タートル・ディスプレイによる怪物曲線

ISHIDA, Norimichi / 石田, 則道

(出版者 / Publisher)
法政大学工学部
(雑誌名 / Journal or Publication Title)
Bulletin of the Technical College of Hosei University / 法政大学工学部研究集報
(巻 / Volume)
20
(開始ページ / Start Page)
119
(終了ページ / End Page)
129
(発行年 / Year)
1984-03
(URL)
https://doi.org/10.15002/00004071

タートル・ディスプレイによる怪物曲線

石 田 則 道*

Draw Monster Curve with Turtle Display
——Wang's Robots Language——

Norimichi ISHIDA*

Abstract

This is a "Tiny" language depending on turtle geometry. I'm call "KAME" meaning of turtle.

"KAME" is implemented on personal-computer with BASIC language.

Rudimentary commands consisting of a single character are executed by the "KAME" as soon as they are typed-in. Immediately respond on display.

The following are a few examples of programs written in "KAME".

These Monster-Curve (Dragon, Hilbert, Sierpinski, Snow-piece) are heavily biased toward graphics and recursive macro definitions.

§1. はじめに

「1歩前進，45度まわって，2歩前進…」とくり返していくと，その足跡が図形となり，いろいろな模様が描ける。これは Lichen Wang（王理璜）のロボット言語として知られている。

ロボット言語は「亀の子幾何学」（turtle geometry）の原理であって，その行動様式をディスプレイに表わした（タートル・ディスプレイ）ものである。この言語はロボットのコントロールに使えるわけだが，ロボットの足跡を画面に置き換え，パソコン（パーソナル・コンピュータ）でディスプレイを見ながら，キー・ボードから指命して，その軌跡を作っていくことができる。

文献 2) に，8080 アセンブラのインタプリタ版があったが，今回，コマンドを追加し，画面も見やすく考慮し，BASIC（インタプリタ）で作ったので紹介する。

§2. コマンドとその使い方

この言語（私は "KAME" 亀と呼んでいる）は処理を簡単にするためにコマンドは1文字で構成する。そのためにプログラム（コマンド列）がコンパクトに書けるし，関数も定義できる。また，新しく，色，円，色ぬりなどのコマンドを追加したので，図形も豊富に描け，とてもきれい

* 法政大学計算センター

です。表1が，この言語のコマンドである。

画面は図1のような構成で！（入力コマンドの促進マークで，ここがコマンドエリア）に続いてコマンドを入力し，return キーを押すと，それに対応する動作が作図エリアに表現される。また，左上は表示エリアで，アキュムレータの値，方向，座標値，カラー・コードが表示されている。実行開始時はアキュムレータは0，北向き，作図エリアの中心に赤い亀がいると思って下さい。

表1 コマンド表

コマンド		意味	説明	使用例
動作	F	Forward	一歩前進（足跡を残して）	*n*F（*n*：1～50），AF
	J	Jump	一歩ジャンプ（足跡を残さず）	*n*J，AJ（A：Accumlator）
	B	Backward	直前のコマンド（F又はJ）を消す Fのとき…足跡消して元の位置へ Jのとき…元の位置へ	B
	E	En (circle)	円を描く	*n*E（*n*：半径）
方向	R	Rotate	右方向45度回転	*n*R（*n*：1～7），AR
	N	North	北向き（↑）に位置付け	N
設定	H	Home	作図エリアの中心へ	H
	I	Iro (color)	色（カラー・コード）の指定	*n*I（*n*：1～7），AI
	P	Paint	色付け	*n*P（*n*(1～7) で色付け，そのときのカラー・コードは境界色）
	C	Clear	作図エリアをクリア(足跡すべて消去)	C
	L	Line	中心線を引き，きざみを入れる	L
変数	A	Accumlator	アキュムレータを示す	AF，AJ，AE
	+	increment	アキュムレータの内容に1を加える A+はアキュムレータの内容を2倍	+，++，3+
	−	decrement	アキュムレータの内容から1を引く A−はアキュムレータを0にする	−，−−，4−
判定	T	Test	T(*a*)(*b*) (A)>0 のとき*a*を，そうでなければ*b*を実行	T(RF)(2R)
定義	D	Define	関数を定義する（再帰的利用可能） D*f*… *f*：予約コマンド以外の英字	DX4(2F2R)
補助	( )		} ペアで利用 くり返しや，見やすく	4(2F2R)(2F)(2R)
その他	HELP	Help	！（入力コマンド促進マーク）に続いて入力すると（以下同じ）上述コマンドを作画エリアに表示する	
	LIST	List	定義済関数をコマンドエリアにリストしなおす	
	SAVE	Save	定義済関数をディスクに格納する（ファイル名を付けて）	
	LOAD	Load	ファイル名の付いた定義済関数をディスクから呼び出す	
	SMPL	Sample	コマンド列のサンプルを表示する	
	END	End	プログラムの実行を終了する	

