

フラクタル幾何学の造形美 (I , II)

— マンデルブロ集合と自己平方ドラゴン —

写真学科 写真科学研究室

米 川 靖 彦

Art of Fractal Geometry (Part 1 & 2)
 —Mandelbrot Set & Selfsquare Dragon—
 Photographic Science dept. of Photography
 Yasuhiko YONEKAWA

PART(I) マンデルブロ集合

I-1 はじめに

フラクタルという概念は10数年前にマンデルブロによって創り出されたものである¹⁾。しかし、このフラクタルの基礎は確立しておらず、厳密な定義はまだ統一されていないが、フラクタルとは特徴的な長さを持たない図形、構造、現象等の総称である²⁾。

このフラクタルの語源はラテン語であり、物が壊れて不規則な破片になった状態を示しており、ガラスの小さな破片や大きな破片がたくさん集まった状態を想像すれば大きな誤解は生じない。自然界の中にはフラクタル現象を示すものが非常に多い。雲の形、川の形態、海岸線の形、肺や血管の構造、星の分布、土星の輪、個体表面、凝集体、放電パターン、乱流、分子のスペクトル、株価の変動等がこのフラクタルであることが知られている。マンデルブロが言っているフラクタル幾何学はコンピュータにおうところが非常に大きい。極言すればコンピュータがなければ発達しなかったともいえる。このコンピュータによる画像の美しさには驚かされることが多い。

複素平面の簡単な写像が非常に複雑で美しいフラクタル図形を生じる例としてはマンデルブロ集合、自己平方ドラゴン(マンデルブロの命名)、ジュリア集合等がある。

ここではマンデルブロ集合について述べる。

I-2 複素数の取り扱い

I-2-1 複素平面

複素数は実数部と虚数部とに分けられる。

$$z = \frac{a}{\text{実数部}} + \frac{bi}{\text{虚数部}} \dots\dots\dots(1)$$

虚数部には i を付けて表記する。 i には 2 乗すると -1 になる性質がある。(1)式の実数部と虚数部を 2次元平面の X 軸に実数部をあて、 Y 軸に虚数部をあてると z の点の位置が決定する。このように実数軸と虚数軸で表記する平面を複素平面という。よって全ての複素数は 2次元平面に表記できることになる。

I-2-2 複素数の 4 則演算

複素数の 4 則演算は以下の方法で行なう。

(1) 加算

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

(2) 減算

$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

(3) 乗算

$$(a + bi)(c + di) = (ac - bd) + (ad + bc)i$$

(4) 除算

$z = a + bi$ の共役複素数 $(a - bi)$ を考え、 \bar{z} とすると、

$$z\bar{z} = (a+bi) * (a-bi) = a^2 + b^2 = |z|^2$$

そこで

$$\frac{1}{z} = \frac{1}{z\bar{z}} = \frac{a-bi}{a^2+b^2}$$

であるから例えば、

$$\frac{1+z}{\bar{z}} = \frac{(1+z)\bar{z}}{z\bar{z}} = \frac{a^2+b^2+a-bi}{a^2+b^2}$$

のように計算する。

I-3 マンデルブロ集合

マンデルブロ集合は IBM のマンデルブロによって作図されたもので、式としては

$$f(z) = z^2 + c$$

を考える。

ここで、 z は複素変数で、 c は複素定数である。 z の初期値は常に 0 である。

暫化式を

$$z_{n+1} = f(z_n)$$

とする。

n を無限大にしても $\{z_n\}$ が無限大にならないような集合をマンデルブロ集合と言う。

Fig. I-1 はマンデルブロ集合全体を示している。

中心部の部分がマンデルブロ集合を示している。この集合は非常にきれいな集合であるが、周辺部を拡大して見ると小さな集合が隣接していることが分かる。また、腕のように伸びた部分の中にも所々に集合が存在している。視点を変化させることにより、非常に多様な形状を取り出せる。図形の中には自己相似形が隠されているが、注意して見ないと見のがしてしまう。以下に多様なマンデルブロ集合の図形 (Fig.) と位置データ (Table I-1) を示す。

I-4 コンピュータプログラム

上記のような図形を描くにはコンピュータが必要であるが、筆者はコンピュータ本体には PC-9801VX の CPU 80286 と数値演算コプロセッサ 80287 を使用した。初期には東海大学短期大学部講師の淵上季代絵氏の『フラクタル CG コレクション』のプログラム³⁾ を MS-DOS 上でコンパイルして描画される方法をとっていた。しかし描画に非常に時間がかかったため、京都大学教養部助知授の宇敷重宏氏の『フラクタルの世界』に掲載されている MS-DOS 上の TURBO PASCAL プログラム⁴⁾ を使用して描画させた。

マンデルブロ集合は z の初期値が常に 0 から出発するために非常に計算時間がかかる。List I-1

Table I-1 図版のデータ

	X_{min}	X_{max}	Y_{min}	Y_{max}
Fig. I-1	-2.3	0.5	-1.5	1.5
Fig. I-2	-1.2551	-1.2529	0.0237	0.0257
Fig. I-3	-0.2	-0.05	-0.95	-0.8
Fig. I-4	-0.115	-0.095	-0.93	-0.91
Fig. I-5	-0.115	-0.114	-0.929	-0.928
Fig. I-6	-1.255	-1.2545	0.0252	0.0257
Fig. I-7	-1.255	-1.2545	0.0252	0.0257
Fig. I-8	0.26219	0.26271	-0.0025	-0.0020
Fig. I-9	-0.75104	-0.74080	0.10511	0.11536
Fig. I-10	-0.74758	-0.74624	0.10671	0.10779
Fig. I-11	-0.17	-0.15	-1.043	-1.023
Fig. I-12	0.26	0.262	-0.0028	-0.0008

