

フラクタル幾何学の造形美 (Ⅲ)

—ジュリア集合の場合—

写真学科 写真科学研究室

米 川 靖 彦

Art of Fractal Geometry (Part 3)

—JULIA Set—

Photographic Science dept. of Photography

Yasuhiko YONEKAWA

① はじめに

前報¹⁾ではマンデルブロ集合と自己平方ドラゴンについて報告したが、今回はジュリア集合について報告する。

前回にも述べたが、自己平方ドラゴンは境界面にジュリア集合が集まっていることが知られている。

② ジュリア集合

ジュリア集合は有理関数のもとで不変な集合のひとつに対して付けられた名前である。

前報の自己平方ドラゴンはジュリア集合としても間違いではないが、マンデルブロの命名を踏襲した。しかし、今回は本来のジュリア集合という名前を用いる。

前報と同様に式① $f(z)=z^2+c$ を考え、暫化式 $z_{n+1}=f(z_n)$ も同様に考える。

また式② $f(z)=cz(1-z)$ をも考える。

今回の描画では吸引領域にも色を付けているが、②の場合吸引領域はないように思われる。

図は Fig. 1 ~ Fig. 12 までであるが、Fig. 1 ~ Fig. 8 までの図が①の場合で、残りが②の場合である。

Table のデータを参照すればよく分かるが、今回なった方法のうち、①の場合、 c の実数部の数値はあまり変化させていない。しかし、虚数部の数値によって図形が大きく変化していることが判明した。また、虚数部の数値を + の方に増した場

合、ある時点で数値演算プロセッサの許容範囲を越えてエラーとなった。- の方への数値を増やした場合も同様にエラーとなった。このことは c の虚数部の符号を変化させた場合、図形の向きが変わるということでも納得できる (Fig. 7 と Fig. 8 の比較)。

c の虚数部を 0 とした時の図形は x 軸と y 軸に対して対象形となるが、0 以外の数値を代入すると円形の形が崩れてくる。虚数部の数値が大きくなるとこの崩れも大きくなっていく。②の場合、虚数部の数値を変化させると①の時と同じことがいえる。しかし、今度は実数部の数値の変化に対して①の場合とは異なった現象が起こる。数値を大きくしていくと出来る図形の y 軸に対する高さがなくなる。即ち上下に押しつぶしたような形となり中の部分の形が円形から圧縮されたような形となった (Fig. 9, Fig. 10, Fig. 11 の比較)。また、この数値を大きくしていくとある時点で黒で描いた部分がなくなる。以上が今回判明したことであるが、その他の数値についても調べなければ詳しいことは分からないと思われる。

③ コンピュータプログラム

前報の段階において、Turbo PASCAL Ver 3.0 であった。しかし、現在は Ver 5.0 が公開されている。筆者はこの Turbo PASCAL Ver 5.0 と NEC PC-9801 RA と数値演算プロセッサの組合せで描画をさせている。前報の自己平方ドラゴン

Table 図版のデータ

	X_{\min}	X_{\max}	Y_{\min}	Y_{\max}	cの実数部	cの虚数部
f(z) = z ² +c の場合						
Fig. 1	-1.0	1.0	-1.0	1.0	-0.1	0.0
Fig. 2	-1.0	1.0	-1.0	1.0	-0.1	-0.5
Fig. 3	-1.0	1.0	-1.0	1.0	-0.6	0.3
Fig. 4	-1.0	1.0	-1.0	1.0	-0.2	-0.5
Fig. 5	-0.5	0.5	-0.5	0.5	-0.1	-0.7
Fig. 6	-1.0	1.0	-1.0	1.0	-0.1	-0.845
Fig. 7	-1.0	1.0	-1.0	1.0	-0.1	-0.8
Fig. 8	-1.0	1.0	-1.0	1.0	-0.1	0.795
f(z) = c*z*(1-z) の場合						
Fig. 9	0.0	1.0	-0.4	0.4	3.2	0.0
Fig. 10	0.0	1.0	-0.4	0.4	3.32	0.0
Fig. 11	0.0	1.0	-0.4	0.4	3.4	0.0
Fig. 12	0.25	0.75	-0.2	0.2	3.3	0.2