FF と入力すると赤い線が2歩分北向きに進んで（足跡を残して），座標値は（0, 2）となる。以降，RRFFRRFFRRFF return と入力すると，□（四角）が描ける。このコマンド列は“2歩進んで90度右にまわることを4回くり返した”ことを意味し，4(2F2R）としても同じ図が描ける。

F, R, J コマンドいろいろ組み合せて入力していくと，作図エリアに足跡ができる。

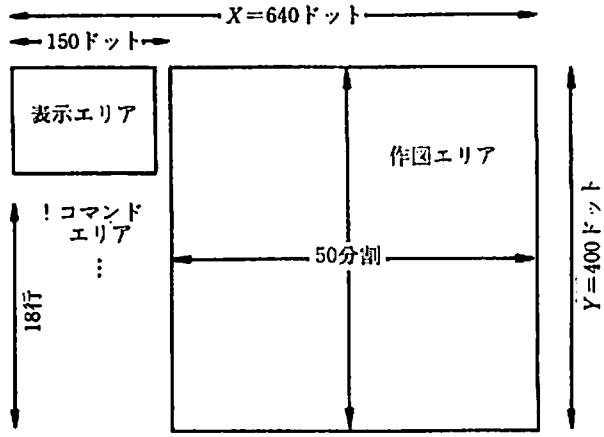


図1　画面構成

この作図エリアは50分割（ソース・プログラムを修正すれば分割は自由に変更できる）なので，$-25 \leqq x, y \leqq 25$ の区間は自由に動きまわれる。この範囲を越えると Range over！のメッセージが出る。そこで，B（戻る）とすればその直前の動作コマンド（FまたはJ）は消去できる（表示エリアの座標値が変化するからわかる)。また，そのまま続けてもよいし，画面をクリア（Cコマンド）して，必要ならライン（Lコマンド）を入れて，書き直すこともできる。使用していない英字を入力すると “Command error!” と表示し，“Next…Y/N” と聞いてくる。続けるならYと応答すればよい。

“KAME” では変数 A (accumlator) を使うことができ，＋で1を加えたり，－で1を引くことができる。特別な場合として，A－はAの内容を0にすること，A＋はAの内容を2倍することを意味する。コマンド忘れたら “HELP”，プログラムの終りは “END” と！の直後に入力する。

