

## x-y プロッタで漢字を書かせる一つの試み\*

高 澤 嘉 光\*\*

### 1. はじめに

計算機で漢字を取り扱うことは、われわれ日本人にとって非常に価値のあることだが、それには大きな難点がある。この難点は字数が多いことに起因している。普通、ラインプリンタでは活字の数は、アルファベット、特殊文字など全部で数十個ぐらいだが、漢字(当用漢字)やひら仮名を含めると、だいたい2,000個ぐらいになる。これだけ多くの活字を持たせるためには、よほど特殊な構造の印刷機を作らなければならない。

一方、カーブプロッタで字を書くことを考えてみよう。その場合、字は図形としてプログラムで作り出され、活字とは全く違った方法で書かれる。この方法ならば、字を書くのに必要なデータを覚えさせておけば、メモリの許す限りの多くの字を書くことができる。さらに、利点として字の大きさ、形、位置などが、全く自由になる。しかし、欠点としては、何といてもスピードが遅いことだ。

漢字を取り扱う場合、字数が多いことからくるもう一つの問題点には、文字の表示方法(コード)のことがある。アルファベットと数字を使って、そのコードを作ろうとするならば、そのコードが使いやすいものであるように考慮しなければならない。しかし、このコードの問題は、テープ穿孔のできる漢字タイプライタがあれば、直ちに解決すると思う。

われわれはこれまで、FACOM 270-20 の XY プロッタを使って、いろいろな図形を描いてきたが、その際、その図の説明文を日本語で書きたいという欲望から、この漢字システムを考えてきたが、この応用として、日本語の文章の編集などが考えられる。あるいは、グラフィックディスプレイを使って、漢字を教えることに利用すれば、かなりの実用価値を持つと思う。

### 2. 漢字の数値による表現

漢字を計算機に覚え込ませるためには、それを数値で表現しなければならない。一方、プロッタによって書かれる漢字も、活字のような字でなくて、できる限り人間味を持たせた方がおもしろい。そうすると、データの原版になる漢字(実際に人の手によって書かれる)の中の複雑な曲線部分を、できるだけ正確に再現しなければならない。その工夫として、曲線部分には円弧近似を行なった。すなわち、与えられた任意の曲線の両端、ならびに曲率半径が変化する点をデータ点として選びとり、これらの点をもとにして、つぎつぎに円弧を作り出していけばよい。

たとえば、図1のようななめらかな曲線があった場合、図のように5点を取り、仮に1,2,3,4,5と番号をつけ、1から書き出すとする。まず、点1,2,3を通る円弧が、一

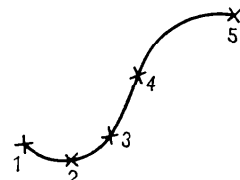


図 1

意的に決まる。つぎに点3で円弧1,2,3に接し、点4を通る円弧も一意的に決まる。同様に点4で円弧3,4に接し、点5を通る円弧も一意的に決まる。また、図2のように点1と2の間が直線、点2と3の間が曲線するとき、点1と2の区間は直線で結び、点2と3の区間は、点2において直線に接し、点3を通る円弧で近似すればよい。

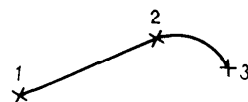


図 2

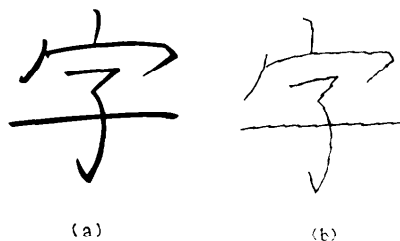


図 3

\* A method to draw Chinese characters by X-Y plotter, by Yoshimitsu Takasawa (Mathematical Engineering, the Faculty of Engineering, Tokyo University)

\*\* 東京大学工学部計数工学科

このような曲線の近似方法は、データ点の割に、近似の度をよくすることができる。図3(a)は人の手によるもので、(b)はこれを上の方法で近似し、プロッタで描いたものである。

つぎに、この近似法によって、漢字を数値表現することを考えよう。まず、図4のように、たて・よこ

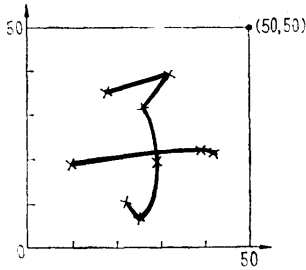
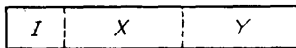