のプログラムと今回のプログラムを比較すれば吸引領域を描くために追加した部分に分かるであろう。

また、Turbo PASCAL Ver 4.0 以降のバージョンでは機種依存型の言語になり現在 NEC PC-9801 以外の機種では走っていない。主な変更点はグラフィックユニットのようにユニットを使用してプログラミングが容易になっている。ユニットの使用はプログラムの始めの部分に使用するユニットを宣言すればよい。数値演算プロセッサもプログラムの始めの部分に、数値演算プロセッサが存在すればこれを選択して使用するようなルーチンを入れることが可能になった。筆者もこのルーチンを使用しているが、ここに掲載しているプログラムは複素数を取り扱っている。そのため宣言をしているが、これは数値演算プロセッサを入れていない場合にはエラーとなり、プログラムはストップする。また、表示色が16色になった。16色表示をディスクにセーブするためには次のようにすればよい。pc-9801 シリーズのグラフィックラム (VRAM) は &HA800 からが blue, &HB000 からが red, &HB800 からが green, &HE000 からが輝度情報となっている。従って、前報のプロ

グラムに拡張 VRAM の情報をディスクに書き込んだり、読み出したりするルーチンを追加すればよいことになる。

プログラムのリストを List-1~4 に掲載する。①が $f(z)=z^2+c$ のメインプログラム、②が $f(z)=cz(1-z)$ のメインプログラム、③がライブラリプログラム、④が表示プログラムである。前報でも述べたが、③のライブラリプログラムは grlib.pas 名でセーブしておく。④の表示プログラムは適当な名前前でセーブしておく。①、②のメインプログラムはコンパイルして .exe ファイルとすることも出来る。しかし、筆者の場合、表示色を変化させるためにプログラムの色を付けるルーチンをよく変更する。だからプログラムの実行を行なう都度、コンパイルするようにしている。

④ おわりに

集合系のフラクタルについてコンピュータに描かせるようになって約 2 年が経過した。Turbo PASCAL Ver 3.0 から Ver 4.0, Ver 5.0 と経過したときはプログラムする場合に容易になったこと、コンパイル速度が早くなったこと、計算速度も早くなったこと等の満足感があった。

