(昭和 60 年 1 月 25 日受理)

三次元CAD/CAMのための衣服型紙と その自動有限要素分割

繊維高分子材料研究所 岡部秀彦・今岡春樹・赤見 仁

PAPER PATTERNS OF DRESS FOR 3-DIMENSIONAL CAD/CAM AND THEIR AUTOMATICAL DIVISION INTO FINITE ELEMENTS

By Hidehiko Okabe, Haruki Imaoka and Hitoshi Akami(Research Institute for Polymers and Textiles,
Yatabe-cho, Tsukuba-gun, Ibaraki-ken, 305, Japan)

We designed a data structure for paper pattern that enables to describe how the fragments or parts of a dress are sewed together and how the dress is put on the body. Such information is required in our system for 3-dimensional CAD for dress. In this system, we input the shape of each fragment, assign the lines to be sewn and mark some lines which are to put at some characteristic position of the body (e.g., waist line). Then the computer divides the fragments into finite elements, establishes the topology of the dress by referring the sewing information, estimates the rough position of each element (i.e., put on the body), calculates the form and distortion of the dress from mechanical characteristics of the cloth and body shape data, and displays the solid view of the result on 3-dimensional graphic display. The fact that our CAD system can simulate the formation of the final shape of the dress, assures that the information given in our paper pattern suffices to indicate how the parts should be assembled and sewn in CAM or automated sewing system. In this paper we describe the data structure of the paper pattern and the automatical division of it into finite elements.

A pattern of a dress usually consists of some pieces of cloth which are to be sewn together, and each piece has rather complicated boundary curves. Even though the lines which will be sewn together have different shapes and lengths, our program generates the points correctly corresponding with each other on every line. Moreover, when the shape of a line is complicated, the pitch of division is automatically reduced so that the line defined by the calculated dividing points well approximates the original line. Then every fragment is divided into a collection of triangles (formally, into a two dimensional simplicial complex).

Our program can divide an arbitrary pattern and can produce well formed triangles, even if there is no additional information for division instructed by the user. Hence, it is suited for the use by those who are not expert with the mechanical analysis by finite element method.

(Received January 25, 1985)

1. 緒 言

型紙は衣服製作のための設計図である。そしてまた型紙は恐らく量も大量に生産されかつ利用される設計図の一つであろう。従来、この型紙を電子計算機によって処理する際、それは単なる裁断片の集合として扱われ、そ

れらがどのように組合わせられ縫合されるかという、アセンブリングに関する情報は型紙上には部分的にマークされるのみで、後は言葉による指示と縫製作業者の常識に頼っていた。我々は型紙作成のCAD化の一つの要素として、与えられた型紙と、用いる布の力学特性及び着せられる人体の型状データの三者より、その型紙に基づい

て作られた衣服が人体に着用された際どのような形状を取りどのような応力・歪を示すかを有限要素法により数値シミュレーションするシステムを開発した^{1~4)}。このシステムの主要な目的は、現在パターンメイキングにおいて試作パターンを評価するのに実際にサンプル縫製を行なっているのを、計算機シミュレーションによって代行することにより、試作-評価のサイクルタイムを「日」のオーダーからせめて「時」のオーダーへと減少させることである。図1に示すように、このシステムでは型紙から始まって縫合、着付け、衣服の力学に従った形状形成までを一貫してシミュレートするものであるため、ここで「型紙」と呼ぶデータ構造体は単に各裁断片の形状だけでなく、製品へのアセンブリと人体への着用へのための情報を含んでいる。そしてこのような「型紙」

は将来CAMにおいて生じる要求を先取りしたものともなっている。なぜなら、この型紙が数値シミュレーションによって完成品の形状を算出するのに十分な情報を有しているのなら、計算機は手足を与えられればそれを実体においても同様に作る事ができる筈だからである。

本報告ではこのシステムにおける「型紙」のデータ構造と、それを自動的に有限要素の集合体へと変換する方法について述べる。このシステムは他に重要な要素として、i) 平面の裁断片の集合が縫合されることにより取る立体形状の概形を求める方法、ii) それを布地や縫合個所の力学特性及び体型データに従って最も安定な状態へと変形する力学解析シミュレーション、iii) 計算結果の表示やそれを参照しての最初の型紙の変更等を行なうためのグラフィックディスプレイによるマンマシンインターフェイス等を含んでいるが、

