8角形追跡法による数式化パターンの自動作図*

斎　藤　た　つ　き**

Abstract

We need an effective method to deal with implicit function when formulated patterns or functional curves are drawn. The purpose of this paper is to give a method for drawing of functional patterns. (A study to formulate patterns has been reported by the auther et al.) A drawing vector is decided simply by checking the sign change of a function at vertexes of the seeking octagon. And owing to the course check that rejects the protrusion at a tip and the uselessness of seeking on the opposite side of the course, the drawing comes to be more precise, faster and more reliable than by any other seeking method. Some patterns which cannot be drawn by other methods can be drawn by this method. Therefore, when we draw not only shapes of parts but also functional curves, it can be applied.

1. まえがき

任意線パターンが数式化パターン形式で表現されている場合，それを図式化するさいに有効な陰関数処理が要求される．本論文は自動設計加工において形状が数式化パターン形式で記述されている場合の処理にウエイトをおいて，陰関数表示されたパターンを確実かつ効率的に追跡作図するための一方法である．

数式化パターンは沖野によって提案されたもので[1,2] 複雑な図形を比較的単純な図形要素に分解し，それらの集合として取扱う．数式化された図形要素の集合をマトリックスにまとめたものを数式化パターン・マトリックスとよんでいる．(図形を自動的に数式化する研究は筆者等によってすでに発表されている[3,4].)

いっぽう，数式化パターンを自動作図する方法として NC で従来用いられてきた代数演算方式[8,9]を任意線図形に拡張する方法（階段近似法）等もある．ここで述べる8角形追跡法は図形上に中心をもつ8角形を設定し，陰関数表示された図形式のそれらの頂点での符号判定のみによって作図ベクトルを決定する方式である．そして，判定順序の決定にさいして，進路チェックを併せておこないうる決定法をとることによって追跡効率が向上し，かつ従来作図困難とされていた曲線も作図可能となった．したがって，設計加工対象の形状作図のみにとどまらず任意陰関数の形を描く場合にも応用できる．

2. 数式化パターン

詳細は文献 1，2 に譲るとして，一般的に複雑な形状をもつ設計加工対象を1個の数式で記述できる場合はほとんどないからこれを比較的単純な図形要素に分解しそれらの集合として取扱う．図形要素を P_{ij} とし全体の図形を P とすると，

$$P=\bigcup_{j=1}^{m}\left(\bigcap_{i=1}^{n}P_{ij}\right)$$

である．

ここで P_{ij} は陽関数，陰関数，ベクトル関数等の適当な数式で表現される．線図形の場合は P_{ij} の中に必ず $f(x,y)=0$ の形の式が存在するが，本論文ではそれを GIJE$(x,y)=0$ と表現し，狭い意味で図形式または曲線式とよぶ．

次に，行と列がUおよび∩のそれぞれ異なる結合関係にあるマトリックスを設定し，

$$P=\left\langle\!\left\langle\begin{matrix}P_{11}\\P_{21}\\\vdots\\P_{i1}\end{matrix}\right\rangle\!\right\rangle\cup\left\langle\!\left\langle\begin{matrix}P_{12}\\P_{22}\\\vdots\\P_{i2}\end{matrix}\right\rangle\!\right\rangle\cup\cdots\cup\left\langle\!\left\langle\begin{matrix}P_{1j}\\P_{2j}\\\vdots\\P_{ij}\end{matrix}\right\rangle\!\right\rangle$$

---

* An Auto-drawing of Formulated pattern by Octagonal Seeking Method, by Tatsuki SAITO (Faculty of Engineering, Hokkaido University)

** 北海道大学工学部

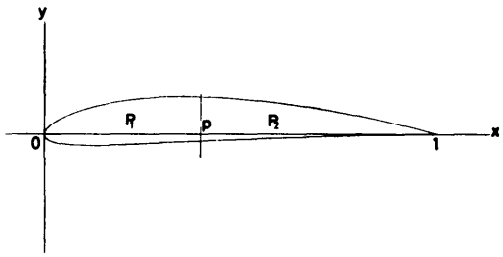


Fig. 1 An example of formulated pattern matrix (NACA airfoil setction).

$$=\left[\left[\begin{matrix}P_{11}\ P_{12}\ \cdots\cdots P_{1j}\\ P_{21}\ \cdots\cdots\\ \vdots\\ P_{i1}\ \cdots\cdots P_{ij}\end{matrix}\right]\right]$$

と定義する．1例を示すと **Fig. 1** のような NACA (現 NASA) の翼型断面形状*は

$$P=[\![P_1\,P_2]\!].$$

$$P_1=\left[\left[\begin{matrix}P_{10}\\ P_{11}\\ P_{12}\end{matrix}\right]\right].\quad P_2=\left[\left[\begin{matrix}P_{20}\\ P_{21}\\ P_{22}\end{matrix}\right]\right].$$

$$P_{10}=\left(\left\{y-\frac{mx}{p^2}(2p-x)\right\}^2+\left[\frac{2m(p-x)}{p^2}\left\{y-\frac{mx(2p-x)}{p^2}\right\}\right]^2-\{5t(0.2969\sqrt{x}-0.126x-0.3516x^2+0.2843x^3-0.1015x^4)\}^2=0\right)^{**}.$$

$$P_{11}=(x\geqq 0).\qquad P_{12}=(P-x\geqq 0).$$

$$P_{20}=\left(\left\{y-\frac{m}{(1-p)^2}(1-2p+2px-x^2)\right\}^2+\left[\frac{2m(p-x)}{(1-p)^2}\left\{y-\frac{m(1-2p+2px-x^2)}{(1-p)^2}\right\}\right]^2-\{5t(0.2969\sqrt{x}-0.126x-0.3516x^2+0.2843x^3-0.1015x^4)\}^2=0\right).$$

$$P_{21}=(x-p>0).\qquad P_{22}=(1-x\geqq 0).$$

と表わされる．

3. 作図アルゴリズム

3.1 作図ベクトル

本方式では図形表示装置として XY プロッタおよび自動製図機の使用を前提としているので作図最小単位*を $+\Delta x$, $-\Delta x$, $+\Delta y$, $-\Delta y$, $+\Delta x+\Delta y$, $+\Delta x-\Delta y$, $-\Delta x+\Delta y$, $-\Delta x-\Delta y$ の8個としたが，CRT ディスプレイに表示する場合でも図形を高精度に表示するためには最小表示単位の追跡が必要となるからハードコピー機器の場合と同等に扱ってよい．したがって，上記の8個の作図最小単位を作図ベクトルとする．