§3. 初期画面

“KAME” を実行すると図2のような初期画面が出てくる。

ここで，0～3の任意の数字を入力することによって，いろいろなモードが選択できる。

“KAME” のコマンドを知っていれば，“3” を選択することで(図2)，ただちにコマンドが入力できるモードになる。

初めての人は “0” を選択し，(図3)，“KAME” の概要から入っていく。コマンドを忘れた

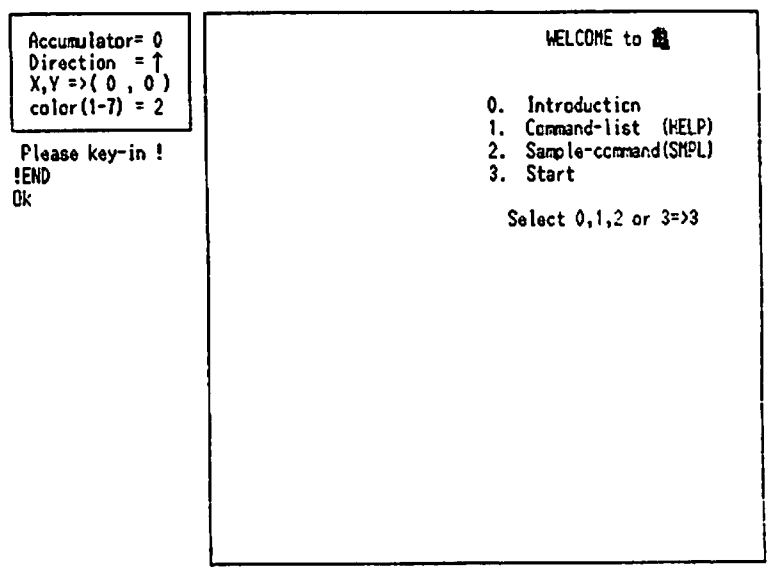


図2　初期画面で “3” を選択した場合

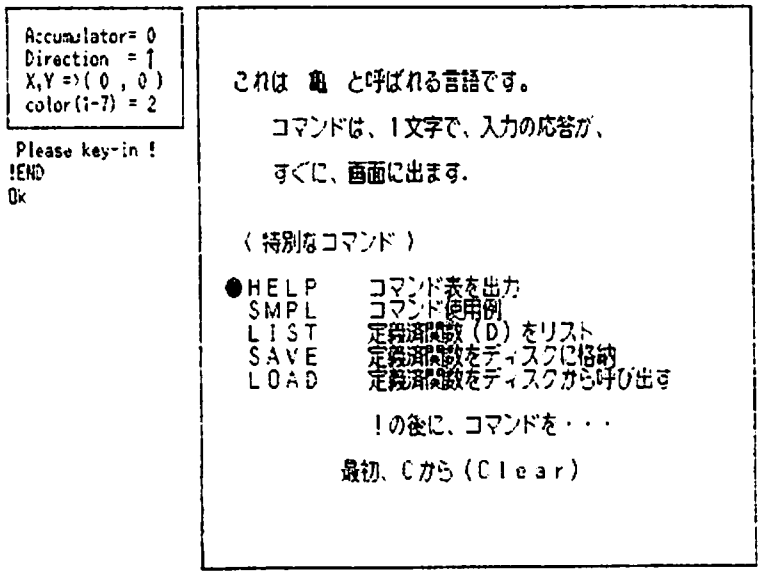


図3　"0" を選択した場合

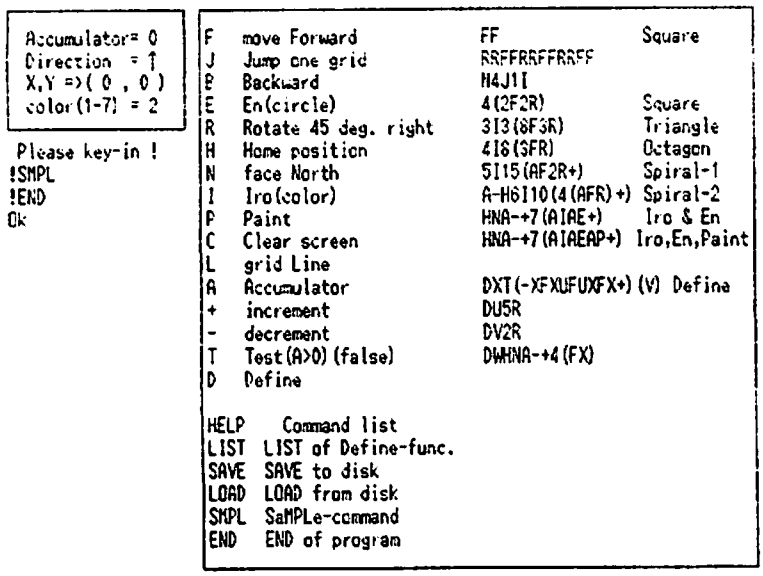


図4　"1" を選択後，"SMPL" コマンド投入

人は "1" と入力すれば（図4），"KAME" で使用できるコマンドを，作図エリアに表示する（"HELP" コマンドを選択した場合と同じ）。

図4では "1" を選択した後，"SMPL" コマンドを入力した場合で，作画エリアの右半分に，コマンド列のサンプルを表示する。これは初期画面で，"2" を入力した場合と同じである。

このように，"KAME" では，いちいち，マニュアルを見なくても，画面と対話しながら，処理が続けれるように考慮してある。

§ 4.　簡単な作図例

三角形，八角形，うずまき，簡単に描ける。コマンド列と図を対応して下さい。

表2　作　図　例

形	コマンド列	説　明	図番号
三　角　形	3(8F3R)	8歩進んで135度まわることを3回くり返す	図5
八　角　形	8(3F R)	3歩進んで 45度まわることを8回くり返す	図6
うずまき1	15(A F2R +)	アキュムレータの内容分前進し90度回転， アキュムレータに1加えることを15回くり返す	図7
うずまき2	10(4(A F R)+)	アキュムレータの内容分前進，45度まわることを4回くり返した後にアキュムレータに1を加えることを10回くり返す	図8

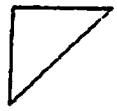


図5　三角形

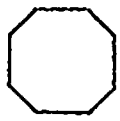


図6　八角形

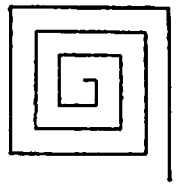


図7　うずまき1

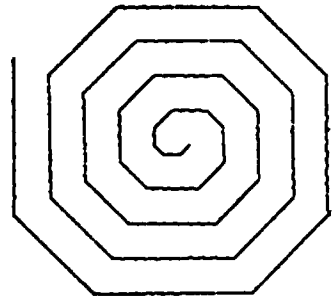


図8　うずまき2

§5.　関数定義と再帰的呼び出し

A（アキュムレータ）の内容を一辺としたとき四角形は 4(AF2R) と書き表わした。それを関数の形で書くと，DX4(AF2R) とな。る先頭のDが関数（define）であることを意味し，関数の名前はX（名前はコマンドとなる英字以外ならよい）で，一度定義すれば，あとは自由にコマンド列に書いてよい。