これらについては今後の報告に譲ることとする。

さて不特定の多様な対象を数値解析又はシミュレートすることを目的としたシステムにおいては、解析の対象をプログラムが要求するデータ構造へと変換し入力する作業がいかにかつ迅速に行なえるかが実用上の大きな問題となる。要素データ作成の自動化・効率化の必要性は有限要素法が汎用システムの形で使われ出すにつれ認識され、既に多くの研究と実用化の歴史を持っており、そこで用いられている手法もまた多様である^{5~17)}。本報告で述べる方法の特徴は次の三点である。i) 任意の平面領域を分割できること。ii) 分割に際して全く人間の介在が不要であること。iii) 領域の周の分割も自動化されており、形状・長さの異なる曲線を縫合指定しても整合性(夫々の曲線の分割点が縫合部において互い違いにならないこと)が保証されること。このような特徴を有するに至ったのは次のような理由がある。i) 型紙は一般に多くの自由曲線を含むため、定まった種類の図形プリミティブの結合(和、差)として表現するには不向きでありまた異なる形

Fig. 1 The general flowchart of the three dimensional pattern CAD system. The parts drawn by broken lines are in planning.

状と長さを持つ複数個の自由曲線を縫合した際各曲線上の分割点がうまく整合して一致するようにそれらを定めるのは入手では困難である。ii) ユーザーである衣服のデザイナーは一般には構造力学や有限要素法についての専門的知識を有しない。iii) 計算結果として必要とされるのは主に衣服のとり立形状の概略であり、高い精度は要求されない。従って計算精度向上のための分割の最適化に対する要求度は比較的低い。しかしながら我々はこのような状況は衣服型紙という比較的特殊な分野に限らず、有限要素解析の広汎な応用分野において生起するものであろうと考えている。

2. データ構造

自動有限要素分割プログラムの役割は自由曲線の集合体として定義される型紙を、縫合部整合性等の諸条件を満たしつつ、平面上の二次元単体的複体^{18,19)}として定義される分割済型紙に変換することである。ここで両者のデータ構造について述べる前に、その記述に用いる略記法について簡単に説明する。

$\langle A \rangle = \langle B \rangle + \langle C \rangle$ は A というデータ構造又は“型(データタイプ)”が、B 及び C というデータ構造の二つ組(B, C)であることを示す。 $\langle A \rangle^n$ (n は自然数) はタイプ A のデータ n 個から成る列を、 $\langle A \rangle^*$ はタイプ A のデータの有限個(0 個も含む)の列を意味する。 $\langle A \rangle | \langle B \rangle$ はタイプ A 又はタイプ B のデータを示す。演算子は $n, *, |, +$ の順で結合優先度が高く、また演算子の有効範囲を示すために() を用いるものとする。即ち、ここでは構文記述のための BNF 記法²⁰⁾を多少変更・拡張したものをデータ構造の略記法として用いる。

2.1 型紙のデータ構造

$\langle \text{型紙} \rangle = \langle \text{付記} \rangle + \langle \text{裁断片} \rangle^* + \langle \text{縫合指示} \rangle^* + \langle \text{位置指示} \rangle^* + \langle \text{ライン} \rangle^* + \langle \text{座標点} \rangle^*$
 $\langle \text{裁断片} \rangle = \langle \text{付記} \rangle + \langle \text{周ラインポインタ} \rangle^* + \langle \text{ラインポインタ} \rangle^* + \langle \text{座標点ポインタ} \rangle^*$
 $\langle \text{縫合指示} \rangle = \langle \text{縫合仕様番号} \rangle + (\langle \text{縫合パラメータ} \rangle + \langle \text{要素ポインタ} \rangle)^*$
 $\langle \text{位置指示} \rangle = \langle \text{位置づけ仕様番号} \rangle + (\langle \text{位置指示パラメータ} \rangle + \langle \text{要素ポインタ} \rangle)^*$
 $\langle \text{要素ポインタ} \rangle = \langle \text{ラインポインタ} \rangle | \langle \text{座標点ポインタ} \rangle$
 $\langle \text{ライン} \rangle = \langle \text{ライン属性} \rangle + \langle \text{座標点ポインタ} \rangle^*$
 $\langle \text{座標点} \rangle = \langle \text{座標点属性} \rangle + \langle \text{XY 座標} \rangle$

付記は型紙や裁断片の名前、素材の指定、管理情報等のためのフィールドである。仕様番号は縫合や着用位置指示の際、どのような手続き(例えば二枚縫合や三枚縫

合等)を用いるかを指示し、各パラメータはそれに続く要素(ライン又は座標点)がその手続きの中でどのように扱われるか(例えば縫合の方向)等を指定する。属性はその要素の表示特性等を指示する。XY 座標系は一つの裁断片の内でのみ共通であり、異なる裁断片に属する二つの座標点の XY 座標間には何の関係もない。ラインは縫合指示や着用位置指示における最小単位であり、例えば一つのラインの中間点から始まるような縫合を定義することはできない。