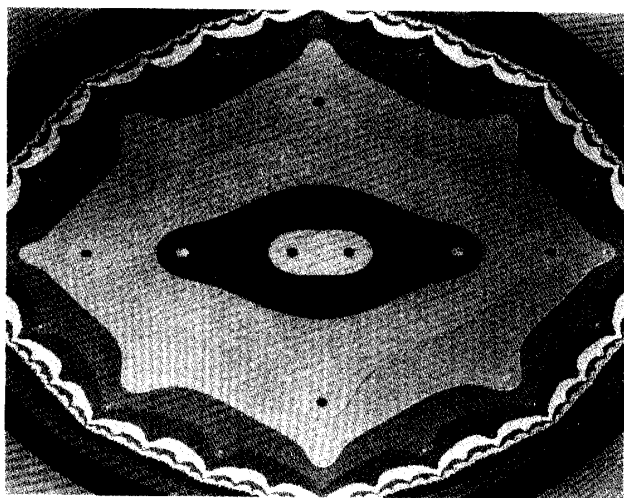


Fig. 1 $X: -0.1 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.1 + 0.0i$



Fig. 2 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.1 - 0.5i$

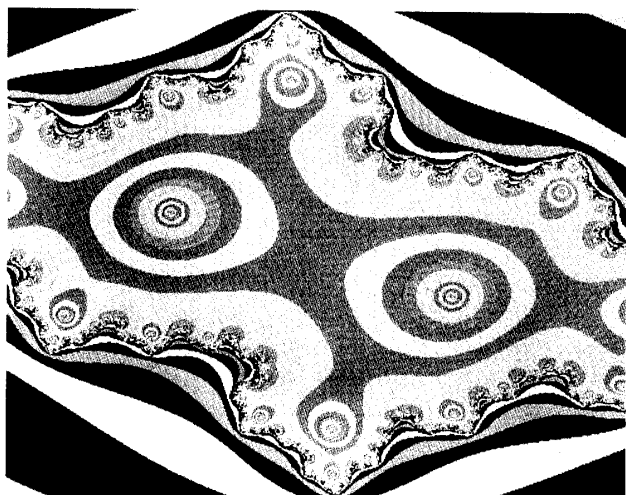


Fig. 3 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.6 + 0.3i$

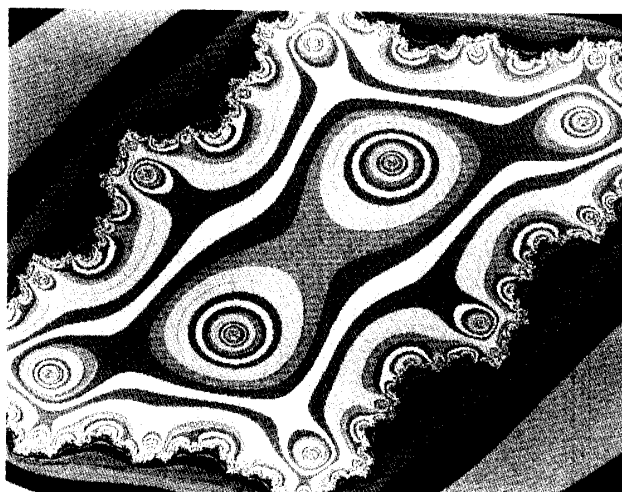


Fig. 4 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.2 - 0.5i$

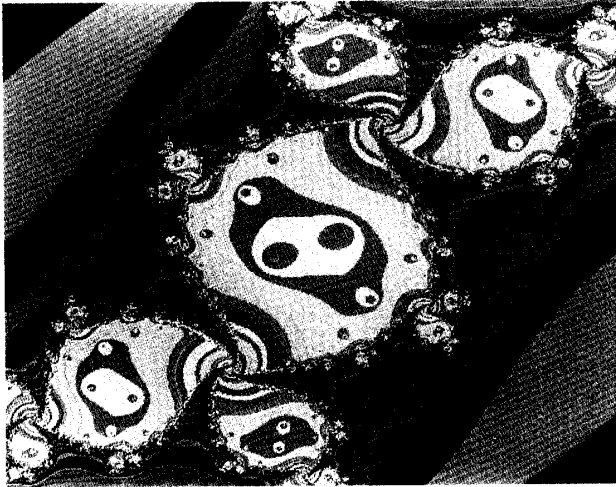


Fig. 5 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.1 - 0.7i$

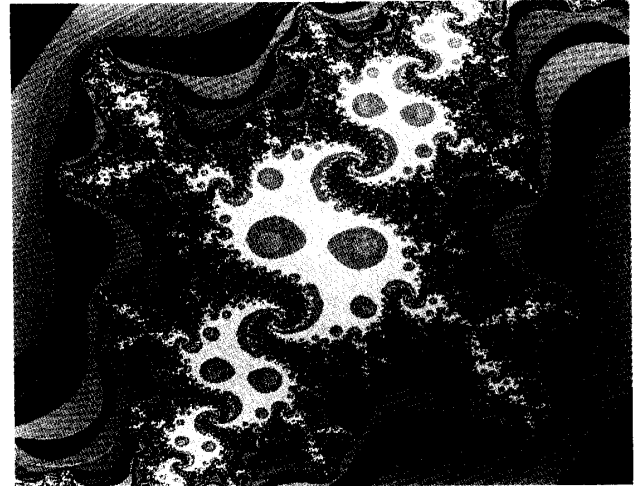


Fig. 6 $X: -0.5 \sim 0.5$
 $Y: -0.5 \sim 0.5$
 $C: -0.1 - 0.0845i$

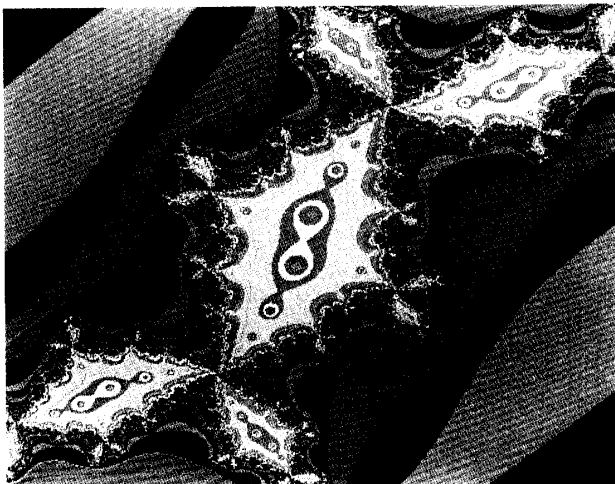


Fig. 7 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.1 - 0.8i$



Fig. 8 $X: -1.0 \sim 1.0$
 $Y: -1.0 \sim 1.0$
 $C: -0.1 + 0.795i$