関数はコマンドと違って，定義しただけでは実行せず，例えば，関数定義のあとに，3(N+XRAJ) と入力すると，北向き，Aに1を加え四角を描き（関数Xを呼び出し），右45度の方向に，Aの内容だけジャンプすることを3回くり返す（図9）。

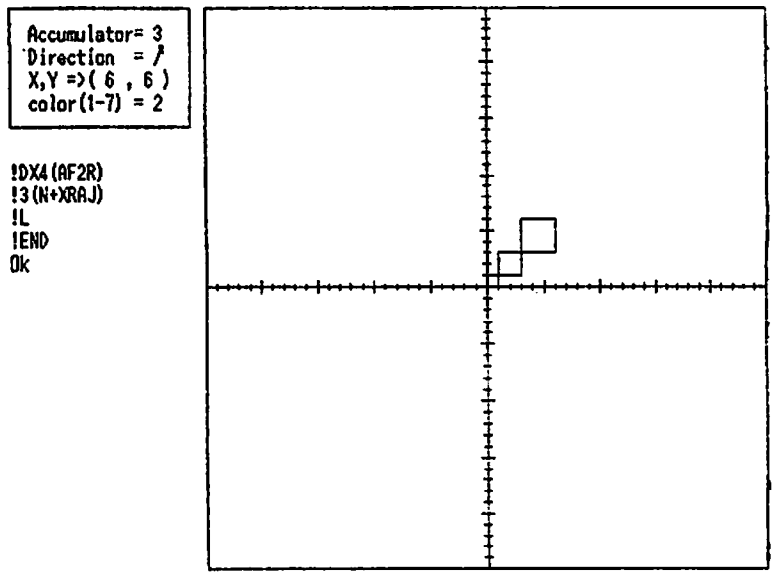


図9　関数定義

つぎのように関数X，Y，Zを定義し，

DX6R23J6R20J N2R

DY46F6RF6RZ

DZ46F2RF2RY

ここで，XYと入力すると，最初に関数Xが呼ばれ，作図エリアの左下，右方向に位置付けされ，続いて関数Yが呼ばれる。関数Yは46歩進み（足跡を残して），270度（6R）回転，上向き（↑）になり，1歩進んで270度回転，左向き（←）になる。そして関数Zを呼びに行く。関数Zは46歩前進90度回わり（↑），1歩前進し90度回わり（→），Yを呼ぶ。

関数Yから関数Zを呼び，その中で関数Yを呼んでいる。これが再帰的呼び出しであり，FORTRANなどでは許されない呼び方である。

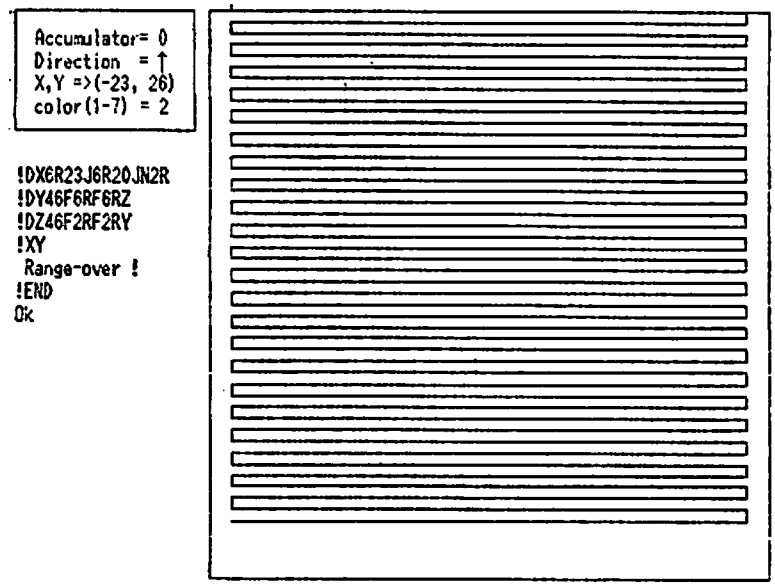


図10 再帰的呼び出し

この結果は，図10のようになり，作図エリアから飛び出し，“Range-over !” のメッセージを出し，つぎのコマンド待ちになる。

§6. 怪物曲線

“KAME” での関数，そして再帰的呼び出しを使うことによって，短いコマンド列で，いろいろな怪物曲線が描ける。ここではドラゴン曲線，ヒルベルト曲線，シェルピンスキー曲線と雪片曲線について描く。