また、ラインは一つの自由曲線に対応するため、角度を持って折れる線の角の個所ではラインを切り替えなければ、その個所は後続する処理で、丸みづけられてしまう。自動分割において、ラインの始点と終点は必ず分割後の頂点となるが、ラインの中間の座標点についてはそのような保障は全くない。一つの裁断片の周をなすラインはしばしば利用されるため、各裁断片について特別にリストを作っている。

2.2 分割済型紙のデータ構造

$\langle \text{分割済型紙} \rangle = \langle \text{付記} \rangle + \langle \text{裁断片} \rangle^* + \langle \text{縫合指示} \rangle^* + \langle \text{位置指示} \rangle^* + \langle \text{三角形} \rangle^* + \langle \text{辺} \rangle^* + \langle \text{頂点} \rangle^*$
 $\langle \text{裁断片} \rangle = \langle \text{付記} \rangle + \langle \text{周周辺ポインタ} \rangle^* + \langle \text{三角形ポインタ} \rangle^* + \langle \text{辺ポインタ} \rangle^* + \langle \text{頂点ポインタ} \rangle^*$
 $\langle \text{縫合指示} \rangle = \langle \text{縫合仕様番号} \rangle + (\langle \text{縫合パラメータ} \rangle + \langle \text{要素ポインタ} \rangle)^*$
 $\langle \text{位置指示} \rangle = \langle \text{位置づけ仕様番号} \rangle + (\langle \text{位置指示パラメータ} \rangle + \langle \text{要素ポインタ} \rangle)^*$
 $\langle \text{要素ポインタ} \rangle = \langle \text{辺ポインタ} \rangle | \langle \text{頂点ポインタ} \rangle$
 $\langle \text{三角形} \rangle = \langle \text{三角形属性} \rangle + \langle \text{裁断片ポインタ} \rangle + \langle \text{辺ポインタ} \rangle^3 + \langle \text{頂点ポインタ} \rangle^3$
 $\langle \text{辺} \rangle = \langle \text{辺属性} \rangle + \langle \text{裁断片ポインタ} \rangle + \langle \text{頂点ポインタ} \rangle^2 + \langle \text{三角形ポインタ} \rangle^2$
 $\langle \text{頂点} \rangle = \langle \text{頂点属性} \rangle + \langle \text{裁断片ポインタ} \rangle + \langle \text{XY 座標} \rangle + \langle \text{辺ポインタ} \rangle^*$

上にあげたデータ構造の記述は極めて概念的なものであり、実際にこれらを計算機で実現する方法は一意的ではないし、有限個列“*”が可変長(最初長さが決まれば、以後は変化しない)を意味するか、不定長(プログラムの実行にともなって長さが増減する)を要求するかによっても大きく異なってくることに留意する必要がある。

3. 自動分割のアルゴリズム

我々の自動分割アルゴリズムは大きく二つのステップに分かれる。第一のステップでは、型紙内の縫合ライン

をすべて整合的に線分に分割し、さらに他のラインもすべて線分に分割する。またこの分割に従ってパターンにおける縫合指定、位置指定を分割済型紙におけるそれらの形に変換する。第二のステップではパターンの各裁断片ごとに、裁断片の内部領域を三角形に分割しながら、分割済型紙のデータとトポロジーを構成してゆく。

3.1 ラインの分割

ここで用いた分割法の原則は、自由曲線の入力に際して、通過座標点の指示が密である個所は分割も密に行なう、というものである。一般には通過座標点の指示が密であるということは、曲線の形状が細かく変化していることを示しているが、曲線が単純な場合でも入力の際に応力集中を予測して座標点指示を細かくするという使い方も可能である。これに加えて、分割の基準長(ルール)を越えないことと、縫合線分割の場合、整合的に分割するという条件を満たすことが必要である。我々が用いた方法を以下に示す。

I. p 本のラインが縫合されるとして(縫合されないラインの時は $p=1$) q 番目のラインは座標点列 $x_0^q, x_1^q, \dots, x_{n_q}^q$ ($1 \leq q \leq p$) で定められているものとする。但しこの点列の順は縫合されるラインすべてに共通な縫合方向にそろえられているものとする。これはデータ構造内でのラインを定義する座標点列の順がこれと逆になっている可能性があるので、縫合パラメータの指示に従って順序づけの方向をそろえる必要があることによる。