3.2 追跡8角形の設定

1過程の作図ベクトルの終点を A, E, B, F, C, G, D, H とすると4角形 ABCD は正方形である．この正方形を以後探索格子とよぶことにする．次に各終点の中点をおのおの J, K, L, M, N, R, S, I としこれらの各点を結んでできる8角形 JKLMNRSI を追跡8角形，点0を追跡中心点とする．ここで AJ=JE=EK=KB=BL=LF=FM=MC=CN=NG=GR=RD=DS=SH=HI=IA になるように追跡8角形を設定した理由は **3.3** で述べるが誤差を最小にするためである．なお，自動製図機といわれる大型のものには作図ベクトルが8方向以上の機器もあるが，それらの製図機を利用して作図する場合は作図ベクトル数と同数の辺を有する多角形を設定すれば以下のアルゴリズムはそのまま適用できる．通常の作図においては探索格子間隔は最小作図単位（軸に平行なもの，以下同様）の2倍に設定するが作図時間を短縮して描かしたい場合はそれよりも大きく，また，拡大して精密な図面が必要な場合は小さくすることによって任意の大きさ，または任意の精度の作図が可能である．

3.3 アルゴリズム

いま作図しようとする曲線 GIJE$(x,y)=0$ が **Fig. 2** のように追跡中心点Oを通り，辺 IJ 上の点Qを通っているものと仮定する．一般的に GIJE$(x,y)=0$ は図

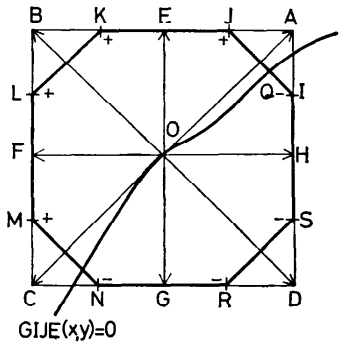


Fig. 2 The seeking octagon and 8 drawing vectors.

---

* NACA 4桁の翼型.

** m, p, t は定数でそれぞれ平均矢高曲線の最大値の最大弦長に対する割合，同値をとる時の x 座標値，最大弦長に対する肉厚比である．

* 1ステップは 0.2 mm から 0.01 mm まで.

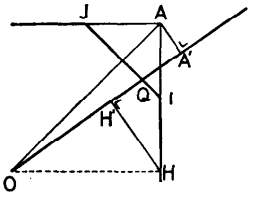


Fig. 3 Minimum error when AA′=HH′.

のように微細に屈曲しているが，微視的なカーブまで作図することは精度的に不可能であるから曲線 GIJE$(x, y)=0$ はこの部分では線分 $\overline{OQ}$ と等価とみなす．(O が曲線から多少離れている場合でも同様にこの微小部分では $\overline{OQ}$ と等価と考える．) したがって，今後は追跡中心点と図形が追跡8角形から外に脱出する点の2点を問題にして議論を進める．次に線分 $\overline{OQ}$ に最も近似したベクトルを作図ベクトルの中から選択する．いま仮に **Fig. 3** のように線分 $\overline{OQ}$ を線分 $\overline{OA}$ で近似した場合と，線分 $\overline{OH}$ で近似した場合を考えてみる．A点から直線 OQ に垂線を下しその足を A′，同様に H 点から垂線を下し H′ としたとき，線分 $\overline{AA'}$ の長さおよび線分 $\overline{HH'}$ の長さを作図誤差とすると図から明らかなように誤差の小さい方は線分 $\overline{AA'}$ である．つまり，曲線 GIJE$(x, y)=0$ が辺 IJ を横断しているときは作図ベクトルには辺 IJ の中心を通る $\overrightarrow{OA}$ が作図しようとする曲線に最も近い．このことより，現在の追跡中心点の座標を (x_o, y_o) とし，

$$[I]=\mathrm{sign}(\mathrm{GIJE}(x_o+s, y_o+s/2))$$
$$[J]=\mathrm{sign}(\mathrm{GIJE}(x_o+s/2, y_o+s))$$
$$[K]=\mathrm{sign}(\mathrm{GJIE}(x_o-s/2, y_o+s))$$
$$[L]=\mathrm{sign}(\mathrm{GIJE}(x_o-s, y_o+s/2))$$
$$[M]=\mathrm{sign}(\mathrm{GIJE}(x_o-s, y_o-s/2))$$
$$[N]=\mathrm{sign}(\mathrm{GIJE}(x_o-s/2, y_o-s))$$
$$[R]=\mathrm{sign}(\mathrm{GIJE}(x_o+s/2, y_o-s))$$
$$[S]=\mathrm{sign}(\mathrm{GIJE}(x_o+s, y_o-s/2))$$

としたとき，[I]・[J]，[J]・[K]，[I]・[S]，[K]・[L]，[S]・[R]，[L]・[M]，[R]・[N]，[M]・[N] の中で負になるものを探索して，その2点にはさまれたベクトルを作図ベクトルと決める．そして，作図ベクトルの終点を新らしい追跡中心点とする．なお，CRT ディスプレイに表示する場合は，作図ベクトルの代りに追跡中心点の座標値を出力すればよい．s は探索格子間隔の 1/2 で，通常の作図においては最小作図単位である．

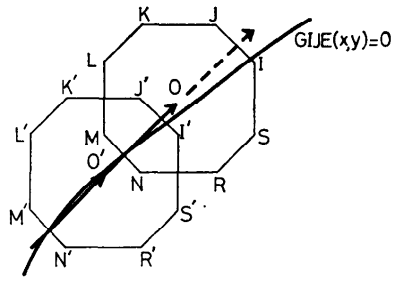


Fig. 4 To search first in the direction of the previous vector is to find the curve rapidly.