6.1 ドラゴン曲線

図13～図18がドラゴン曲線です。海中を竜が鼻や尾をくねくねはいまわっている感じがする。この曲線は“直角に曲がるとき交差しない”ことが特徴で，ウィリアム・G. ハーター（カリフォルニア大学），ジョン・E. ハイウェイ（NASA），ブルース・A. バンクス（NASA）らによって解析され，ドナルド・E. クヌース（計算機科学者）やチャンドラ・デイビス（数学者）らの研究論文もある。このドラゴン曲線の作り方にいくつかあり，ここで簡単に述べる。

1) 2進の数列法

曲線を一端から他端へたどるとき，2進数字の列に対応させ，1を左曲がり，0を右曲がりとしたとき，各次数は帰納的な手順で作成できる。そのやり方は1を加えて1より前にあるすべての数字列を書き加えるが，そのとき書きたす数字の中央の値を反転させる。このようにして作ると表3のようになる。

表3 数 列 法

次数	数 列
1	1
2	1 1 0
3	1 1 0 1 1 0 0
4	1 1 0 1 1 0 0 1 1 1 0 0 1 0 0
5	1 1 0 1 1 0 0 1 1 1 0 0 1 0 0 1 1 1 0 1 1 0 0 0 1 1 0 0 1 0 0

2) 紙の折りたたみ法

まず，1枚の紙を用意し，最初半分に折って，直角に開いたとき，それを横からながめたものが1次のドラゴン曲線である。以降同じ方向に半分に折り，直角に開いていく。2回折りで2次，3回で3次，一般に n 回折れば n 次のドラゴン曲線ができる。山折りを「1」，谷折りを「0」とすれば，さきほどの2進の数列法に対応する（図11）。

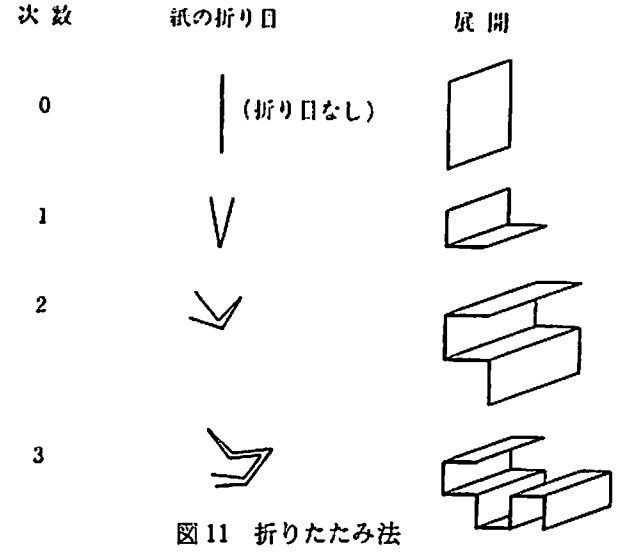


図11　折りたたみ法

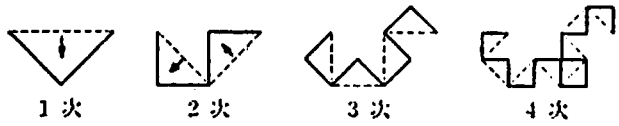


図12　幾何作図法

図13　1次　　図14　2次　　図15　3次

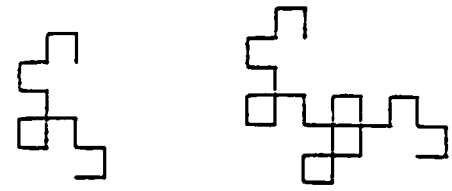


図16　4次　　図17　5次

3)　幾何学的作図法

これは1つの直角から始め，随時，各線分を直角な2辺に置き換えていくやり方である。この方法でも簡単に作れる（図12）。

さて，ドラゴン曲線を，“KAME”でプログラムすると，2つの関数で表わすことができる。

DXT(−X6RK+)(2F)

DKT(−X2RK+)(2F)

この関数はAの内容を判定するTコマンドで定義されていて，つぎのような意味になる。

if A>0 *then* (−X6RK+)

else (2F)