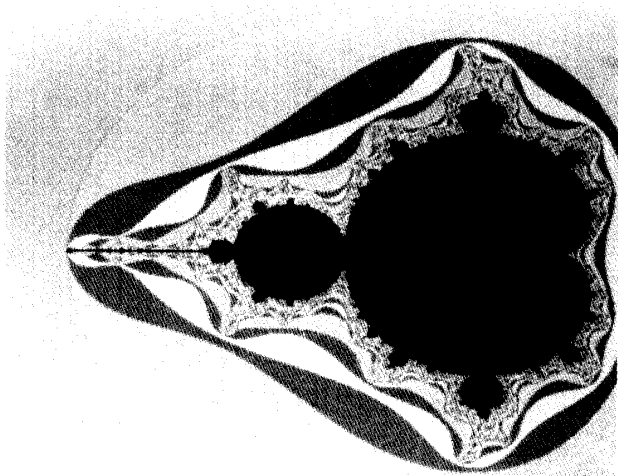


Fig. I-1 $X: -2.3 \sim 0.5$
 $Y: -1.5 \sim 1.5$

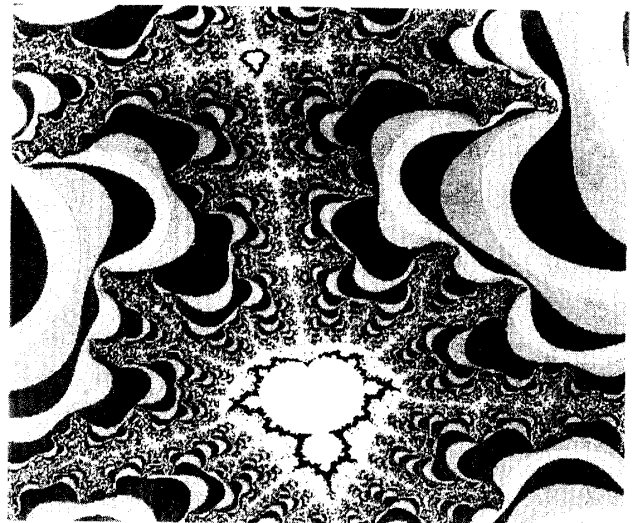


Fig. I-4 $X: -0.155 \sim 0.095$
 $Y: -0.93 \sim -0.91$

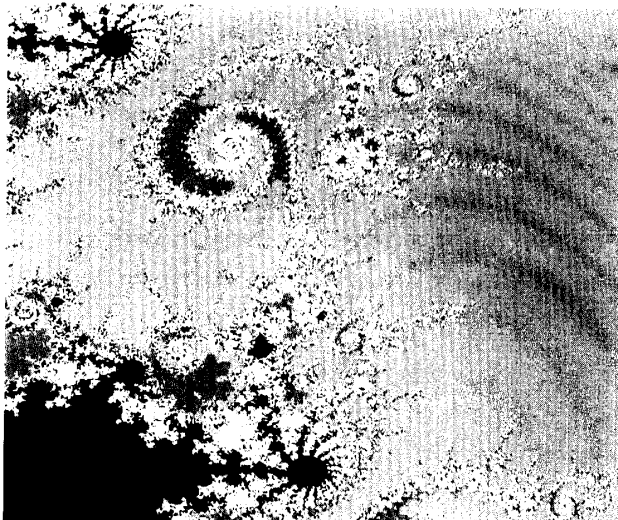


Fig. I-2 $X: -1.2551 \sim -1.2529$
 $Y: 0.02371 \sim 0.0257$

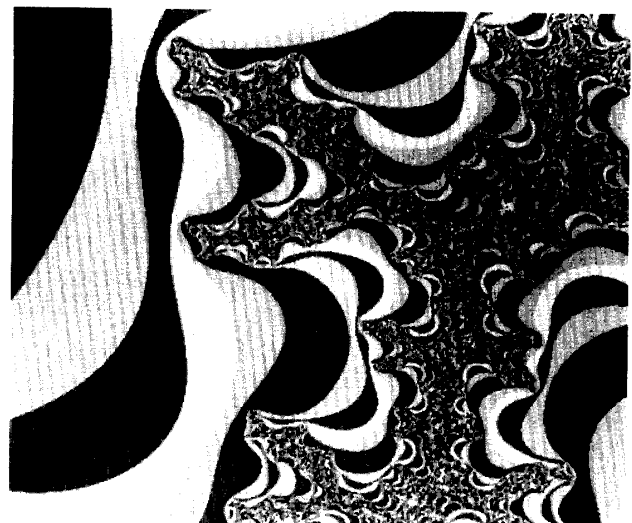


Fig. I-5 $X: -0.115 \sim -0.114$
 $Y: -0.929 \sim -0.928$



Fig. I-3 $X: -0.2 \sim -0.05$
 $Y: -0.95 \sim -0.8$

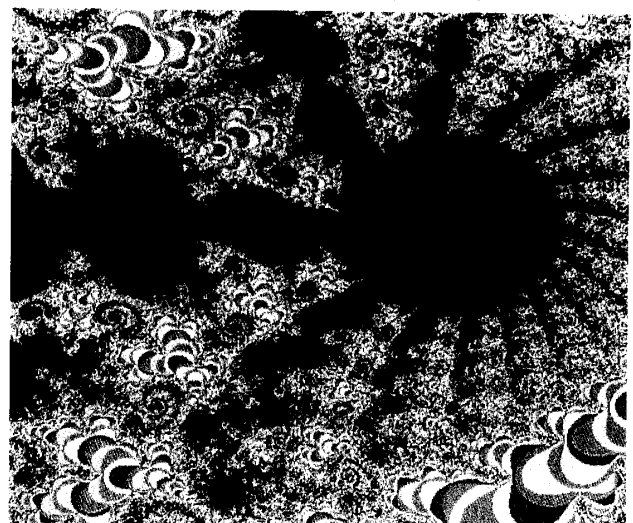


Fig. I-6 $X: -1.255 \sim -1.2545$
 $Y: 0.252 \sim 0.0257$

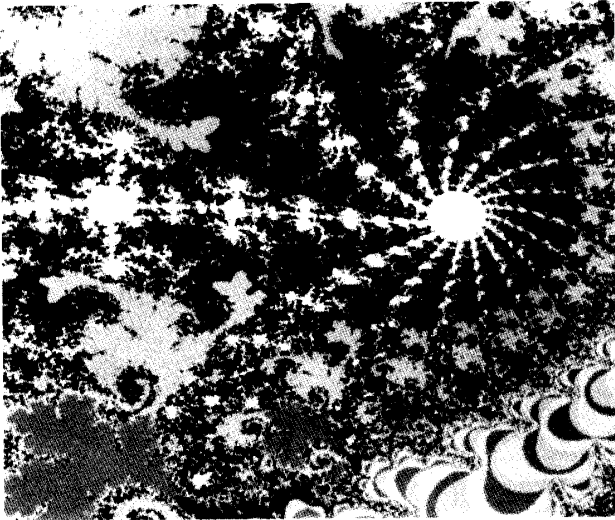


Fig. I-7 $X : -1.255 \sim -1.2545$
 $Y : 0.252 \sim 0.0257$

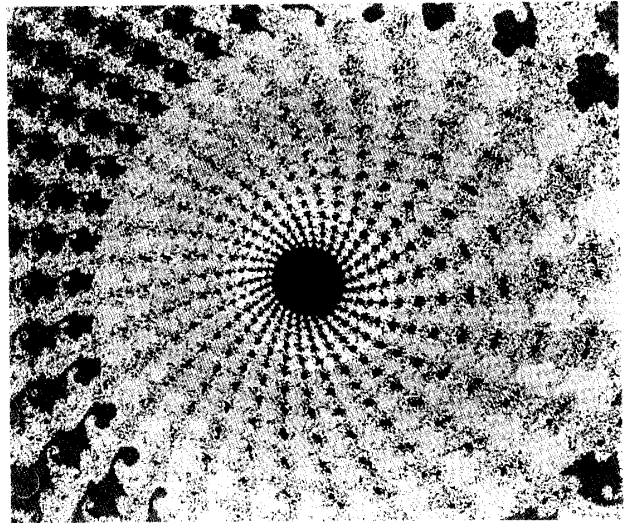


Fig. I-10 $X : -0.74758 \sim -0.74624$
 $Y : 0.10671 \sim 0.10779$

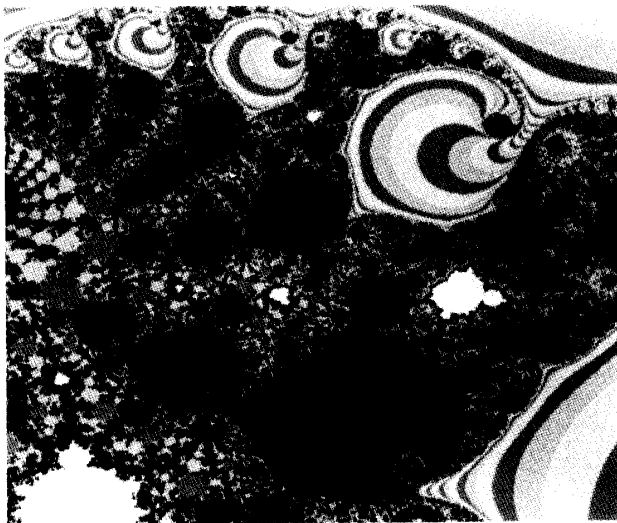


Fig. I-8 $X : 0.26219 \sim 0.26271$
 $Y : -0.0025 \sim -0.002$

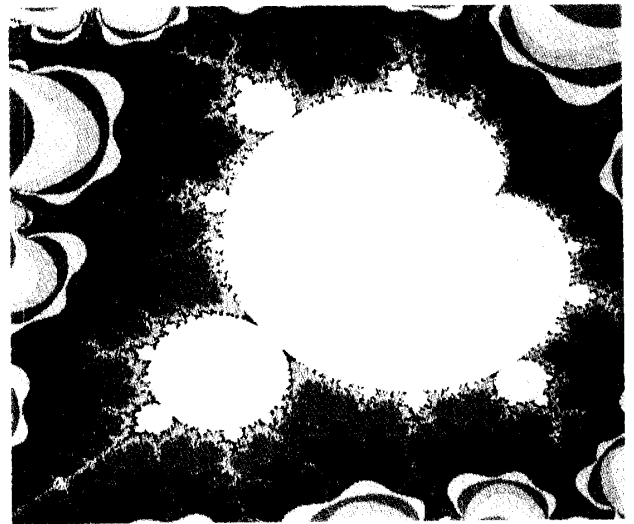


Fig. I-11 $X : -0.17 \sim -0.15$
 $Y : -1.043 \sim -1.023$

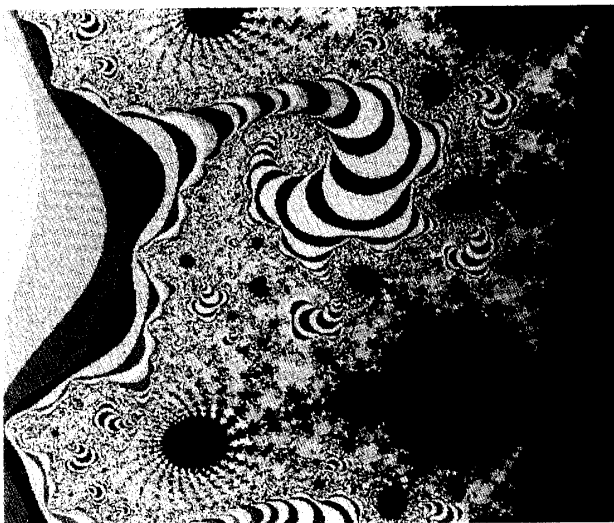


Fig. I-9 $X : -0.75104 \sim -0.748$
 $Y : 0.10511 \sim 0.11536$

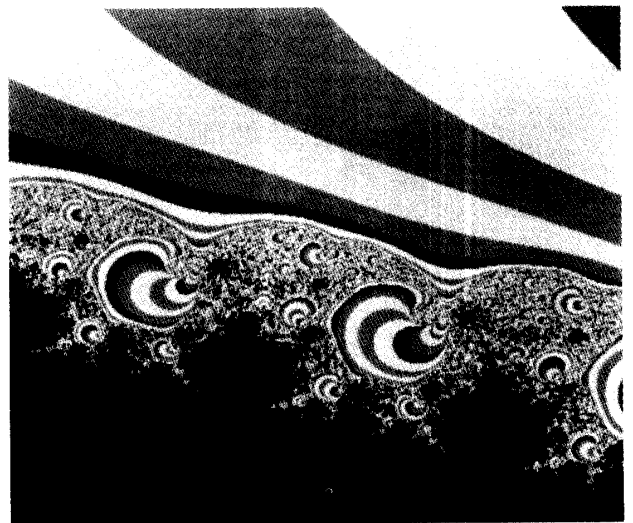


Fig. I-12 $X : 0.26 \sim 0.262$
