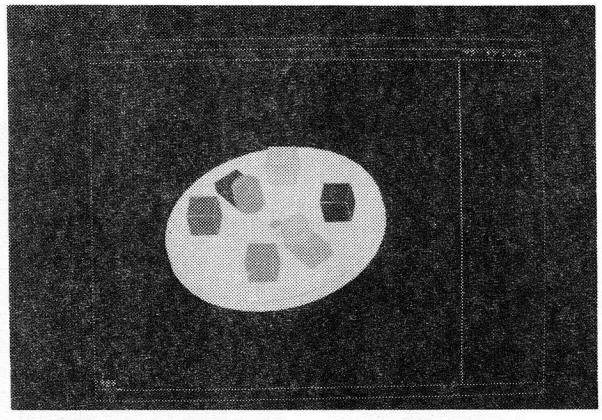


図7.19　プレート上での搭載機器の配置設計例

```
( DEBUG WRITE (1)ST ( INPUT DATA) )
   *** COMPONENT ATRIBUTE TABLE GIOMETRICAL DATA ***

   TOTAL  MASS -     3.000

  60.000      .000      .000      .000      .000
    .000     1.000    75.000   120.000    60.000
  30.000    30.000    30.000      .000      .000
   2.000   450.000   450.000   450.000      .000
    .000      .000     3.000    -1.000      .000
    .000      .000      .000      .000      .000

( DEBUG WRITE (2)ST ( RESULT )    )
   ** RESULT ( 1.COG / 2.MOI / 3.POI ) ABOUT COMPONET **

/ M 1-    3.00 / M 2-     .00 / M 3-     .00 / M 4-     .00 / M 5-     .00
/ M 6-     .00 / M 7-     .00 / M 8-     .00 / M 9-     .00 / M10-     .00
/ M11-     .00 / M12-     .00 / M13-     .00 / M14-     .00 / M15-     .00
/ M16-     .00 / M17-     .00 / M18-     .00 / M19-     .00 / M20-     .00
/ M21-     .00 / M22-     .00 / M23-     .00 / M24-     .00 / M25-     .00
/ M26-     .00 / M27-     .00 / M28-     .00 / M29-     .00 / M30-     .00

          (1) COG           (2) MOI           (3) POI

      750000.E-04       450000.E-03          0.E+00
      120000.E-03       450000.E-03          0.E+00
      600000.E-04       450000.E-03          0.E+00
```

図7.20　質量特性解析例

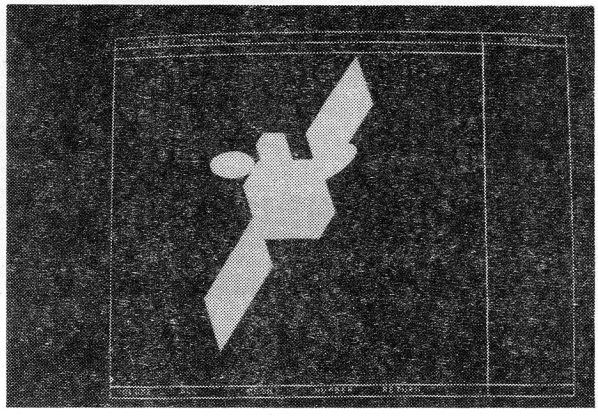


図7.2 1　三軸衛星定義例

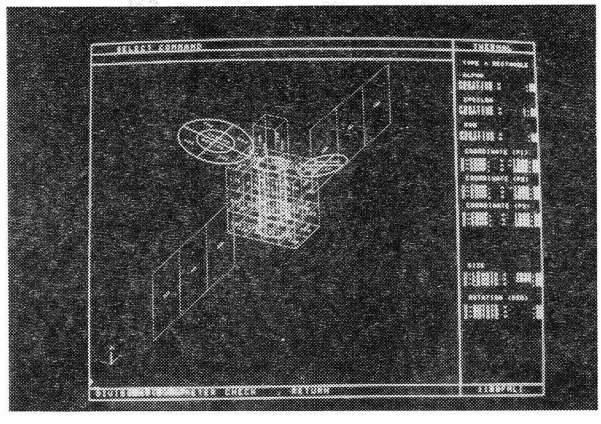


図7.2 2　熱特性解析例

7.6 結 言

本章で示したシステムは，既に実際の人工衛星の設計に活用されている。システムの実用化によって，人工衛星用機器の配置の検討が簡単になり，作業量は半分程度に軽減される効果が確認されている。本章で得られたことの要約を以下に示す。