判定順序と追跡効率および追跡性の関係

以上の操作を繰返せば曲線を追跡作図できるわけであるが，毎回8点全部を計算するのは非能率的である．そこで，曲り角を除いて図形は探索格子近傍ではほぼ直線と考えてよいから，直前の作図ベクトルの延長方向から探索を始めると曲線を早期に追跡できる．したがって，**Fig. 4** のように直前の追跡8角形を I′J′K′L′M′N′R′S′ とし，作図ベクトルが $\overrightarrow{O'O}$ であればこれから追跡する8角形が IJKLMNRS の場合，I点とJ点から判定をおこなう．もしも，同符号であれば次にK点またはS点を調べる．以下同様に順次L点またはR点，M点またはN点というように符号が逆転するまで探索する．毎回8点全部を計算し，符号が異なる連続2点のうちで絶対値の小さい方の点に作図ベクトルを伸す方法[11]（この方式をここでは“最小法”とよぶことにする）もあるが，計算時間が長くなり効率的に不利である．また直前の作図ベクトル方向から判定せずに **Fig. 5** において，例えば判定順序を常にI点から左回りとすると，辺 SI 上の横断点を追跡する以前に辺 MN 上の横断点（直前に作図した部分）を追跡するために，O′ 点と O 点の間を往復することになり作図不能に陥いる場合がある．これを解決するため

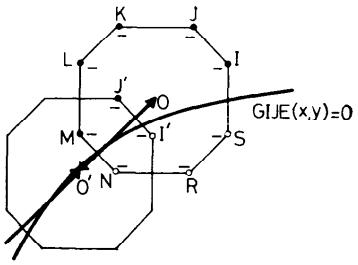


Fig. 5 Going and returning between O′ and O due to constantly clockwise (or counter-clockwise) searching.

に直前の作図ベクトルを記憶させておき，その逆方向のベクトルを除外することもできるが，逆進のチェックを毎回おこなわなければならないうえに，8点目に符号が反転するので探索時間が長くなりやはり不利である．最辺の2点が同符号の場合に，それらの絶対値の差より次の探索方向を決定する方法もあるが，計算ステップが増え，特にミニコン使用の場合は効率的に有利とはいえずまた，追跡性の点でも適当な方法とはいえない．

追跡性の比較検討

次に特殊図形の追跡性を検討する．これらの例はいずれも同一図形要素に属する図形間について問題になることであり，いかなる特殊な図形でも同一図形要素以外のもの同志では問題は起きない．なお，陰関数表示された方程式の解をグラフィカルに求める場合は，分岐の問題や方程式を満足する解曲線が複数個分離して存在しているとき，それらをもれなく補捉する方法が重要な問題になるのでここでは除外して考える．

図形の幅が探索格子間隔より狭くなっている部分では次のような問題が生ずる．

（1） 根元が追跡8角形の1辺の長さよりも細くくびれた突起は無視される．

（2） 間隔が追跡8角形の1辺の長さより狭くなった尖端部は作図できない．

（3） わん曲した尖端で逆行し，作図困難になる場合がある．

（1）の場合を **Fig. 6**（**a**）に示す．（細線が8角形追跡法，破線が最小法，点線がクロス・パターン法，鎖線が階段近似法による追跡状況である．）この場合は8角形追跡法以外の追跡法においても同様なことが起るのであるが，I点とJ点の符号が異符号になり O′ 点と O 点との間で“飛越し現象”を起す．実際の作図例を（**b**），（**c**）に示す．（**b**）では飛越し現象は起きていないが（**c**）ではそれが2個所で起きているため突起部が作図されていない．（この図は飛越しを起すように探索格子を 10 倍にして作図したものである．）また，（2）の場合を **Fig. 7** に示す．これは追跡進行方向につぼんでいるために，もはや符号が反転しなくなるためであるが，この場合，（**a**）の8角形追跡法では尖端部分が無視されるのみであるが，他の追跡法では追跡困難になる．（**b**）が最小法の場合で，A点における追跡でB点に行くべきか，C点に行くべきか選択の基準がない．クロス・パターン法の場合（**c**）

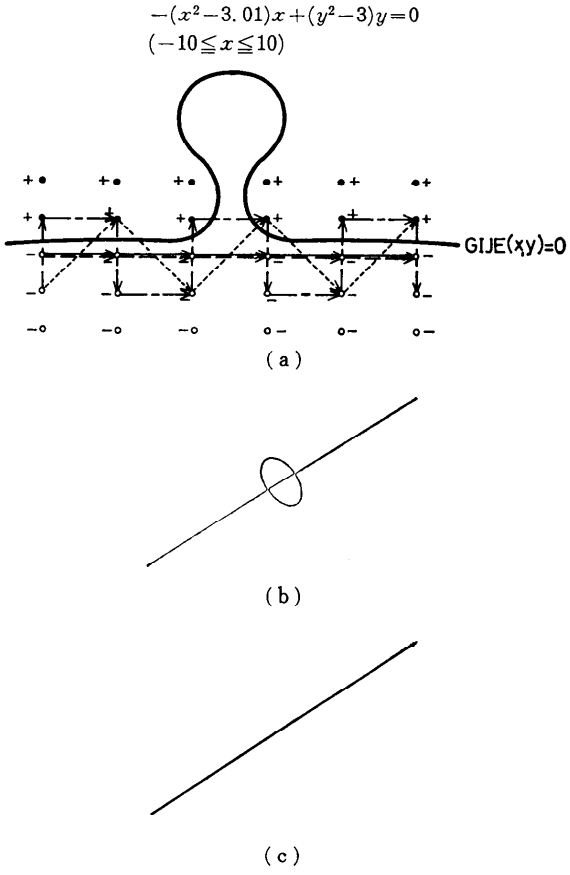


Fig. 6 An explanation of skipping over:
(a) skipping over at the base which is narrower than a step.
(b) a drawing example of no skipping over.
(c) a drawing example of skipping over (drawn in ten times search step as large as (b).