 $Y : -0.0028 \sim -0.0008$

の BASIC プログラムでは Fig. I-1 の図形では時間が約 6 時間位かかるが、TURBO PASCAL では約 2 時間位で描画できる。プログラムリストを以下に示す。

List I-2 の TURBO PASCAL のプログラムには PART (II) と共通に使用する List II-2 のプロシージャ部の GRLIBPAS が必要である。ディスクにセーブした画像データを表示するには List II-3 の GROUT. PAS が必要である。

最大計算回数 (IMAX) を多くすると図形は複雑になるが、描画時間が非常にかかるようになる。プログラムを簡単に説明すると計算回数 (K) の値により色を変化させているが、白黒の写真でしか示せないのが残念なほどカラフルな図形が出来る上がる。

Fig. I-1 の最大ループ (IMAX) は 50 回であるが、Fig. I-10 の IMAX は 1,000 回ほどである。

Fig. I-10 での BASIC の場合、高速になるコン

```

100 ' マンデルブロ集合
110 '
120 ' 初期値設定
130 '
140 CONSOLE 0, 25, 0, 1
150 SCREEN 3 : CLS 3
160 IMAX=50
170 TATE=400 : YOKO=500
180 XL=-2.3 : XU=.5
190 YL=-1.3 : YU=1.3
200 WINDOW (XL, -YU) - (XU, -YL)
210 VIEW (70, 0) - (569, 399)
220 DR=(XU-XL)/YOKO
230 DI=(YU-YL)/TATE
240 FOR XX=XL TO XU STEP DR
250   FOR YY=YL TO YU STEP DI
260     ZR=0 : ZI=0
270     KA=0
280     *LOOP
290     KA=KA+1
300     IF KA>IMAX THEN *NE
310     ZZR=ZR^2-ZI^2+DR
320     ZZI=2*ZR*ZI+DI
330     IF (ZZR^2+ZZI^2)>4 THEN *DISP
340     ZR=ZZR : ZI=ZZI
350     GOTO *LOOP
360     *DISP
370     C=KA MOD 8
380     PSET (XX, -YY), C
390     *NE
400   NEXT YY
410 NEXT XX
420 *MACHI
430 A$=INKEY$
440 IF A$="" THEN *MACHI
450 END