図4

50mmの箱の座標系を作り、この中に漢字を書く。この漢字の上に適当にデータ点をとっていく。それらの点の座標(X,Y)と、その点までペンがどんな状態であるかという情報Iによって、1個の(その1点に関する)データが作られる。データはつぎの形を有する。



ここで、X,Yは  $0 \leq X, Y \leq 50$  なる整数で、図4のような座標系において、1mm以下を四捨五入して無名数化したものである。

ペンの状態を示す情報Iは、第1表の規則に従って決められる。

第1表

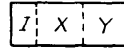
I	ペンの状態
0	ペンを上げてその点(X,Y)までくる。点(X,Y)は線分の書き出す始点である。
1	前の点からその点(X,Y)まで、直線を引いてくる。
2	前の点からその点(X,Y)まで、円弧を引いてくる。
3	前の点からその点(X,Y)まで、直線を引いてきて、かつ、その点はその漢字の最後のデータ点である。
4	前の点からその点(X,Y)まで、円弧を引いてきて、かつ、その点はその漢字の最後のデータ点である。

図4の漢字を数値化すると、つぎのようになる。

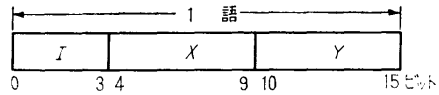
01835 13139 12632 02632 22920 22507

12211 01019 13922 34221

1個のデータ



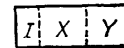
は十進五桁の数からなるが、これを1個の十進の整数として覚えさせようとする、FACOM 270-20では整数1語に納らないこともある。そこでX,Yは50より小さい正の整数であることを利用して、I, X, Yを分解しつぎのように格納する。



1語は16ビットであるから、最初の4ビットにI、つぎの6ビットにX、次の6ビットにYを入れる。

### 3. 漢字の作成

計算機の中に覚えているデータ



の座標(X,Y)はあくまでも50x50の大きさの仮の座標系である。実際にプロッタで1個の漢字を書くには、この他に5個のデータ  $x_0, y_0, h, w, \theta$  が必要である。

( $x_0, y_0$ ): 紙面上の座標系に基づく、その漢字の位置

h: 漢字の高さ

w: 漢字の横幅

$\theta$ : 紙面上の座標系に対するその漢字の傾き

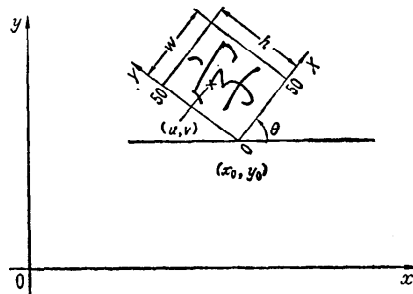


図5

これを紙面上の座標系(x,y)と、仮座標系(X,Y)の関係で図示すると図5のようになる。座標系(X,Y)上の点(U,V)の座標系(x,y)上での位置を(u,v)と

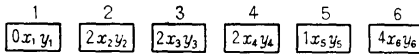
すると、両者につきの関係がある。

$$u = x_0 + U \frac{w}{50} \cos \theta - V \frac{h}{50} \sin \theta$$

$$v = y_0 + V \frac{h}{50} \cos \theta + U \frac{w}{50} \sin \theta$$

この関係式によって、仮座標上における点の位置 (U, V) と、5個のデータ  $x_0, y_0, w, h, \theta$  から求められた (u, v) が紙の上における絶対位置になる。

つきにペンの状態 I に関してめんどうなことがある。いま



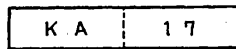
のような6個のデータに基づいて曲線を作り出すことを考える。ここで最初の数はペンの状態 I を示し、 $x_i, y_i$  は点の座標を表わす。

この場合、 $I=0$  のデータに続いて  $I=2$  のデータが現われても、それはすぐに使われなくて、つぎの  $I=2$ 、すなわち3番目のデータにきて、初めて円弧を引くことができる。それ以外の場合には、データが与えられれば、すぐに直線あるいは円弧が描かれる。

なお、円弧は長さ 1mm の弦を連ねて近似される。

#### 4. 漢字のコード

漢字のコードを決める方法として、1, 2, 3, ……の一連番号をつけることも考えられるが、それでは使用には少々めんどうである。ここでは、アルファベット2個と数字2桁の計4個の文字で作った。