関数Xは，A（アキュムレータ）が正なら，アキュムレータから1を引いて，Xを呼ぶ（自分自身）。270度まわってKを呼び，アキュムレータに1を足す。アキュムレータが0のときは一辺のサイズ（今は2Fとした）が実行され線が描かれる。関数Kも同じ形をしているが，回転角（90度）だけが違う。

```
Accumulator= 6
Direction  = ←
X,Y =>( 16, 0 )
color(1-7) = 2
!DXT(-X6RK+)(2F)
!DKT(-X2RK+)(2F)
!6+X
!END
Ok
```

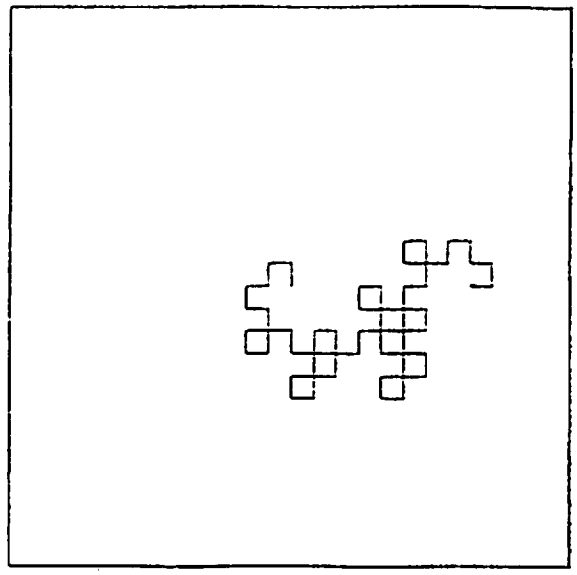


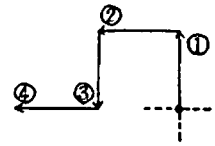
図18　6次　ドラゴン曲線

このように，2つの関数を定義して，A（アキュムレータ）に次数をセットし，Xを呼び出すとドラゴン曲線が作り出される。

2次の場合のドラゴン曲線の動きを追ってみると：

表4

階層	実行過程 (→)
0 ACC	2 + X 2
1 ACC	XT(−X 6R K +) 1 1 2
2 ACC	XT(−X 6RK+) KT(−X2R K+) 0 1 0 1
作図	2F① 2F② 2F③ 2F④



この結果は図14となり，Accの値は2，方向は←，座標は（−4, 0）で完結する。

6.2 ヒルベルト曲線とシェルピンスキー曲線

“正方形の中を連続的に動く点が，正方形内または境界上のすべてを通る線”をペアノ曲線といい，2つの端点をもつペアノ曲線がヒルベルト曲線(図19～図23)であり，開いた形の曲線である。

一方，ペアノ曲線の閉じた形の曲線がシェルピンスキー曲線（図24～図27）である。この曲線は見た目にも，きれいで，それに色付けするとなおさら，あざやかです（図27)。これらの曲線は，すっきりした形で再帰的に定義できる。

図19 1次

図20 2次

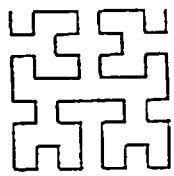
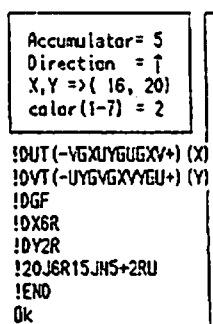


図21 3次

ヒルベルト曲線の場合

```
DUT(−V2F6RU2R2FU2F6RV+)(6R)
DVT(−U2R2FV2F6RV2R2FU+)(2R)
3 +2RU
└┘
 ↑
 次数
```