は，この方式には4点のうちで符号が反転する2点の合成ベクトル方向に進む方式と，その2点の中で絶対値の小さい方に作図ベクトルを伸ばす方式とがあるが，前者の場合（点線左側部分）はX点での追跡で，Y, Z, W の3点のいずれを選択すべきかという問題があり，後者（点線右側部分）では，P点における追跡でA点を択るべきかC点をとるべきかが問題になる．（**d**）の階段近似法では，Q点から追跡の腕をいくら伸ばしても符号が反転せず追跡不能となる．（3）の場合を **Fig. 8** に示す．**Fig. 7** と同じく（**b**）が最小法，（**c**）がクロス・パターン法，（**d**）が階段近似法であるが，（2）の場合と同様に追跡困難となっている．特に（**b**）の最小法の場合はA点からD点に戻る可能性もあ

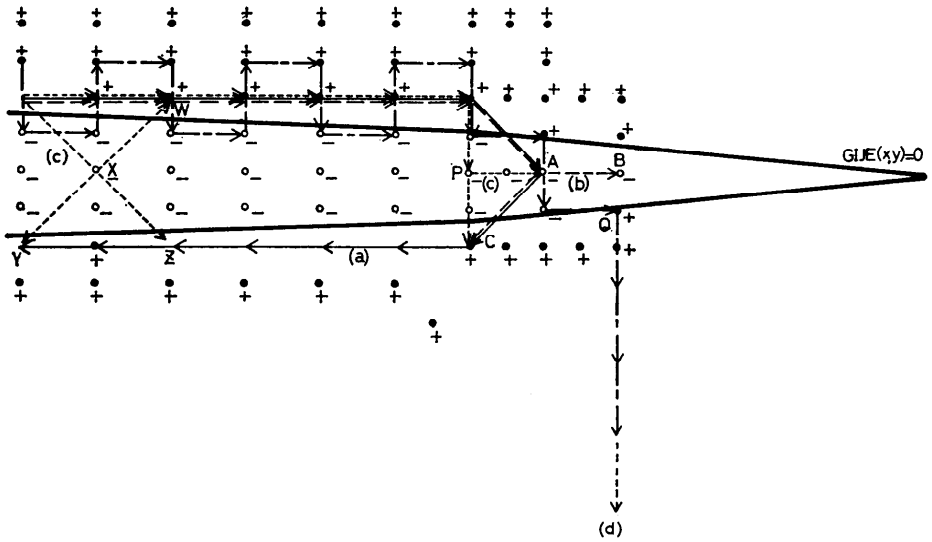


Fig. 7 Impossibility of seeking on a sharp tip by any method (b), (c), (d) except octagonal seeking method (a).

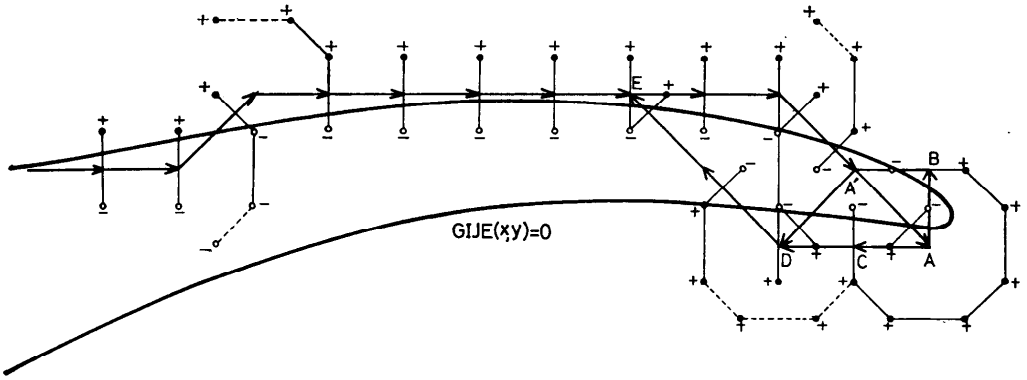


Fig. 8 (a)-1 Impossibility of seeking at point E by octagonal seeking method without course check.

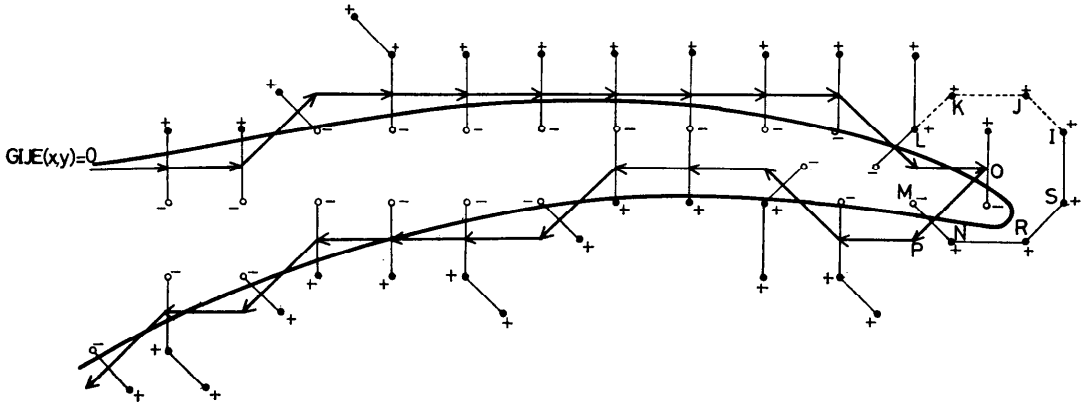


Fig. 8 (a)-2 Seeking certainly and rapidly by octagonal seeking method with course check.

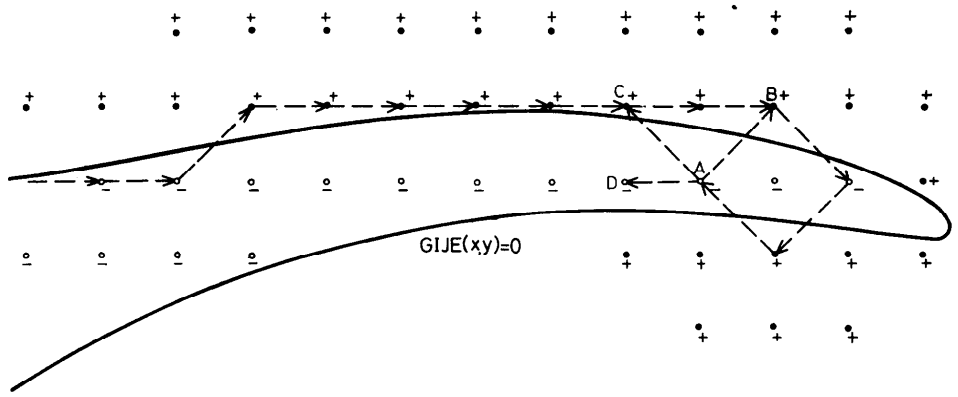


Fig. 8 (b) Impossibility of seeking at point A by minimum method.