II. 各ラインを通過座標点列で定められる折れ線と考え、それに沿って始点が0、終点が1の値を取る、折れ線の長さに比例して増加するパラメータ t へのライン上の点からの写像 φ^q を次のように定める(図2)。

$$l_i^q = |x_i^q - x_{i-1}^q|, L^q = \sum_{i=1}^{n_q} l_i^q, t_i^q = \sum_{j=1}^i l_j^q / L^q \text{ とし}$$

て線分 $x_{i-1}^q x_i^q$ 上の点 $x = (\alpha-1)x_{i-1}^q + \alpha x_i^q$ に対して $\varphi^q(x) = (\alpha-1)t_{i-1}^q + \alpha t_i^q$ 。

III. t 上の関数 $\phi^q(t)$, ($0 \leq t \leq 1$, $1 \leq q \leq p$) を $\phi^q(t) = \begin{cases} 1/(t_i^q - t_{i-1}^q) = L^q/l_i^q & \text{for } t_{i-1}^q \leq t < t_i^q \\ 0 & \text{for } t = 1 \end{cases}$

で定める(図3)。これは q 番目のラインでの通過座標点指示の密度である。

IV. ライン分割の許容最大長(ルール)を λ_M , 許容最小長を λ_m とする ($0 < \lambda_m < \lambda_M$)。

我々は分割密度関数を ($L_m = \min\{L^q\}$, $L_M = \max\{L^q\}$)

として $\phi(t) = \min\{\frac{L_m}{\lambda_m}, \max\{\frac{L_m}{\lambda_M}, \phi^1(t), \phi^2(t), \dots, \phi^p(t)\}\}$ ($0 \leq t \leq 1$) で定める。但しこのままでは分割数 $W = \int_0^1 \phi(t) dt$ が整数とならないため、補正分割密度関数を $\tilde{\phi}(t) = \phi(t) \cdot \tilde{W}/W$ (但し $\tilde{W}$ は $W \leq \tilde{W}$ と

Fig. 2 The mappings onto the common coordinate from two lines which are to be sewn together. Lines $x_0^1, \dots, x_4^1$ and $x_0^2, \dots, x_5^2$ which are to be sewn together are mapped onto the common coordinate t ($0 \leq t \leq 1$) by φ^1 and φ^2 . $\varphi^1(x)$ and $\varphi^2(x)$ are linear functions of the arc-length of each line.

Fig. 3 Calculation of the division density function from the densities of the passing points of the two lines (Fig. 2). ϕ^1 (dotted line) and ϕ^2 (solid line) are the densities of the assigned passing points of the lines in Fig. 2. Then the division density function ϕ is defined by $\phi(t) = \min\{L_M/\lambda_m, \max\{L_m/\lambda_m, \phi^1(t), \phi^2(t)\}\}$ ($0 \leq t \leq 1$), where L_M/λ_m (resp. L_m/λ_m) denotes the admissible maximum (resp. minimum) density of dividing points in t .

なる最小の整数)で定義する。 $T = \bigcup_{q=1}^p \{t_i^q \mid 0 \leq i \leq n_q\}$

の元を大きさの順にソートして、 $T = \{\tilde{t}_j \mid 0 \leq j \leq N, 0 = \tilde{t}_0 < \tilde{t}_1 < \tilde{t}_2 < \dots < \tilde{t}_{N-1} < \tilde{t}_N = 1\}$ とすれば、 $\tilde{\phi}(t)$

Fig. 4 The determination of dividing points on the common coordinate. To get an integral division, we correct ϕ by an appropriate factor and obtain $\tilde{\phi}$. Then the dividing points $\tilde{t}_0, \tilde{t}_1, \dots, \tilde{t}_8$ are defined to fulfill the conditions,

$$\begin{aligned} \int_{\tilde{t}_0}^{\tilde{t}_1} \tilde{\phi}(t) dt &= \int_{\tilde{t}_1}^{\tilde{t}_2} \tilde{\phi}(t) dt \\ &= \dots = \int_{\tilde{t}_7}^{\tilde{t}_8} \tilde{\phi}(t) dt = 1. \end{aligned}$$

は区間 $[\tilde{t}_{j-1}, \tilde{t}_j]$ では夫々一定の正の値をとるステップ状の関数である (図 4)。

V. $\Psi(t) = \int_0^t \tilde{\phi}(t') dt'$ とすれば、 $\Psi: [0, 1] \rightarrow [0, \tilde{W}]$ は単調増加する全射である。ここで我々は整数値 $0, 1, \dots, \tilde{W}$ の Ψ についての逆像 $\tilde{t}_0 = \Psi^{-1}(0)$, $\tilde{t}_1 = \Psi^{-1}(1)$, $\dots, \tilde{t}_{\tilde{W}} = \Psi^{-1}(\tilde{W})$ によって共通パラメータ t での分割点を定める。