(1) ここで示したモデリング手法は，ＣＳＧ，Ｂ－Repsなどの手法と比べるとサブセットではあるが，人工衛星形状の分析に基づいたプリミティブを設定したため，プリミティブの単純な接合によって形状創成ができ，かつ，各種形状処理も高速に行なえる。

(2) 熱特性解析，質量特性解析などの技術計算モジュールで必要な入力データは３次元ＣＡＤデータベースからほぼ直接抽出できる。

(3) 図形処理は，ディブスバッファを持つディスプレイによるハードウェアで行う方法にしたため，形状のデータ構造は単純であり，動画表示に必要な高速図形処理が可能なシステムを実現することができた。

今後の課題として，既に蓄積された設計技術を有効活用できるエンジニアリングデータベースの開発がある。人工衛星用３次元ＣＡＤシステムとエンジニアリングデータベースとの統合により，さらに総合的な設計支援が可能となることが期待される。

参 考 文 献

1） A. A. G. Requicha and H. B. Voelcker : Solid Modeling, A Historical Summary and Contemporary Assesment, IEEE Computer Graphics and Applications, 2.2 (1982) 9.

2） 木村文彦 ： ＦＡにおける情報処理技術の役割，情報処理，25.4 (1984) 283.

3） 塩谷景一，倉島徳幸，鈴木庸彦 ： 人工衛星用３次元ＣＡＤシステムの開発，投稿中（情報処理）.

4） B. G. Baumgart : Geometric modeling for Computer Vision, Rep. STAN－CS－74－463, Stanford Artificial Intelligence Lab., Stanford Univ., Stanford, Calif. (1974).

5） A. Baer, C. Eastman and M. Henson : Geometric Modeling －A Survey, Computer Aided Design, 11. 5 (1979) 253.

6） W. Tiller : Rational B－spline for Curve and Surface Representation, IEEE Computer Graphics and Applications, 3. 6 (1983) 61.

第8章　総　　括

本論文では，工業製品の設計や生産過程における形状操作に基づいた曲面や立体のモデリング手法について，種々の提案を行うとともに，これらの手法を用いたシステムの開発例を示した。大きく分けて，自由曲面を外形にもつ製品と機能によって形状が決まる製品の二つを対象にした。

本論文の特徴は，形状創成プロセスと設計・生産プロセスとの間を密接に結びつける形状モデリング手法によって，実用的なＣＡＤ／ＣＡＭシステムが構築できることを示したことであるが，得られた結果を要約すると次のようになる。

第２章では，自由曲面をもつ製品の造形過程で必要な曲面形状の形状制御の方法について提案を行なった。まず，ノットの間隔が１であるＢ－spline 曲線について，接線ベクトルや曲線の通過点などの幾何学的特徴をよく表す量を，陽に扱える形式の新しい式を導いた。次に，初期曲面の創成を容易に行なうため，局所的制御と大域的制御を併用する方法を示した。局所的制御については，従来から使われている制御頂点の操作を用いる。大域的制御は，アウトラインの曲面上の位置（入力位置）を変える手法と，新たに導入した形状制御関数により，特定のアウトラインの曲面形状への寄与を強める手法によって行えることを示した。この手法を用いると，ＣＲＴディスプレイを用いた対話処理によって曲面の創成を行うシステムの開発が容易であることを示した。