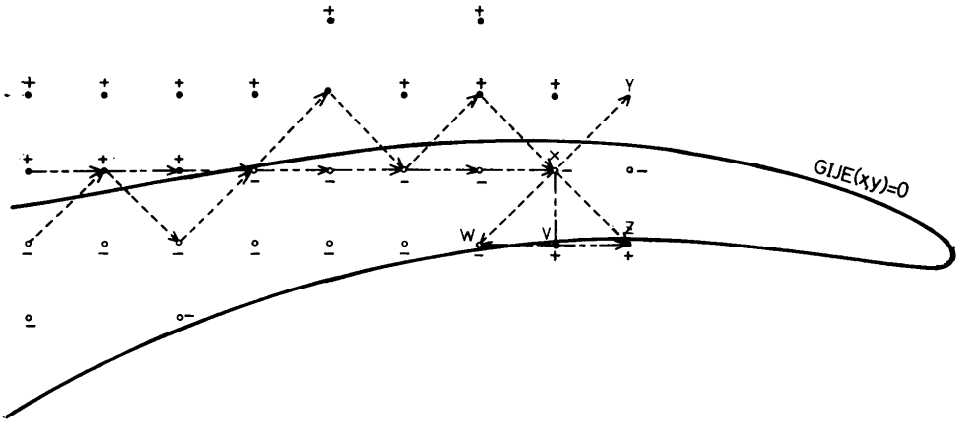


Fig. 8 (c) Impossibility of seeking at point X by cross pattern method.

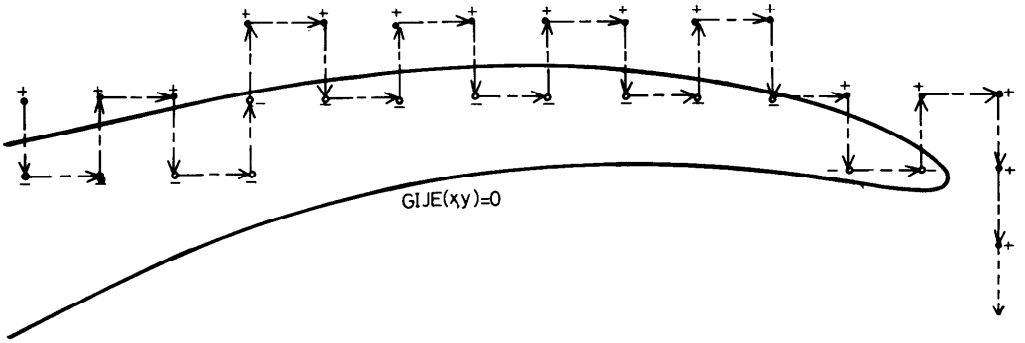


Fig. 8 (d) Impossibility of seeking at the right edge by staircase seeking method.

り，これをチェックすることはほとんど不可能に近い．それはこの場合，D点はA点から数えて5つ前に通過した点であるが，直前の点に逆行したかどうかの判定は容易におこなえても，それ以前の点になると何点前までチェックすれば逆行を防止できるという基準はないからである．いっぽう，8角形追跡法でも今まで述べた判定方法では，**Fig. 8 (a)-1** のように A′ 点での探索で反対側に突出てしまったような場合で，左側の点が先に判定されたときは，A点での探索で作図ベクトルは $\overrightarrow{AB}$ となり，次のB点での探索で A′ 点に戻り，続いてD点を通り結局E点に戻ることになる．また，右側の点が先に判定されたときは，C点，D点を通りやはりE点に戻ることになる．このような問題は8角形の各点の判定順序に方向性を考慮に入れなかったために生じたのである．そこで，進路チェックもかねて符号の判定順序を次のように決める．まず，作図ベクトルの延長方向に対して左側の点から判定を開始する．(付録参照.) もしもその点が直前の追跡での左側の点と同符号であれば右隣りの点を判定する．以下，符号が反転するまで右回りに順に判定する．もしも，最初の左側の点が異符号の場合はその左隣りの点を判定する．以下，符号が反転するまで左回りに順に判定をおこなう．例えば **Fig. 9 (a)** の場合，追跡中心点Oでの探索ではI点の符号をまず判定する．＋でI′点と同符号であるから次にS点を判定する．符号が反転したので作図ベクトルが $\overrightarrow{OP}$ と決定される．(**b**)の場合は，I点が＋で同符号であるから右回りにS点，R点を判定しN点で符号が反転するので作図ベクトルが $\overrightarrow{OP}$ となる．(**c**)の場合は，I点で異符号になるために左回りにJ点，K点と判定していくとK点で反転するので作図ベクトルは $\overrightarrow{OP}$ となる．判定順序を以上のようにすることにより次のような利点が生ずる．

i) 追跡困難な場合がほとんどなくなる．

ii) 判定回数が減少する．角の部分でも最大5点を判定すればよいので追跡速度が向上する．

例えば，**Fig. 8 (a)-2** のO点での探索においてまずI点を判定する．＋の同符号であるから次にS点を判定する．同符号であるから同様にR点，N点を判定しM点で符号が反転するので $\overrightarrow{OP}$ が作図ベクトルとなる．以前のままの判定順序では作図終了部分に逆戻りするうえに J, K, L の不必要な点まで計算することになる．しかし，この進路チェックでも **Fig. 10** のように軸にほとんど平行でかつ，幅が探索格子間隔にほとんど等しい細長い図形の尖端部では，A点の探索で α 点を最初に判定すると＋なので左回りに β 点，γ 点と判定が移り結局，B点に逆戻りし追跡不能となる．このようなことを防止するためには直前の作図ベクトルが，その前の作図ベクトルと同一でない場合は，進路チェックを進路の右側の点から開始すればよい．最終的に進路チェックをこのように決定することによって(3)の作図困難な場合がなくなる．なお，以上のことは分解能に関する問題であるから，拡大図を描くことによって解決することもできる．