VI. $\varphi^q(x)$ も各 q についてライン上で単調増加する関数であるから逆関数を定義でき、各ラインでの分割点は $\tilde{x}_r^q = (\varphi^q)^{-1}(\tilde{t}_r)$ ($0 \leq r \leq \tilde{W}$, $1 \leq q \leq p$) で与えられる (図 5)。

例えば、両端だけで定義された長さ L の線分は L/λ_M を整数に切り上げた数に等分割される。また通過座標点の間隔がすべて λ_m と λ_M の間にある自由曲線の場合には、通過座標点そのまま分割とされることがわかる。

この方法によれば、長さの異なる曲線が縫合される際には、両者が同じ長さになるように一様な伸縮を加えた上で整合的に分割した後、夫々の曲線をもとの長さへ戻してやることになるが、この効果は実際の縫製において、縫合個所の両端をあわせてピンと張りながらミシンをかけていくやり方によく対応している。いせ込みに見られるように、縫合個所での双方の縫合長の差が段階的に変化している場合には、ラインを次々に切り替えることにより、これを表現できることになる。

現在のところ簡単のためラインは折れ線であるとして、 $(\varphi^q)^{-1}$ を区間線型関数としているが、これをより滑らかな曲線近似に変更しても上記の方法の本筋が変わらない

Fig. 5 Mutually adjusted dividing points on the two lines. The dividing points of each line are determined by the inverse images of the dividing points on t ($\tilde{t}_0, \dots, \tilde{t}_8$) of the mappings φ^1 and φ^2

ことは明らかであろう。

プログラムにおいてはライン分割に伴って、分割済型紙のデータ構造に生成された辺と頂点を裁断片毎にソートして登録し、縫合指示や位置指示等も辺に分割した形で登録する等の作業を行なうが、アルゴリズムとして特筆すべき点はないので言及しない。

3.2 領域の三角形分割

分割の手続きにおいて必要とされる平面図形の位相的な性質についてまず説明する。

平面内の辺は二つの“側”を持つ。本文では辺の始点から終点に向かって左を左側、右を右側と言うことにする。辺 $(x_1, y_1) \rightarrow (x_2, y_2)$ に対して点 (x, y) は $S = (x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1)$ が正の時左側、負の時右側に位置する。領域分割の終了はすべての辺の両側が三角形によって占められることにより定義される。但し裁断片の周を成す辺については、最初からその右側が外界により占められていると解釈しておく。このために、入力の際には裁断片の周を成すラインは必ずその内部を進行方向に対して左手に見る方向で行なうと定める。即ち領域の外周は反時計回りに、内部の穴の周囲は時計回りに入力を行なう。またこの時ある点 (x, y) がこの領域に含まれるか否かは、周をなす辺を $(x_{i1}, y_{i1}) \rightarrow (x_{i2}, y_{i2})$ ($i = 1, 2, \dots, n$) として、それらが点に対して張る (符号付の) 角度の和 T によって次のように判定できる。

$$\begin{aligned} T &= \sum_{i=1}^n \left\{ \text{Arctan} \frac{(x_1 - x)(y_2 - y) - (y_1 - y)(x_2 - x)}{(x_1 - x)(x_2 - x) + (y_1 - y)(y_2 - y)} \right\} \\ &= \begin{cases} 2\pi & \text{内部} \\ 0 & \text{外部} \end{cases} \end{aligned}$$

我々の分割アルゴリズムは領域が連結していなくてもよいし、また各連結成分が単連結¹⁸⁾である必要もない。上

記の判定法についても、この点は同じである。

さて領域の分割は以下のステップにより行なわれる。

I. 領域の周をなす辺を定める各頂点の座標の最大・最小値から領域を包含する長方形を求め、それを覆う三角格子の範囲を定める。但し三角格子は $(0, 0)$, $(a, 0)$, $(a/2, \sqrt{3}a/2)$ なる正三角形で定められるものとする。 a は分割の基準長で我々は $a = \lambda_M$ としている。

II. 上で求めた格子点の中で、i) 領域内にあり、ii) 既存の辺に対して張る角の最大値が $2\pi/3$ 未満であり、iii) 既存の頂点との距離の最小値が $a/2$ 以上であるものを頂点として採用する(図6)。このステップで頂点の生成は終了する。