第３章では，図面などによって要所要所の断面形状が指定される場合の自由曲面の表現式について検討を行った。この場合に問題となる円弧のスプライン式による表現は，有理式 B－spline を用いると厳密な定義ができるが，同次座標値や，等間隔パラメータに対応する曲線上の点の間隔が不規則であることを示した。そのため，３次の Bézier 式で円弧を近似する場合の制御頂点の計算式を示した。この式によれば，中心角40°の円弧で，最大誤差は $0.95\times10^{-4}\%$ であるため，実用上十分な精度であることが分った。次に Brown の重み関数を用いて，一般Coons 曲面に含まれる二つのロフト曲面の凸結合を行い，曲面を創成する方法を示した。この表現式を用いると，複数セグメントを持つ自由曲面の創成が可能となり，さらに応用上，三角形曲面や五角形曲面の定義ができることを示した。

第４章では，曲面を扱うシステムへの入力データが点群の場合について検討し，任意に分布した点群を近似的に等価な格子状の点群におきかえて曲面を創成する一手法を示した。提案した手法は，点群を補間する空間曲線を創成し，平面あるいは線分との交点を計算するプロセスの繰返しである。そのため，13 MIPS の計算機で 600 個の任意分布点群から１３×１３の格子点を推定するための計算時間は４～７秒であり，実用上十分速い処理速度であることを示した。また，点群を補間する曲線式として，種々の理論式を用いることが可能なため，補間に使用する式を変えることによって，許容される誤差を満足し，かつ，短い計算時間で格子状の点群を求める手法が選択できること

を示した。

第5章では，機能的な要求によって決まる立体をモデリングする手法について検討した。製品形状の分析に基づき，形状を適当に平面によって切断し，それぞれの部分について適当な局所座標系を用意すると，$x\,y$平面上の一価関数によって部分形状の上界が定義されることを見い出し，この部分形状を立体セグメントとして積極的に取扱える手法を開発した。この手法は，任意の一点が形状の内部にあるか外部にあるかの判定を，2次元処理と1次元処理で行えるため，各種形状処理における計算量の軽減ができる。また，形状モデリングに際して用いるオペレーションは，形状として反映したい面を基礎としており，加工など生産における形状操作の概念に近い。つまり，従来のモデリング手法と比較するとサブセットではあるが，応用を考えた場合有効である。

第6章では，モールド金型などの製作に利用する金型用3次元ＣＡＤ／ＣＡＭシステムの開発例を示した。形状定義から加工までプログラムできる専用言語を開発し，ＡＰＴ系の言語で不明確であった変数の有効範囲の問題等を改善した。また，曲面を構成する曲線間の相互関係に基づき，曲面創成法を１５種類に分類した。この曲面創成法により直接定義できる曲面要素を接続したり，重ね合せることにより複雑な曲面の定義を容易にできることを示した。

第7章では，人工衛星用の3次元ＣＡＤシステムの開発例を示した。人工衛星の設計で頻出する形状をプリミティブに設定したため，プリミティブの単純な接合で形状定義ができる手法となっている。隠面処理などの図形処理はディプスバッファを持つグラフィックディスプレイによるハードウエア対応としたため，形状のデータ構造は単純である。汎用のモデリング手法と比べるとサブセットを扱っているに過ぎないが，人工衛星への適用では有効であることが確められた。

謝　　辞

本研究の遂行にあたり，終始御指導と御鞭撻を賜わりました大阪大学牧之内三郎教授ならびに，井川直哉教授，赤木新介教授に深甚の謝意を表すとともに，有益な御教示を賜わりました大阪大学川辺秀昭教授，山田朝治教授，梅野正隆教授，森勇蔵教授，岸田敬三教授に厚く感謝の意を表します。またとくに，モデリングの基礎技術については大阪大学田村坦之助教授，山縣敬一講師から多くの御助言を頂きました。

最後に本論文の作成に当って御世話になりました三菱電機株式会社生産技術研究所渡辺光人部長，川嶋良和グループマネージャーならびに同社名古屋製作所，大船製作所，鎌倉製作所の各位に厚く御礼を申し上げます。