```

List I-1

BASICによる
マンデルブロ集合プログラム

```

Program Mandelbrot;

{$I GRLIB.PAS}

Type Complex = Record re,im :Real end;

var Wmin,Wmax,Wd,a,x0,x1,x2,x3,z1,z2,XNC : Complex;
    rr : Real;
    lcc : Integer;
    Ch1 : Char;
    FileNmRad,TodayDate : String14;

Procedure WindowSet;
begin
  GrClr;
  Writeln('Region');
  Writeln(' Ax : ',Wmin.re:10:6,' : ',Wmax.re:10:6);
  Writeln(' Ay : ',Wmin.im:10:6,' : ',Wmax.im:10:6);
  Write(' AXmin = '); Readln(Wmin.re);
  Write(' AXmax = '); Readln(Wmax.re);
  Write(' AYmin = '); Readln(Wmin.im);
  Write(' AYmax = '); Readln(Wmax.im);
  Wd.re := 1.0 / (Wmax.re -Wmin.re);
  Wd.im := 1.0 / (Wmax.im -Wmin.im);
  ClrScr;
end;

Procedure FileAssign;
begin
  Write(' FileNameRadical = '); Readln(FileNmRad);
  Write(' TodayDate = 1988/');Readln(TodayDate);
  ClrScr;
end;

Procedure MandelbrotPicture;
var lx,ly,lcol,IYT,IXT,K,IMAX :Integer;
    Axd,Ayd,EPSI,zz : Real;
Label PA;

Procedure CheckInterruption;
begin
  if KeyPressed then
    begin
      read(Kbd,Ch1); if Ch1 = 'S' then begin GrStop; Beep; Halt; end;
    end;
end;

Begin
  IMAX := 1000;EPSI := 4.0;
  Writeln('Mandelbrot'); Writeln('set');
  Writeln;Writeln;Writeln('Region');
  Writeln(' Ax : ');Writeln(Wmin.re:10:6);Writeln(Wmax.re:10:6);
  Writeln(' Ay : ');Writeln(Wmin.im:10:6);Writeln(Wmax.im:10:6);
  GotoXY(1,23);Writeln('(C) s.Ushiki'); Write(' 1988/',TodayDate);
  GotoXY(1,15);
  Axd := (Wmax.re -Wmin.re)/500; Ayd := (Wmax.im - Wmin.im)/400;
  for IYT := 1 to 400 do
    begin
      a.im := Wmin.im + Ayd*IYT; IY := 400 - IYT;
      for IXT := 0 to 500 do
        begin
          IX := 100+IXT;

```

```

Icol := 7; Pset(Ix,Iy,Icol);
a.re := Wmin.re + Axd * IXT; X0 := a;
for K := 1 to Imax do
begin
  x2.re := Sqr(X0.re) - Sqr(X0.im) + a.re;
  X0.im := 2.0 * X0.re * X0.im + a.im; X0.re := X2.re;
  ZZ := Sqr(X0.re) + Sqr(X0.im);
  if ZZ > EPSI then
    begin
      ( diverged case )
      if K <= 180 then
        begin
          ICOL := 1 ;goto PA;
        end;
      if K <= 240 then
        begin
          ICOL := 4 ; goto PA;
        end;
      if K <= 300 then
        begin
          ICOL := 3 ;goto PA;
        end;
      if k <= 450 then
        begin
          icol := 6 ;goto PA;
        end;
      if k <= 600 then
        begin
          icol := 5 ;goto PA;
        end;
      if Icol <= Imax then
        begin
          ICOL := 2 ;goto PA;
        end;
    end;
  end;
  ( chaotic case )
  Icol := 0;
PA:  Pset(IX,IY,ICOL);
  end;
  CheckInterruption;
end;
end;

begin
  GrInit; GrClr; WindowSet; FileAssign; MandelbrotPicture;
  GrSave(FileNmRad); GrStop;
end.

```

List I-2

TURBO PASCAL (ver 3. 0) による
マンデルブロ集合プログラム
(*** .PAS でセーブする)

パイラを使用してもループ回数が非常に大きいので、描画時間は96時間もかかってしまった。

TURBOPASCAL のプログラムによる場合は数値演算コプロセッサを効率良く計算に使用出来、描画出来るようになった。

また画像データをディスクに記録すること

により画像の再表示が早く出来るようになった。

I-5 おわりに

フラクタルという聞きなれない言葉を耳にしてから数年がたつ。マンデルブロのフラクタル幾何学が1985年に広中平祐氏の訳により発行¹⁾されてから、いかにすればあのような図形が描かれるの

かを考え続けてきた。最近になっていろいろな本²⁾³⁾⁴⁾⁵⁾が発行されてきて、なんとか自分なりに消化できるようになった。しかし、筆者が使用しているパソコンの BASIC の限界(計算時間の限界)が見えて来た。

また、出来たフラクタル画像を記録する場合、色々な方法があるが、CRT 画面を直接撮影するとフィルム上にグレアが写し込まれてしまう。高解像度の CRT になればなるほどこのグレアが出易いとのことで、一時は途方にくれていたがなんとか解決出来たみたいだ。しかし、全く消えた訳ではなく全体に広がり、色が薄まっただけみたいだ。

本年になって、TURBO PASCAL の新バージョンがマイクロソフトアソシエイツより発売された。旧バージョンよりもコンパイル速度が早くなり、数値演算プロセッサの組み込みが簡単になり、取り扱える実数の大きさも大きくなったとのことで早速購入したが、旧バージョンとはグラフィックのユニットの取り扱い、画面制御、DOS への割り込みなどが多少違っていた。しかし、GRAPH, CRT のユニットが充実したためにプログラムの長さは短くなった。また、プログラムの作成は短時間で出来るようになったと思う。

現在は新バージョンの TURBO PASCAL でプログラムを組み、新しいプログラムで描画させているが、List には掲載していない。

PART(II) 自己平方ドラゴン

II-1 はじめに

PART (I) ではマンデルブロ集合について述べてきたが、ここでは自己平方ドラゴンについて述べることにする。

自己平方ドラゴンとはマンデルブロによって命名されたものである(マンデルブロはこのような命名が好きである)。自己平方ドラゴンとは境界線にジュリア集合を持った非常に複雑怪奇な図形である。