III. 各辺の空いている側について以下の操作を行なう。i) その辺のその側にある(除外されていない)頂点すべての中でその辺に対して張る角度が最大であるものを取る。ii) その頂点と辺が作る三角形の各辺が既存の辺と交差しなさをチェックし、それを通過すればiii)へ、しなければその頂点を除外してi)へ戻る。iii) 採用された三角形の各辺が既存の辺でない時には、その辺を登録する。また各辺の該当する側が、今採用された三角形によって占められていることを登録する(図7)。

このステップでは各辺の両側を順に処理するに従って新たな辺が生成されてくるので、辺の集合を一つのキューと考えて処理して行き、キューが空になったら終了すればよい。

IV. ステップIIで生成された頂点についてのみ、その座標をそれに辺で直接結ばれている頂点(隣接頂点)の座標の平均値へと移動する。但し、この結果その頂点を要素とする三角形に裏返してしまうものがある場合には移動は行なわない。この操作を我々の場合二回反復することにより、極度に歪んだ分割要素の修正をはかっている。

各辺に対してそれを見込む角が最大となる(非交差性を満たす)頂点を選ぶことにより、この三角形の内部に別の(非交差性を満たす)頂点が存在する可能性が排除されていることに留意すれば、以上の手続きが任意の正しく定義された領域と分割基準長の組に対して、正しい単体分割を与えることを証明することもできるがここでは省略する。むしろ実際に問題となるのは丸め等の計算誤差に起因する分割のエラーである。例えば二つの線分のクロスチェックの際、線分AとBのチェックと線分BとAのチェックが異なった結果となることは計算機処理においては起こり得ることである。我々のプログラムではこのような誤差を避け、また仮に誤差が生じても最終的には単体の複体として辻褄の合った結果を与えるよう種々の配慮をしている。例えば点が領域の内にあるか外にあ

Fig. 6 The determination of the nodes of triangular dividing mesh. The points calculated by the line dividing step (■), and the interior lattice points which are not too close to the lines (●) are adopted as the nodes of the two dimensional simplicial complex.

Fig. 7 The determination of the interconnection structure (=the internal edges) of the mesh. For every open side of each edge, we find the node such that the angle facing to the edge is maximum, and registrate the triangle formed by the edge and the node to the two dimensional simplicial complex.

るかを判別するのにその点から無限遠へ引いた半直線が領域の周と交わる回数の偶奇によって判定する方法を取っていないこと等はそれにあたる。

4. 実 例

図8にここで述べた方法で分割した4枚接ぎミニスカートの型紙の二つの裁断片を示す。入力に際して曲線部はこの時の許容最大長 $\lambda_M = 5\text{ cm}$ よりも細かい間隔で

Fig. 8 Two areas (parts of a pattern of a skirt) divided by our program. The lines facing between two areas are to be sewn together. Though they have different shapes and lengths, there is a proportional correspondence among the dividing points on these lines.

Fig. 9 The same area in Fig. 8 - below, divided under smaller λ_M and λ_m . As λ_M is smaller than the length of every interval of assigned passing points, the distribution of the nodes on each boundary line is uniform.

Fig. 10 An example of the division of a more complicated pattern for a skirt, actually used in an apparel factory.

入力され、直線部はその両端のみを入力されている。この結果、曲線部では分割が幾分か細くなり、そのために曲線が折線近似の劣化が防がれている。また二つの裁断片の縫合部の長さや形状は双方で異なっているが、分割点は両方を同じ長さに強制的に一樣に伸縮した際、うまく対応するように取られている。図9は図8の下の方の裁断片と全く同じ座標点列によって定義された型紙を $\lambda_M = 2\text{ cm}$ で分割した結果を示す。この時には入力した曲線部での入力座標点の間隔よりも λ_M の方が小さいため、周辺の分割はほぼ均等になっている。

図10はもう少し複雑な形状を有するスカートの型紙の例で、裁断片間の縫合だけでなく、腰のダーツの部分

に裁断片内の縫合を有している。

ここにあげた各例の計算はMELCOM 70/150 を中心としたシステムで行なわれた。我々のプログラムは最大5000個までの三角形への分割を扱うことができるもので、使用言語はFORTRANでステートメント数は約900、コードサイズは約34KB、使用する配列の総量は約1.2MBである。計算所要時間は図10に示した例の場合、生成された三角形の総数約1092個に対して14分を要した。但しこれはファイル入出力や分割の結果のハードコピーへの出力のための時間を含めた値である。