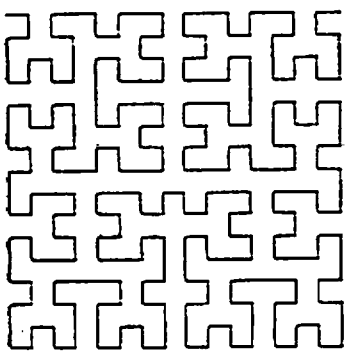


図22 4次

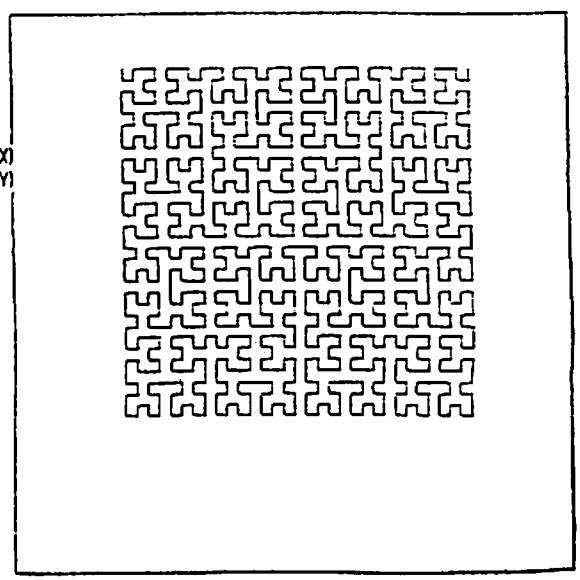


図23 5次 ヒルベルト曲線

シェルピンスキー曲線の場合

```
DXT(−X2FX2FX5R2F5RX2FX+)(2R)
DY3R4(2FX)
3+Y
└┘
 └──次数
```

これらの曲線を描くとき，次数が大きくなると，作図エリアにうまく入るように始点を決める必要がある。

図24　1次

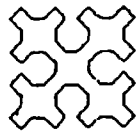


図25　2次

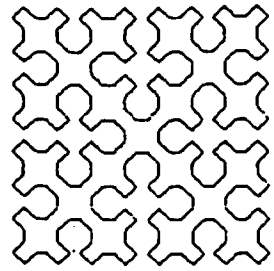


図26　3次

6.3 雪片曲線

前述の怪物曲線（ドラゴン曲線，ヒルベルト曲線，シェルピンスキー曲線）の他にもう1つマンデルブロ（ポーランド生まれのフランスの数学者，現在I．B．M社ワトソン中央研究所）の正方雪片曲線を描いてみる。これも簡単に関数の再帰的呼び出しで定義できる（図28～図30）。

図28　1段

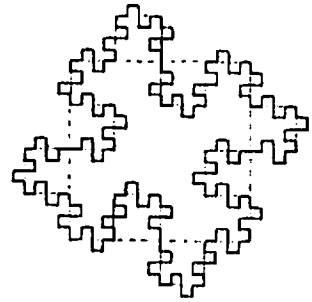


図29　2段

“KAME”では，ほかにもいろいろなものが描ける。円（Eコマンド）が使えて，それに色付け（Pコマンド）ができるので，とてもカラフルな作品もできる（図27，図30，図33）。

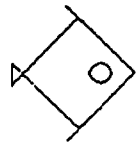
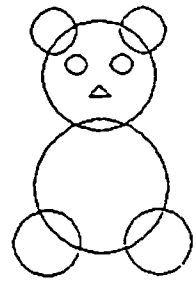


図31

```
!R4(5F2R)4RF3R2F3RF
!5J2RF4RJ7R10J7RF
!4R6J3R3JE
!END
```



```
!6E10J5E
!J6R2JE4R4JE
!2J6R3J2E6R8J2E
!HN4R5J6R5J3E4R10J3E
!HN8J2RJ4R2F3RF2RF
!END
```

図32

§7. おわりに

タートル・ディスプレイ言語を作ってみて，これだけのコマンド（追加コマンド5つ）で，使い方によってはとても大きなことができるような気がする。ただ，1文字のコマンドという制約から，プログラムの作成には少し慣れないと大変かも知れない。いままで何人かの人に使ってもらい，改良してきたが，もっときめをこま

かくするにはR（回転角）を1度きざみにする必要もあり，処理スピードも上げたい。

Fのキーの上に，“行く”，Rに“まわる”など付けておけば，小学生でも充分使えるのではないか！ なにしろキーを押せば結果がすぐ画面に出てくるので，コンピューターという意識をせず，電子黒板の役割ができそうだ。絵をかかせるつもりなら，案外，奇抜で，独創的なものがでてくるかも……。

そういう点から，最近，話題の“Logo”と共通する点が多々ある。

この言語は手近にあるif 800/30（図示表示8色カラー，640×400ドット）というパソコンのBASICで書いた。BASICの記述なので汎用性があり，マイクロソフト系のパソコンなら移植は比較的簡単に行くと思う。実際にif 800/20とFM-8，FM-7に移した（基本部分）ときも数箇所の変更ですんだ（ソース・プログラム必要な方は申し出て下さい）。

最後に資料を提供していただいた東京大学（大型計算機センター）石田晴久先生，（理学部）小野芳彦先生，その他助言をいただいた法政大学（工学部）駒木悠二先生，（計算センター）所長 大地羊三先生，所員の方々に感謝致します。