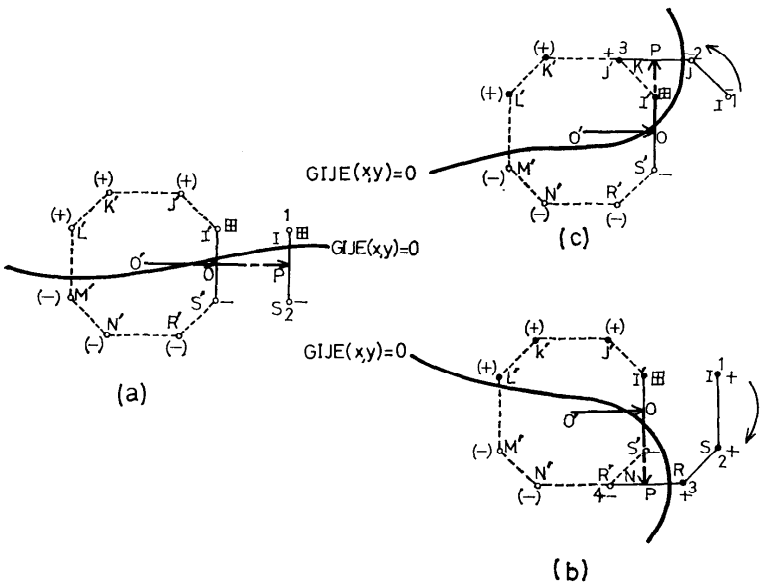


Fig. 9 Course check; when left (or right) hand sign check, clockwise (or counter-clockwise) sequential check if I′·I or (S′·S)>0 otherwise counter-clockwise (or clockwise) sequential check.

追跡効率と近似性の比較検討

階段近似法では xy 軸の45度方向に図形が延びている部分では1点目の判定で符号が反転するが，階段状に追跡するために45度の作図ベクトルを1回出力するためには，結局2回の判定が必要になり

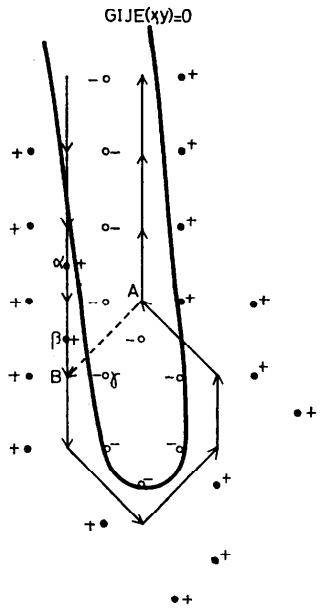


Fig. 10 Course check; if vecctor change occurred at the previous search, when left (or right) hand sign check, the right (or left) hand point must be searched in the first place. Otherwise, the center of the seeking octagon will come back to point B when the pattern is as narrow as a step of the vector and almost parallel to axis.

最良の状態でも8角形追跡法と同じ効率になる．そのうえ，第1象限に存在する図形のみを取扱いうるという制約を別としても，図形が軸に平行になっている部分や，角の部分では数ステップ先まで探索しなければ曲線を追跡できず，さらに，追跡できない場合も多い．また，クロス・パターン法は，1ステップ進むのに4点を判定しなければならなく，最小法では8点全部を計算しなければならないために効率が悪い．8角形追跡法の場合は曲率半径が探索格子間隔の 1/2 以上の部分では2,3点の判定で追跡でき，角の部分でも最大5点を判定すれば必ず追跡できる．近似性に関しては，図からも明らかなように8角形追跡法と最小法が良く，階段近似法，クロス・パターン法で近似度を上げようとすると，数ステップ先まで探索した後に x, y 両成分がある場合は，それを合成して斜方向の作図ベクトルに変換する操作をさらにする必要がある．しかし，8角形追跡法では追跡アルゴリズムを考えるさいに近似の向上をはかっているのでその必要がない．以上の比較をまとめて概略的に表にすると **Table 1** のようになろう．

4. プログラムと作図例

4.1 プログラム

General flowchart を **Fig. 11** に示す．以下に示す作図例で実験したプログラムは FORTRAN で約 150 ステップ*，作図システムとしてはコアメモリ 4 kW のミニコン OKITAC 4300**に WATANABE, WX 530 XY プロッタを直接接続したものである．

4.2 作図例

作図例 1 理想流体中に円柱が存在している場合の2次元理論流線

数式化パターン・マトリックス P は次式で与えられる．

$$P=\left\langle\left\langle\begin{matrix}P_1\\P_2\\P_3\\P_4\end{matrix}\right\rangle\right\rangle.$$

Table 1

		Octagonal Seeking Method		Minimum Method	Cross Pattern Method	Staircase Seeking Method
		without course check	with course check			
Reliability	on a tip	impossibility of seeking occasionally	no impossibility of seeking	impossibility of seeking occasionally	impossibility of seeking occasionally	impossibility of seeking occasionally
	on a corner	impossibility of seeking occasionally	no impossibility of seeking	impossibility of seeking occasionally	impossibility of seeking occasionally	impossibility of seeking
Speed		fast	faster	slower	slow	slow
Approximation		excellent	excellent	excellent	good	not good