アルファベット2個は、漢字を原則として訓読みにして、それをローマ字で書き表わした前2文字をとったものである。後の数字は同じアルファベット2個のグループの中での通し番号である。この方法だと、漢字の持つ情報をわずかではあるが生かしており、また取り扱う数も比較的小さいものだけであるから都合がよい。

#### 5. データの索引方法

1個の漢字に平均20個のデータが必要であるから、2,000個の漢字だと40,000個のデータ、すなわち

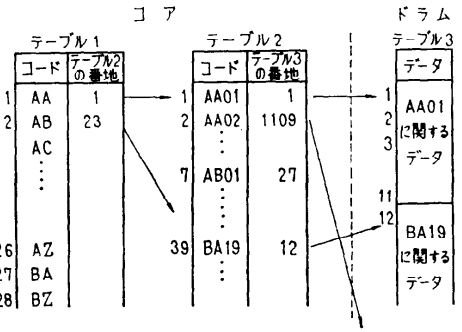
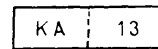


図6 メモリマップ

40K語の記憶容量が必要になる。これだけのものはコアには格納できないから、ドラムを使わなければならない。この40K語の情報の中から、必要なデータを速く取り出すには少々工夫がいる。

この場合、コア上に図6のテーブル2のような対応表を持ち、ドラム上の各漢字の先頭番地を記憶しておかなければならない。いま、コード KA 13 が現われたとき、テーブル2において、最初から順にこのコードを捜していかなければならないが、テーブル2のほかに、もう1つ余分のテーブル1を作っておけば、コード KA 13 がくれば、KA によってテーブル1からテーブル2におけるKAの先頭番地がわかり、それより (13-1) 先の番地に KA 13 のデータのドラム上の開始番地が記憶してある。もちろん、テーブル2はコードに関して、ソーティングしてあるものとする。



というコードの漢字のデータのドラム上における先頭番地は、つぎの式で表わされる。

$$T2(T1(KA)+13)$$

$$\begin{matrix} \uparrow \\ K & A \\ 26 \times (11-1) + 1 \end{matrix}$$

$T1(X)$ ,  $T2(Y)$  はそれぞれテーブル1, 2のX, Y番目の要素の値である。

このような余分のテーブル1を設けることによって、コードを見つける時間をかなり短縮することができる。

ここに述べた方式で、計算機を使って書いた実例を次に示す。

あ	い	う	え	お	か	き	く	け	こ
AA11	II11	UU11	EE11	OO11	KA11	KI11	KU11	KE11	KO11
さ	し	す	せ	そ	た	ち	つ	て	と
SA11	SI11	SU11	SE11	SO11	TA11	TI11	TU11	TE11	TO11
な	に	ぬ	ね	の	は	ひ	ふ	へ	ほ
NA11	NI11	NU11	NE11	NO11	HA11	HI11	HU11	HE11	HO11
ま	み	む	め	も	や	ゆ	よ	ら	り
MA11	MI11	MU11	ME11	MO11	YA11	YU11	YO11	RA11	RI11
る	れ	ろ	わ	み	ゑ	を	ん	が	で
RU11	RE11	RO11	WA11	WI11	WE11	WO11	NN11	GA11	DE11
ば	。	。		菌	管	鼻	算	光	策
BA11	KK11	PP11	BL11	HA12	KA12	HA13	SA12	TO12	SA13
点	答	黒	等	黄	筆	麦	第	塩	鳴
TE12	KO12	KU12	HI12	KI12	FU11	MU12	DA11	SI12	NA12
竹	鳥	競	計	機	漢	字	処	理	非
TA12	TO13	KI13	KE12	KI14	KA13	JI11	SY11	RI12	HI13
常	難	何	工	夫	行	思	例	分	様
JY11	MU13	NA13	KU13	FU12	IK11	OM11	RE12	WA12	SA14
一	度	情	報	記	憶	平	移	動	拡
IC11	DO11	JY12	HO12	KI15	OK11	HE12	ID11	UG11	KA14
大	縮	小	回	転	要	易	出	来	容
DA12	SY12	SY13	KA15	TE13	YO12	II12	DE12	KU14	YO13

図 7 かなと漢字のコード化の一例

計算機で漢字を処理することは、非常に難しいですが、何とか工夫すれば、うまく行くと思います。しかし、この例からも分かる様に、一度情報を記憶してしまえば、平行移動、拡大、縮小、回転等が容易に出来ます。

図 8 計算機で書いた文章

(昭和 44 年 5 月 12 日受付)