参考文献

1) Lichen Wang : An Interactive programming Language for control of robots, Dr. Dobb's Jaurnal, Vol. 2, Issue 10, p. 60～63 (1977).
2) 石田晴久：ドラゴン曲線やヒルベルト曲線を描くマイコン，bit, Vol. 10, No. 7, p. 35～40.
3) 小野芳彦：Wangのロボット言語インタプリタ（8080リスト），bit, Vol. 10, No. 7, p. 41～50.
4) マーチン・ガードナー著（一松信訳）：数学ゲームI 日本経済新聞社
5) マーチン・ガードナー著（一松信訳）：続々数学魔法館 東京図書
6) S．パパート著（奥村貴世子訳）：マインド・ストーム 未来社
7) パパート＆ソロモン（奥村訳）：コンピュータを使って学ぶ（1），bit, Vol. 5, No. 3, p. 15～22.
パパート＆ソロモン（奥村訳）：コンピュータを使って学ぶ（2），bit, Vol. 5, No. 4, p. 16～21.
8) 石田則道：ロボット言語で怪物曲線を描こう，bit, Vol. 15, No. 10, p. 4～11.

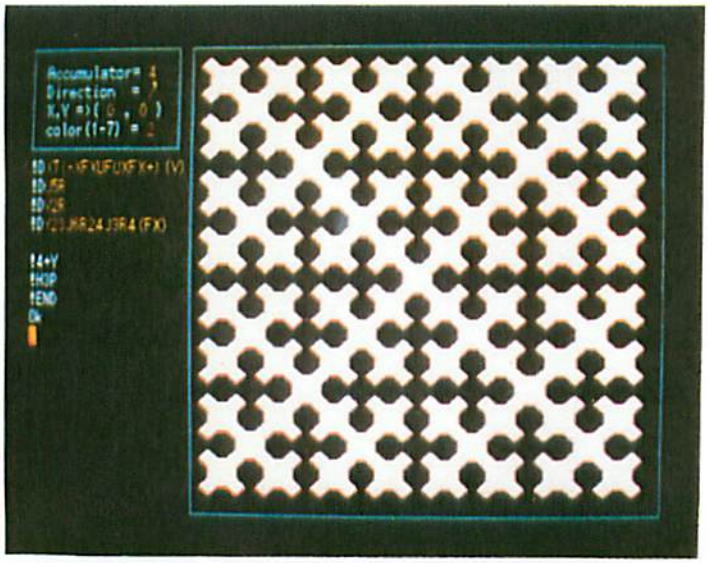


図27　4次シエルピンスキー曲線

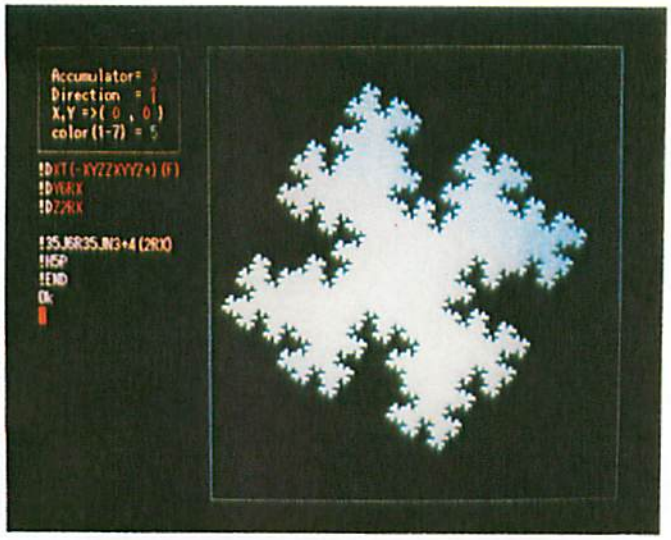


図30　雪片曲線３段(分割数120の場合)

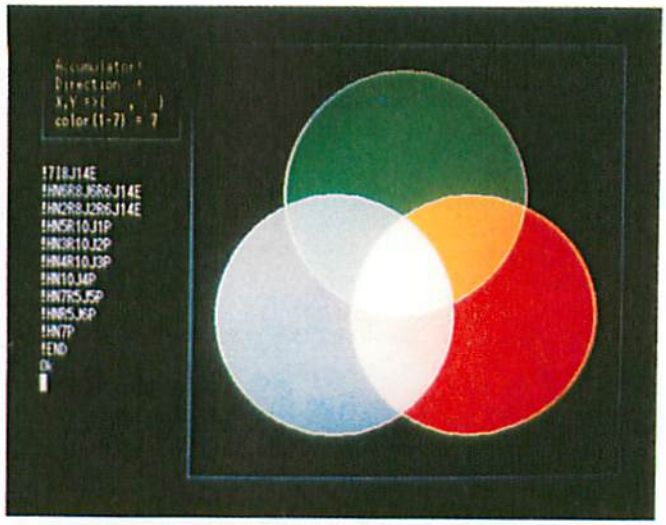


図33　三原色の組み合せ