* 3000 レベルの FORTRAN で作図ベクトルの出力命令も含めて，追跡アルゴリズムの部分は約 100 ステップ．

** 1 word=16 bit

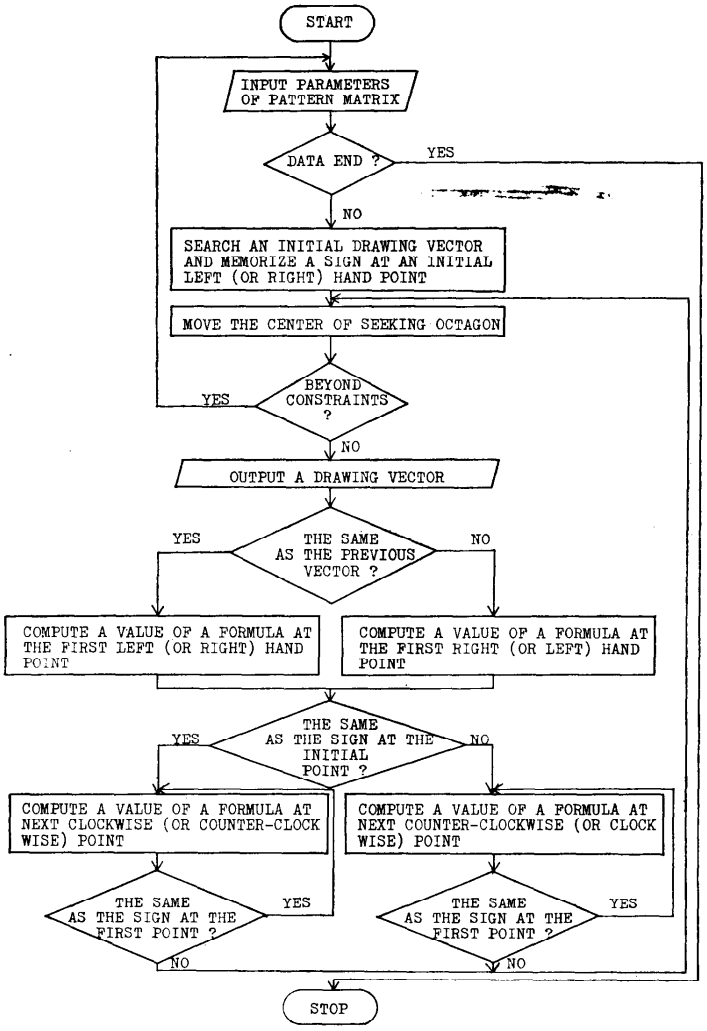


Fig. 11 General flowchart.

$$P_1=\left(-v_0x+u_0y-\frac{(-v_0x+u_0y)}{x^2+y^2}R^2+\frac{2\pi}{\Gamma}\log\frac{\sqrt{x^2+y^2}}{R}-\phi=0\right).$$

$P_2=(x^2+y^2-R^2\geqq 0)$.

$P_3=(-x+b\geqq 0)$.

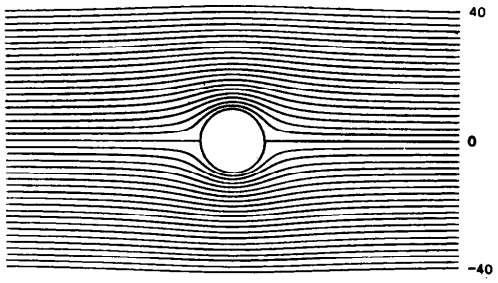
$P_4=(x-a\geqq 0)$.

実際の作図では，流体の y 方向の速度成分 v_0 を 0 とし同じく x 成分を 1，円柱の半径 R を 10 として，円柱の周りの循環量に比例した値 $\Gamma/2\pi=$CONS とおいたとき，CONS をパラメータとして ϕ を 2 きざみで -40 から 40 まで変化させて作図した．**Fig. 12** の（**a**）が CONS=0 のとき，（**b**）が CONS=-20 のとき，（**c**）が CONS=-30 の場合である．

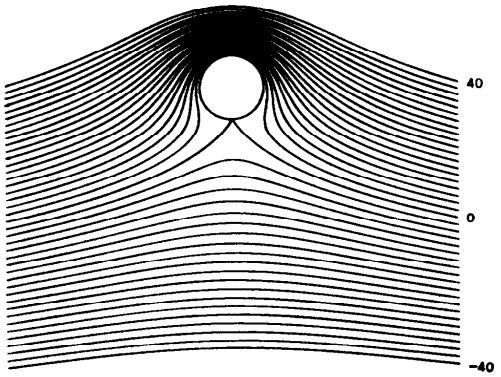
作図例 2　NACA 4 桁の翼型断面形状

数式化パターン・マトリックスは 2 の例示の形式になる．**Fig. 13** で左側は $m=0.05$，$t=0.1$ のとき，p を 0.1 から 0.9 まで 0.1 きざみに変化させたもので，まん中のは $t=0.08$，$p=0.4$ としたときに，m を 0.01 から 0.09 まで 0.01 きざみに変化させたときのであり，右側のは $m=0.09$，$p=0.3$ としたとき，t を 0.05 から 0.40 まで 0.05 きざみに変化させたときの翼型断面形状である．

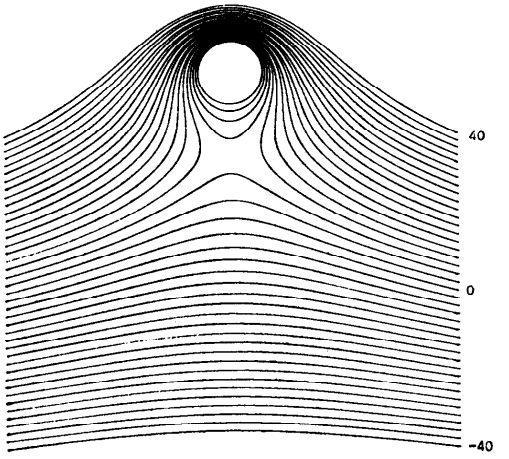
作図例 3　Rosenbrock の式の contour



(a)



(b)



(c)

Fig. 12 A drawing example (stream lines).