この自己平方ドラゴンの図形はマンデルブロ集合の場合と同じように自己相似形がいたるところに

隠されている。

複素数の取り扱いについては PART (I) と同じなのでここでは省くことにする。

II-2 自己平方ドラゴン

マンデルブロ集合の時と同じ方程式である $f(z)=z^2+c$ を考える。前回と同様に z は複素変数とし、 c は複素定数とする。暫化式 $z_{n+1}=f(z_n)$ も前回と同様である。しかし、 z に対する初期値が前は 0 からであったが、今回は入力値である。また、複素定数である c にも入力した数値を代入する。この様にして計算を繰り返し、マンデルブロ集合の時と同じような方法でコンピュータで描画させると、ジュリア集合を境界線とした非常に複雑な図形が出来上がる。これをマンデルブロは自己平方ドラゴンと呼んでいる。この自己平方ドラゴンは複素定数である c の値によって劇的な変化をすることが知られている。自己相似な図形が連続として続いており、いくら拡大しても相似な図形が出て来る。入力する c の値を極く僅かづつ変化させていくと、出来上がる図形も少しずつ変化をして行く。

このようにこの自己平方ドラゴンの図形は試行錯誤によって選択をした。出来た図形を下に示し、データも Table II-1 にして示す。Fig. II-1 の拡大が Fig. II-2 である。また、Fig. II-5 の左下に半分見えている渦巻きの拡大が Fig. II-6 である。このように図形を拡大して見れば相似な図形があちこちに存在する。

II-3 コンピュータプログラム

自己平方ドラゴンのプログラムはマンデルブロ集合のプログラムを変更すれば良い。だからここでは宇敷重宏氏のプログラム⁴⁾に変更を加えて描画させている。

List II-1 がマンデルブロ集合に変更を加えたメインプログラムである。プログラムを検討すれば分かると思うが、 z の初期値が入力値になり、 c にも入力値を代入している。このようにプログラムの変更を行なった。