5. 考 察

詳しい報告は別の機会に譲るが、ここで述べた型紙のデータ構造とその自動分割により衣服が人体に着用した際とる形状を算出するシステムは現在実際に稼動中であり⁹⁾、それらが十分なものであることを立証している。緒言にも述べたように、この「型紙」はCAMに対してアセンブリングにおける製品側の情報として十分であり、現在自動縫製システムへの適用を検討している。但しここで述べた縫合・着用に関するデータ構造は、それに従ってデータが縫合・着用をシミュレート又は実行するプロシジャーにパラメータリストとして渡される際に十分に一般的な型式を有しているのであるが、このプロシジャー自身は様々の縫い合わせ方、折り合わせ方、着付方に対応して各々サブルーチンの形でシステムに組み込む必要があり、現在の所実際に組み込まれたものは夫々数種類にしか過ぎないので、今後これらを充実させて行く予定である。このような拡充はCAMとの接続においては特に問題はないが、CADのための力学シミュレーションにおいては新たに定義された縫合方式の部位の変形エネルギー計算式の確立をも要求するため、今後一層の実験が必要である。また縫合・着用の指定が複雑化するに応じて、それを扱うエディターも機能の向上が必要とされよう。現在我々はこれらの問題をCAMとの結合の実現とからめて検討中である。

3.1に述べたラインの辺への分割法は分割密度関数 $\phi(t)$ を選ぶことにより様々に応用が可能である。ここでは $\phi(t)$ を長さも形状もまちまちな縫合されるべき各曲線の通過点指定密度の最大値をとることにより、縫合される曲線の片方が直線であっても他方が複雑な形状を持つ場合、それに合わせて十分に細かい分割が与えられるようになっている。ここで最大分割密度 L_m/λ_m による $\phi(t)$ の頭切りは、過度に細かな分割に対する制限を与えているわけであるが、この方法は一般に曲線を点列として表現する際、適当な点密度に調整することによりデータ圧縮のために用いることができる。この方法はスプラインや非線型最小自乗法によるカーブフィッティングとは異質のものであり、むしろこれらと併用することにより、効率的かつ正確な曲線の表現を可能とするように思われる。我々のシステムにおいても、型紙の形状を入力する際ライン通過点の指定の細かさは大幅に変動するが、要素分割後の分割密度は適当なものとなっている。TVカメラから型紙を入力する際には、ピクセルの大きさのオーダーで極めて凹凸のある輪隔線が得られるが、これのスムージングとデータ圧縮にこの方法を用いる事を検討中である。 $\phi(t)$ としては $\Psi(t) = \int_0^t \phi(t') dt'$

とした時区間毎に Ψ^{-1} が陽に計算できるものであればよく、我々のシステムにおいて該当するサブルーチンを交換すれば、新たな $\phi(t)$ による分割を行なうことができる。例えば曲線が通過点例ではなく曲線の長さに沿ったパラメータ t のベクトル関数 $\mathbf{x}(t)$ として与えられている場合には、 $\phi(t)$ として曲率 $|\frac{d^2\mathbf{x}}{dt^2}|$ の適当な増加関数を用いればよい。

今の所、我々のシステムにおいては曲線を折線で近似しているため、点列で与えられたラインを分割して新たな点列によって表現すると、それらの点は常に元のラインが凹になっている側に取られてしまうという欠点を持っている。これを改善するためには元のラインを点列で定められるスプライン曲線と考えて、その上に新たな分割点を取ればよく、この時 t による表現や Ψ^{-1} が計算機にとって容易に計算できるようなスプライン曲線族を探すのも今後の課題の一つである。

領域内部の三角形分割法は本質的にShaw and Pitchen⁶⁾及び奥田⁶⁾の方法と同じものである。即ちi)領域内の格子点で辺に近接していないものをすべて内部頂点とする⁷⁾。ii)辺をベースに三角形を構成してゆく。その際の頂点選択基準は、作られる三角形の頂角の大きさである。iii)分割完了後、要素形状の調整のため、頂点の重心移動を繰り返す⁸⁾。本報告を含め、三者は全く独立に分割完了後の調整にまで同じ方法を採用しており、興味ある偶然の一致と言えよう。勿論細部においては差異があり、特に上記両者においては部分領域毎に分割密度を指定しなければならないのに対し、我々は緒言で述べた理由によりこの点をプログラムにまかせている。また我々のシステムでは領域周以外のラインや点を許容しているため、分割密度を高くしたい場所にあらかじめ点やラインを指定しておくことが可能である。一方我々のシステムでは複数の裁断片を結合することができるので、分割密度の異なる個所を夫々別の縫断片に分割することにより上記の二方法と同じに運用することも簡単である。我々の方法は、任意領域を分割できることや最小限の入力しか必要としないといった長所を、乱数法⁸⁾やベルトゾーン法^{9~11)}等と共有している。写像法^{12,13)}やゴムネット法¹⁴⁾、格子法^{15,16)}等は余分の入力が多量に必要であったり、分割要素の歪みが大きかったり、領域間が複雑な場合への適合が不十分であったりするため、緒言に述べた我々の状況においては不適當であるように思われる(ここでの分割法の分類及び呼称は横山¹⁷⁾におおむね従った)。