わん曲した鋭い ridge をもつために極値探索問題の探索法のテスト関数としてよく検定に利用される Rosenbrock の式 $100(y-x^2)^2+(1-x)^2-f=0$ を作図してみた．外側の contour から順に，$f=10, 5, 1, 0.5, 0$ (点) の場合である．(この式は進路チェックなしでは追跡不能となる例である．) (**Fig. 14**)

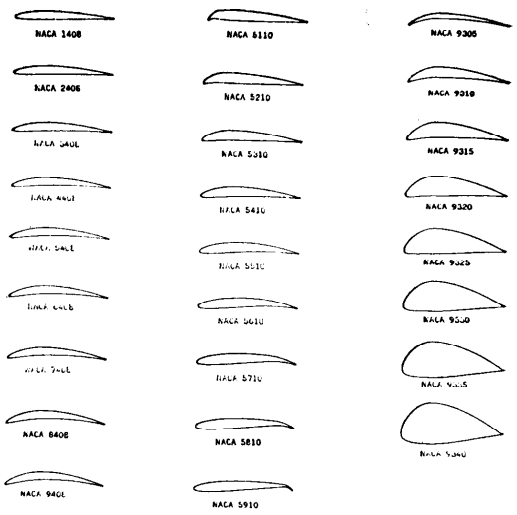


Fig. 13 A drawing example (NACA airfoil sections).

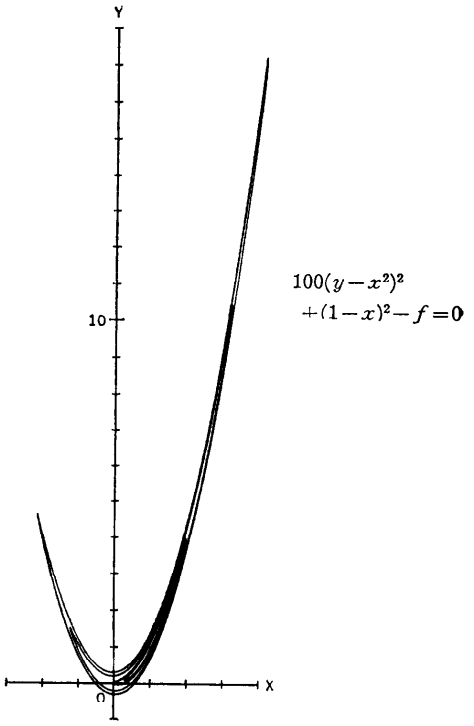


Fig. 14 A drawing example (contours of Rosenbrock's fucntion).

5. あ と が き

本方式は自動設計加工システムにおける数式化2次

元線図形の自動作図を主目的としているが陰関数表示された方程式の図式解法への適用も考えられる．しかし，後者の場合，曲線が交叉，分岐しているようなときはそれらを的確に追跡することは時として多くの困難がある．8個の格子点での符号の変化が2個所より多くの連続2格子点間で検出されればその格子点は分岐点であるとみなす方法[11]もあるが，この方法でも鋭い角度で交叉していて，しかも格子点がその中にはさまれていない場合は分岐の検出ができない．また，解曲線が複数個分離して存在するような問題については，確実にそれらを補捉する方法がさらに重要な課題となろう．したがって，ここではその目的に利用する場合の検討をおこなわなかったが，沖野の方法[13]を併用すれば図式解法も可能となろう．また，数式を作図最小ステップ規模で追跡するこのような方式においては追跡性，近似性もさることながら追跡効率が大きな問題となる．特に自動製図のような作図ステップ数がぼう大な量に達する作図システムにおいてはできるかぎり判定回数を減少させることが重要である．おわりに，本方式に重要な示唆を与えられた北海道大学工学部精密工学科沖野教郎教授に深謝致します．

付　録

判定する点は左側にとっても右側にとってもよい．右側の点に着目する場合は左側の場合と全く対称的に判定順序を決定するとよい．ただし，図形によっては左側を判定点にするか，右側を判定点にするかによってまた，初期追跡方向の違いによって，実用上さしつかえない程度であるが，追跡結果に若干のちがいがでる．例えば，**Fig. 15** のように上の方から作図してきた場合，左側を判定点にすると細線のような結果になり，右側を判定点にすると，点線のようになる．いっぽう下の方から作図してきた場合は，左右どちらを判定点にとっても鎖線のような結果になる．

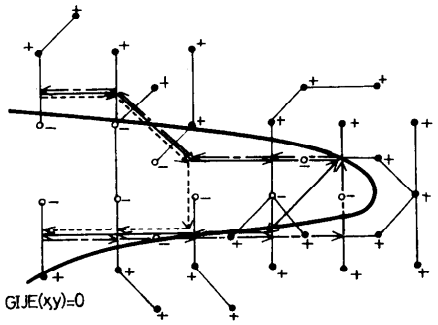


Fig. 15 A little difference between left hand course check (hashed) and right hand course check (real) or between starting at upper and lower (chain) segment of a curve.

参考文献

1) 沖野：“コンピュータによる自動デザイン”，日刊工業新聞社，昭 42.
2) 沖野：“形状の数式化処理とその自動設計問題への寄与”，日本機械学会昭和46年度通常総会講演論文集.
3) 斎藤，野口，村上，沖野：“走査方式による図形認識の基礎研究”，昭43情報処理学会講演予稿集.
4) 斎藤：“走査方式による図形認識装置の研究”，日本機械学会道支部精機学会道支部連合第12回講演論文集 (1968-10).
5) 斎藤：“8角形追跡法による数式化パターンの自動作図”，昭45情報処理学会講演予稿集.
6) 斎藤：“数式化図形の自動作図の1方法（8角形追跡法)”，昭45精機学会秋期大会‘自動設計・加工システムシンポジウム’論文集，pp. 89～100，(1970-10).
7) 斎藤：“数式化パターンの自動作図”，計測・制御に関する北海道地区研究集会予稿集 (1970-11).
8) 研野，稲葉：“数値制御工作機械”，ラジオ技術社，昭 43.
9) 稲葉：“数値制御入門”，日刊工業新聞社，昭 42.
10) 斎藤：“数式化パターンの自動作図に関する2，3の考察”，計測・制御に関する北海道地区研究集会講演論文集 (1971-11).
11) 釜江，村上：“2変数陰関数のグラフ表示”，情報処理，Vol. 12, No. 9, (1971, pp. 582～586).
12) 斎藤：“8角形追跡法による理論流線の自動作図”，昭46情報処理学会講演予稿集.
13) 沖野，佐々木：“モンテカルロ積分を利用した多峰性関数の各峰分離”，第10回 SICE 学術講演会前刷 (1971-11).

(昭和47年4月12日受付)
(昭和47年5月12日再受付)