前報にも書いたが List II-2 は前報と共通に使用できるプロシージャー(描画プログラムとディ

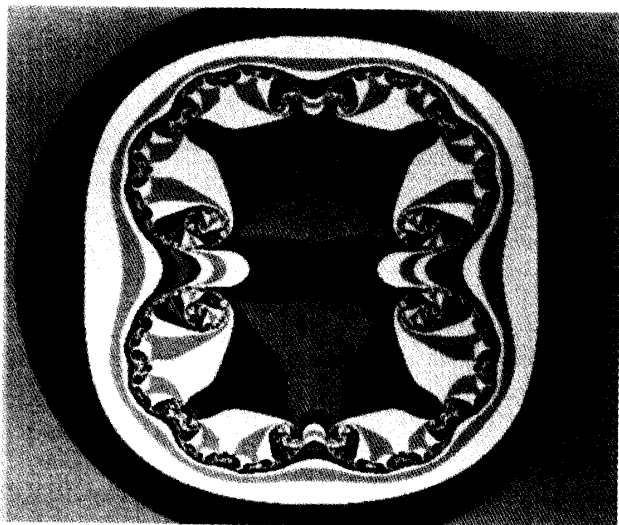


Fig. II-1 $X: -1.5 \sim 1.5$
 $Y: -1.5 \sim 1.5$
 $C: 0.3 + 0.0i$

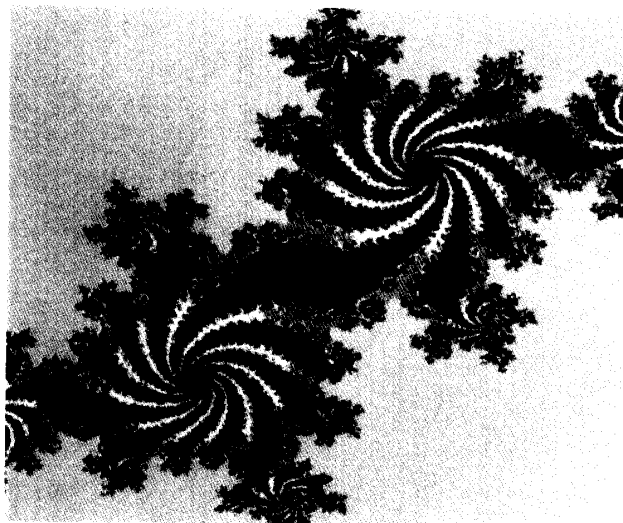


Fig. II-3 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.3 \sim 0.63i$

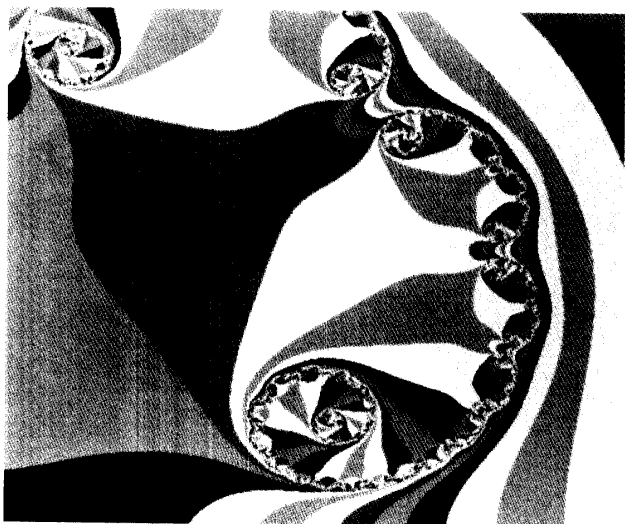


Fig. II-2 $X: 0.0 \sim 1.0$
 $Y: 0.0 \sim 1.0$
 $C: 0.3 + 0.0i$

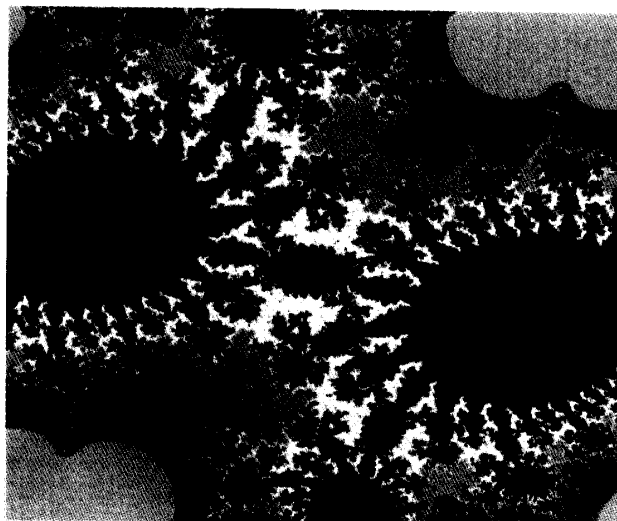


Fig. II-4 $X: -0.3 \sim 0.3$
 $Y: -0.3 \sim 0.3$
 $C: -1.24 + 0.075i$

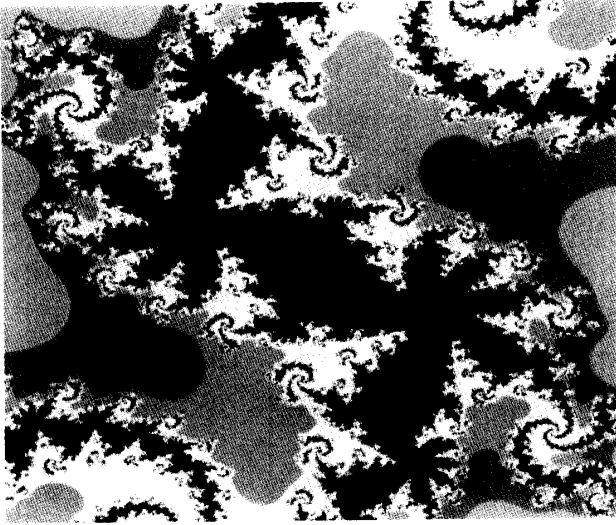


Fig. II-5 $X: -0.5 \sim 0.5$
 $Y: -0.5 \sim 0.5$
 $C: -0.04 \sim 0.695i$

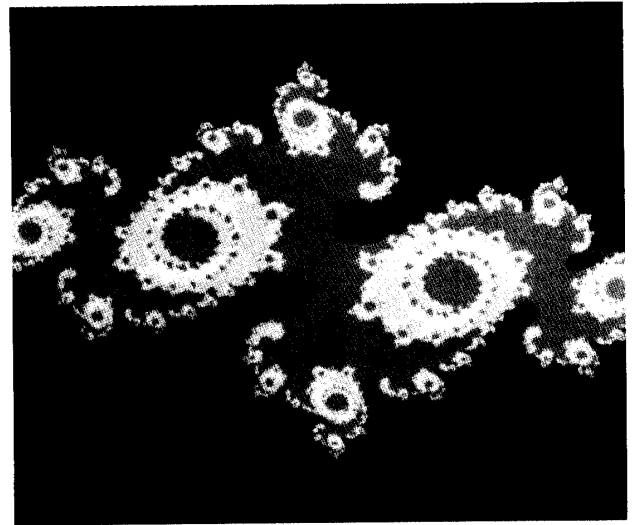


Fig. II-7 $X: -1.3 \sim 1.3$
 $Y: -1.3 \sim 1.3$
 $C: -0.75 \sim 0.2i$

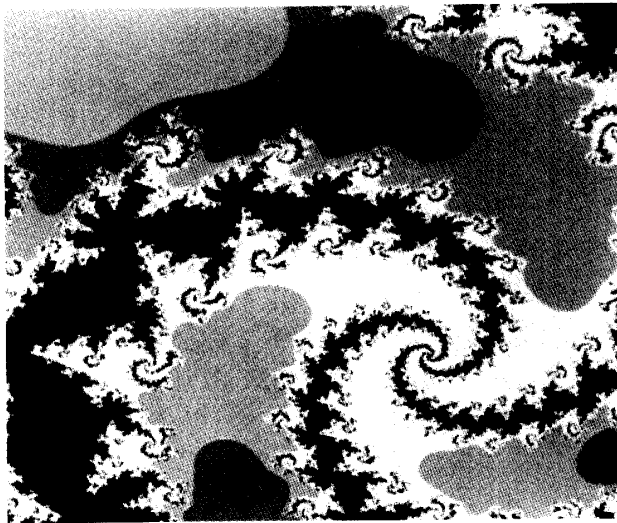


Fig. II-6 $X: -0.7 \sim 0.0$
 $Y: -0.7 \sim 0.0$
 $C: -0.04 \sim 0.695i$

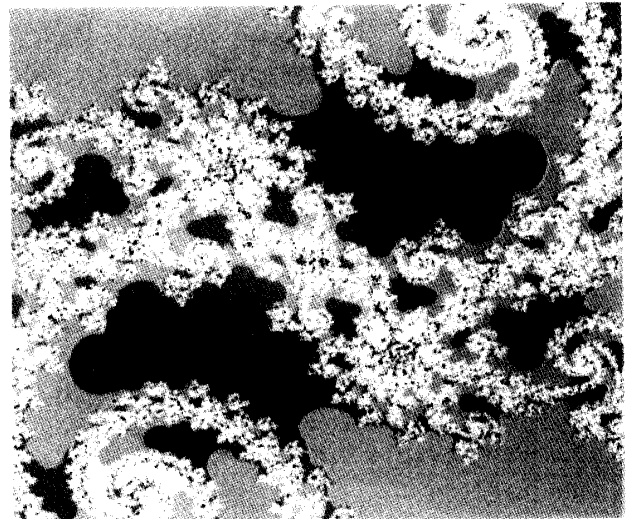


Fig. II-8 $X: -0.55 \sim 0.55$
 $Y: -0.52 \sim 0.52$
 $C: -0.2 \sim 0.675i$

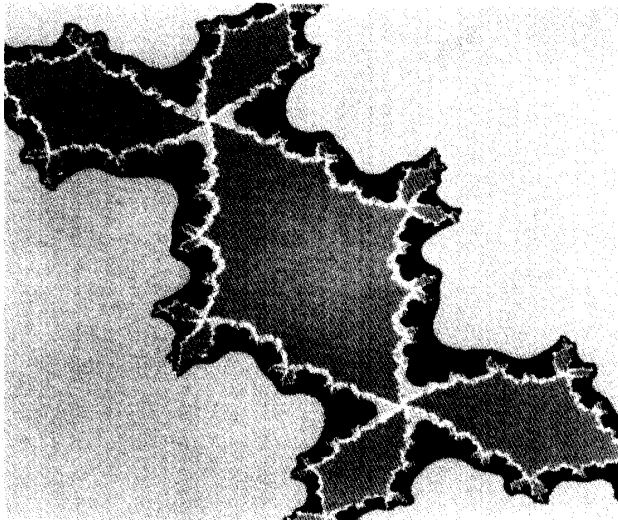


Fig. II-9 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.1 + 0.795i$

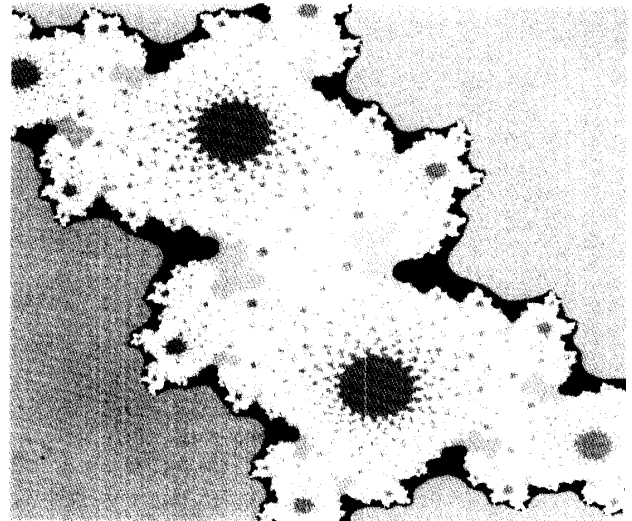


Fig. II-11 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.035 + 0.65i$

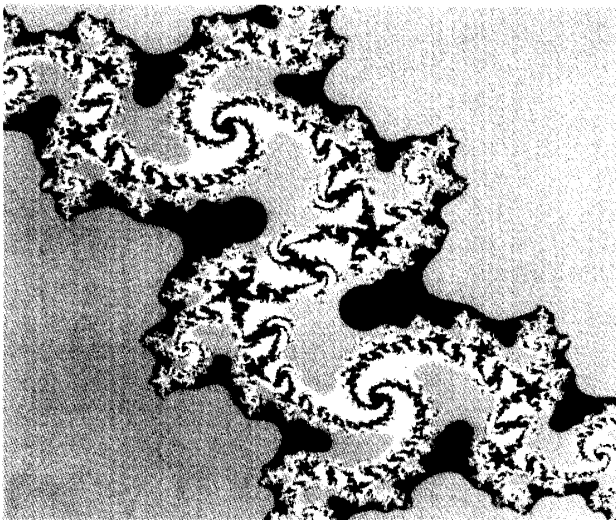


Fig. II-10 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.02 - 0.695i$

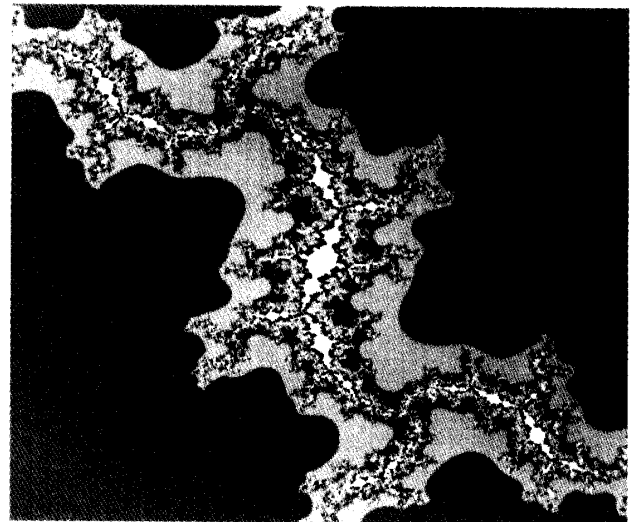


Fig. II-12 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.02 - 0.795i$

```

Program SelfSquareDragon;

{$I GRLIB.PAS}

Type Complex = Record re,im :Real end;

var Wmin,Wmax,Wd,a,x0,x1,x2,x3,z1,z2,XNC,AA : Complex;
    rr : Real;
    lcc : Integer;
    Ch1 : Char;
    FileNmRad,TodayDate : String14;

Procedure WindowSet;
begin
  GrClr;
  Writeln('Region');
  Writeln(' Ax : ',Wmin.re:10:6,' : ',Wmax.re:10:6);
  Writeln(' Ay : ',Wmin.im:10:6,' : ',Wmax.im:10:6);
  Writeln(' AA : ',AA.re:10:6,' : ',AA.im:10:6);
  Write(' AXmin = '); Readln(Wmin.re);
  Write(' AXmax = '); Readln(Wmax.re);
  Write(' AYmin = '); Readln(Wmin.im);
  Write(' AYmax = '); Readln(Wmax.im);
  Write(' AA.re = '); Readln(AA.re);
  Write(' AA.im = '); Readln(AA.im);
  Wd.re := 1.0 / (Wmax.re -Wmin.re);
  Wd.im := 1.0 / (Wmax.im -Wmin.im);
  ClrScr;
end;

Procedure FileAssign;
begin