現在のところ、実際に大規模に用いられている分割法は分割対象が単純なプリミティブ図形の代数で表現され

ていることを前提としたものが多く、その場合計算量は比較的小さいが、対象図形が限定され入力の手間もかかる。我々の方法や乱数法、ベルトゾーン法等では自動分割のための計算量はかなり大きくなるが対象図形の制約はなく入力も最小限で済むため、電子計算機の高速大容量化とコストダウンに伴ってこのような方法が一般化する可能性は高い。有限要素分割における重要な問題は応力・変形が集中する部位ではメッシュを細かく、そうでない所は粗くすることにより計算精度を効率よく高くすることである。型紙の場合、人体の形状の曲率が大きい部分では応力が集中しやすく、一方平面図形を縫い合わせてそのような個所にフィットさせるにはダーツを入れる等型紙がその部分で複雑になる傾向があるため、その個所が自動的に比較的細かく分割される(図10)。また上に述べたように入力の際、応力集中が起りそうな個所の分割が細くなるように誘導してやることもできる。しかしながら応力集中個所が事前には明白でない場合には、変形解析の結果をフィードバックして再分割を行なうことが必要になるであろう。布のように柔らかい素材では応力が比較的小さくても変形は大きいため、幾何学的非線型性の問題からメッシュを間引くことは許されない。従って計算機的能力が許す限り、最初から分割を細かく取っておくだけでも布の場合現在の要求水準に対しては十分であると我々は判断しているが、この点については改めて本システムの力学シミュレーションに関する報告で述べる予定である。

文 献

- 1) 岡部秀彦, 赤見 仁; 織高研究報告, **142**, 63 (1984)

- 2) 今岡春樹, 岡部秀彦, 西川 茂, 渋谷惇夫, 赤見 仁; 織高研究報告, **142**, 73 (1984)
- 3) 別所康男, 赤見 仁; 織高研究報告, **142**, 81 (1984)
- 4) 岡部秀彦, 今岡春樹, 別所康男; 織高研究報告, **142**, 87 (1984)
- 5) R. D. Shaw and R. G. Pitchen; *Int. J. num. Meth. Engng.*, **2**, 93 (1978)
- 6) 奥田 敏; 精密機械, 「有限要素解析のためのパターン発生による任意平面領域の自動要素分割」, (投稿中)
- 7) S. Nordbeck and B. Rystedt; *BIT*, **7**, 39 (1967)
- 8) J. C. Cavendish; *Int. J. num. Meth. Engng.*, **8**, 679 (1974)
- 9) A. Bykat; *ibid.*, **10**, 1329 (1976)
- 10) E. A. Sadek; *ibid.*, **15**, 1913 (1980)
- 11) 城 道介, 秋元芳夫; 精密機械, **46**, No.3, 286 (1980)
- 12) O. C. Zienkiewicz and D. V. Phillops; *Int. J. num. Engng.*, **3**, 519 (1971)
- 13) 川面恵司, 中村正人, 荒木裕子, 南 久美子; 情報処理, **14**, No.5, 334 (1973)
- 14) 鈴木敏之, 鄭 泰亨, 會田俊夫, 藤尾博重, 久保愛三; 機論, **47**, C, 1366 (1981)
- 15) A. M. Winslow; *J. Comput. Phys.*, **2**, 149 (1967)
- 16) W. C. Thacker, A. Gonzalez and G. E. Putland; *ibid.*, **37**, 371 (1980)
- 17) 横山正明; 精密機械, **48**, No.12, 1664 (1982)
- 18) 田村一郎; 「トポロジー」(岩波全書276), 岩波, (1972)
- 19) 加藤十吉; 「組合せ位相幾何学」(岩波講座基礎数学 幾何学IV), 岩波, (1976)
- 20) P. Naur; *Comm. ACM*, **6**, No.1, 1 (1963)