  Write(' FileNameRadical = '); Readln(FileNmRad);
  Write(' TodayDate = 1988/'); Readln(TodayDate);
  ClrScr;
end;

Procedure SelfSquareDragon;
var Ix,Iy,Icol,IYT,IXT,K,IMAX :Integer;
    Axd,Ayd,EPSI,zz : Real;
Label PA;

Procedure CheckInterruption;
begin
  if KeyPressed then
    begin
      read(Kbd,Ch1); if Ch1 = 'S' then begin GrStop; Beep; Halt; end;
    end;
end;

Begin
  IMAX := 150 ; EPSI := 4.0;
  Writeln('SelfDragon'); Writeln('set');
  Writeln;Writeln;Writeln('Region');
  Writeln(' Ax : ');Writeln(Wmin.re:10:6);Writeln(Wmax.re:10:6);
  Writeln(' Ay : ');Writeln(Wmin.im:10:6);Writeln(Wmax.im:10:6);
  Writeln(' A : ');Writeln(AA.re:10:6) ;Writeln(AA.im:10:6);
  GotoXY(1,23);Writeln('(C) s.Ushiki'); Write(' 1988/',TodayDate);
  GotoXY(1,15);
  Axd := (Wmax.re -Wmin.re)/500; Ayd := (Wmax.im - Wmin.im)/400;
  for IYT := 1 to 400 do

```

```

begin
  a.im := Wmin.im + Ayd*IYT; IY := 400 - IYT;
  for IXT := 0 to 500 do
    begin
      IX := 100+IXT;
      Icol := 7; Pset(Ix,Iy,Icol);
      a.re := Wmin.re + Axd * IXT; X0 := a;
      for K := 1 to Imax do
        begin
          x2.re := Sqr(X0.re) - Sqr(X0.im) + AA.re;
          X0.im := 2.0 * X0.re * X0.im + AA.im; X0.re := X2.re;
          ZZ := Sqr(X0.re) + Sqr(X0.im);
          if ZZ > EPSI then
            begin
              { diverged case }
              if k <= 8 then
                begin
                  icol := 4 ;goto PA;
                end;
              if k <= 15 then
                begin
                  icol := 2 ;goto PA;
                end;
              if K <= 25 then
                begin
                  ICOL := 1 ;goto PA;
                end;
              if k <= 35 then
                begin
                  ICOL := 6 ;goto PA;
                end;
              if k <= 45 then
                begin
                  ICOL := 5 ;goto PA;
                end;
              if k <= imax then
                begin
                  ICOL := 0 ;goto PA;
                end;
            end;
          end;
          { chaotic case }
          Icol := 7;
        PA: Pset(IX,IY,ICOL);
        end;
        CheckInterruption;
      end;
    end;
  end;

begin
  GrInit; GrClr; WindowSet; FileAssign; SelfSquareDragon;
  GrSave(FileNmRad); GrStop;
end.

```

List II-1

TURBO PASCAL (ver 3. 0) による
自己平方ドラゴンプログラム
(*** .PAS でセーブする)

```

{ This file is for PC-9801. }
Const SegBase1 = $A800; SegBase2 = $B000; SegBase3 = $B800;

Procedure Beep;
begin
  InLine($b4/$17/$cd/$18); Delay(50); InLine($b4/$18/$cd/$18);
end;

Procedure GrInit;
begin
  InLine($b4/$40/$cd/$18/$b4/$42/$b5/$c0/$cd/$18);
  Write('^['[>1h'^['*'^['[>5h''');
end;

Procedure GrStop;
begin
  Beep;
  Write('^['[>5l'^['[>1l'''); InLine($b4/$41/$cd/$18);
end;

Procedure Gclr;
begin
  FillChar(Mem[SegBase1:0],32000,Chr(0));
  FillChar(Mem[SegBase2:0],32000,Chr(0));
  FillChar(Mem[SegBase3:0],32000,Chr(0));
end;

Procedure GrClr; begin ClrScr; Gclr; end;

Procedure Wait;
var Cl : Char;
begin Repeat until KeyPressed; Read(Kbd,cl); end;

Procedure Pset(Ix,Iy,Icol : Integer);
begin { This is PC-9801 version. }
  InLine( $8B/$46/$06/$3D/$00/$00/$7C/$46/$3D/$8F/$01/$7F/
    $41/$BB/$50/$00/$F7/$E3/$8B/$5E/$08/$83/$FB/$00/$7C/$34/
    $81/$FB/$7F/$02/$7F/$2E/$BA/$07/$00/$23/$D3/$B1/$03/$D3/
    $EB/$03/$C3/$8b/$F8/$8B/$CA/$BA/$80/$80/$D3/$EA/$80/$F6/
    $FF/$8B/$5E/$04/$B8/$00/$A8/$8E/$C0/$B9/$00/$08/$F7/$C3/
    $04/$00/$74/$09/$26/$08/$15/$EB/$07/$90/$EB/$2A/$90/$26/
    $20/$35/$03/$C1/$8E/$C0/$F7/$C3/$02/$00/$74/$06/$26/$08/
    $15/$EB/$04/$90/$26/$20/$35/$03/$C1/$8E/$C0/$F7/$C3/$01/
    $00/$74/$06/$26/$08/$15/$EB/$04/$90/$26/$20/$35);
end;

Type String14 = String[14];

Procedure GrLoad(var FNameRad : String14);
var FName : String14; GrFile : File;
begin
  GrClr;
  FName := concat(FNameRad, '.IMG');
  Assign(GrFile,Fname); Reset(GrFile);
  BlockRead(GrFile,Mem[SegBase1:0],255);
  BlockRead(GrFile,Mem[SegBase2:0],255);
  BlockRead(GrFile,Mem[SegBase3:0],255);
  close(GrFile);
end;

Procedure GrText(IXS,IYS,IXE,IYE : Integer);

```

```

var Register : Record AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer end;
  FontBuffer : Array[0..17] of Byte;
  IXN,IYN,ICN,ICN1,ICN2,ICN3 : Integer;
begin
  for IYN := IYS to IYE do
  begin
    for IXN := IXS to IXE do
    begin
      ICN := Mem[$a000:IYN * 160 + IXN * 2];
      if ICN <> Ord(' ') then
      begin
        with Register do
        begin
          AX := $1400;BX := Seg(FontBuffer); cx := Ofs(FontBuffer);
          DX := $8000 or ICN;
        end;
        Intr($18,Register); ICN := 16*80*IYN + IXN;
        For ICN1 := 0 to 15 do
        begin
          ICN3 := ICN + ICN1 * 80; ICN2 := ICN1 + 2;
          Mem[SegBase1:ICN3] :=FontBuffer[ICN2];
          Mem[SegBase2:ICN3] :=FontBuffer[ICN2];
          Mem[SegBase3:ICN3] :=FontBuffer[ICN2];
        end;
      end;
    end;
  end;
end;
end;
end;

Procedure GrSave(var FNameRad : String14); (modified for PC-9801 )
var FName : String[14]; GrFile : File;
begin
  GrText(0,0,79,24);
  FName := concat(FNameRad, '.IMG');
  Assign(GrFile,FName); Rewrite(GrFile);
  BlockWrite(GrFile,Mem[SegBase1:0],255);
  BlockWrite(GrFile,Mem[SegBase2:0],255);
  BlockWrite(GrFile,Mem[SegBase3:0],255);
  close(GrFile);
end;

```

List II-2

TURBO PASCAL (ver3.0) による
データセーブ&ロードプログラム
(GRLIB.PASでセーブしておく)

```

Program GrOut;
($I GRLIB.PAS)
var FileNmRad : String14;
begin
  write('Graphic file name = ');
  Readln(FileNmRad);
  GrInit;
  if FileNmRad <> '' then GrLoad(FileNmRad);
  wait; GrStop;
end.

```

List II-3

TURBO PASCAL (ver3.0) による
再表示プログラム
(***.PASでセーブする)

スケットに画像データをセーブするプログラム)である。この List II-2 のプログラムはあらかじめ GRLIB. PAS という名前でディスクにセーブしておく必要がある。

List II-3 はディスクにセーブされている画像データを表示するプログラムであるが、これは List I-2 や II-1 と同じようにメインプログラムであるから、プログラム名にはこだわらないので、好きな名前でセーブすればよい。BASIC のプログラムは描画する時に膨大な時間を浪費するた

Table II-1 図版のデータ

	X_{min}	X_{max}	Y_{min}	Y_{max}	c の実数部	c の虚数部
Fig. II-1	-1.5	1.5	-1.5	1.5	0.3	0.0
Fig. II-2	0.0	1.0	0.0	1.0	0.3	0.0
Fig. II-3	-1.0	1.0	-1.0	1.0	-0.3	-0.63
Fig. II-4	-0.3	0.3	-0.3	0.3	-1.24	0.075
Fig. II-5	-0.5	0.5	-0.5	0.5	-0.04	-0.0695
Fig. II-6	-0.7	0.0	-0.7	0.0	-0.04	-0.695
Fig. II-7	-1.3	1.3	-1.3	1.3	-0.75	0.2
Fig. II-8	-0.55	0.55	-0.52	0.52	-0.2	-0.675
Fig. II-9	-1.0	1.0	-1.0	1.0	-0.1	0.795
Fig. II-10	-1.0	1.0	-1.0	1.0	-0.02	0.695
Fig. II-11	-1.0	1.0	-1.0	1.0	-0.035	0.65
Fig. II-12	-1.0	1.0	-1.0	1.0	-0.02	0.795

めに省いてはいるが、プログラムはそれほど難しいものではない。

II-4 おわりに

自己平方ドラゴンは c の値によって劇的に図形が変化するため c の値を決定することが非常に難しい。試行錯誤の繰り返しで思わぬ時間を費やしてしまった。しかし、出来上がった図形の複雑さや色の美しさには目を奪われてしまう。

PART (I) にも書いたが、現在では TURBO PASCAL ver4.0 の新しいバージョンでプログラムを行ない実行している。

このことにより描画時間の短縮とプログラムの簡素化が出来た。しかし、PART (I) と同様に

List については掲載していない。

参考文献

- 1) B.B. Mandelbrot, The Fractal Geometry of Nature (Freeman, New York) 広中平祐監訳, フラクタル幾何学 (日経サイエンス社 1985)
- 2) 高安秀樹, フラクタル (朝倉書店 1986)
- 3) 淵上季代絵, フラクタル CG コレクション (サイエンス社 1987)
- 4) 宇敷重宏, フラクタルの世界 (日本評論社 1987)
- 5) H.-O. Peitgen, P. H. Richter 宇敷重宏訳 フラクタルの美 (シュプリンガー・フェアラーク東京 (株) 1988)