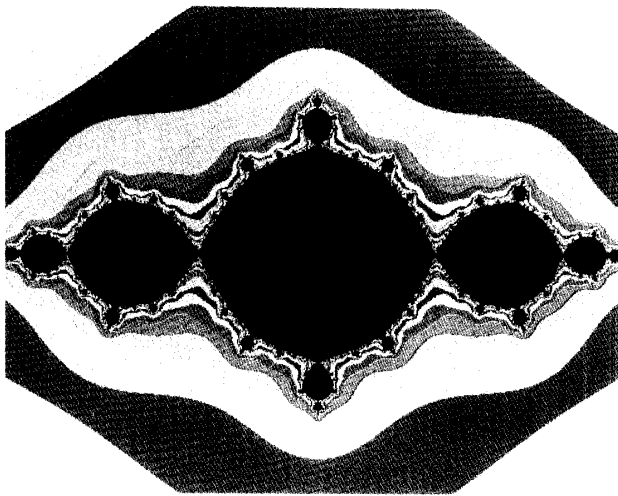


Fig. 9 $X: 0.0 \sim 1.0$
 $Y: -0.4 \sim 0.4$
 $C: 3.2 + 0.0i$

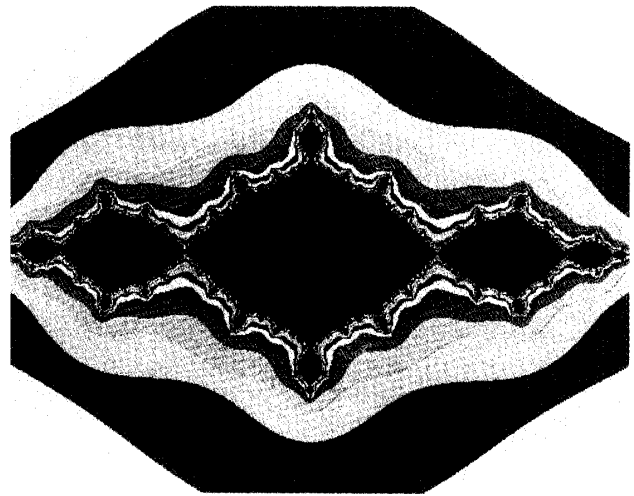


Fig. 10 $X: 0.0 \sim 1.0$
 $Y: -0.4 \sim 0.4$
 $C: 3.32 + 0.0i$

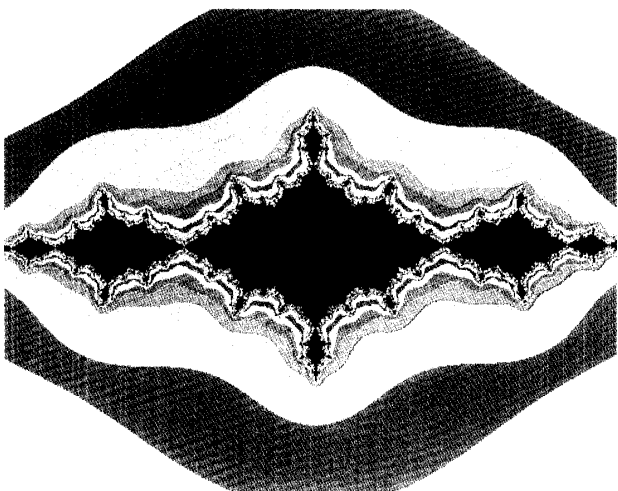


Fig. 11 $X: 0.0 \sim 1.0$
 $Y: -0.4 \sim 0.4$
 $C: 3.4 - 0.0i$

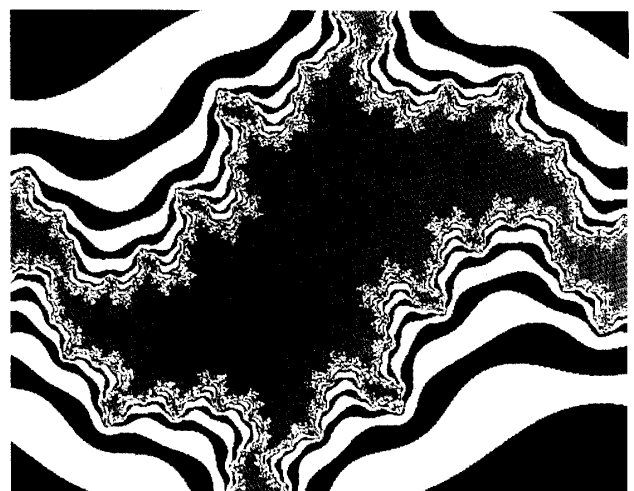


Fig. 12 $X: 0.25 \sim 0.75$
 $Y: -0.2 \sim 0.2$
 $C: 3.3 \sim 0.2i$

しかし、次のような不満も出てきた。

- ①画像のおもしろさをカラーフィルムへハードコピーしたいのだが経費がかかる。
- ②PC-9801 シリーズと Turbu-PASCAL の組合せでは画面の解像力が640*400ドットである。よってハードコピーを大きく引き伸ばした場合に画像としてはあまりよいとはいえない。

③表示できる色数が16色と少ない。

- ④ループ回数が大きくなれば PC-9801RA を使用しても描画に非常に時間がかかり、コンピュータの耐久試験をしているようなものである。

これらの不満を解消するために高性能な EWS などでのこの集合系のフラクタルを行うつもりである。

List-1

ジュリア集合 ($f(z)=z^2+c$) のメインプログラム

```

Program JuliaSet;

{$IFDEF CPU87}
{$N+}
{$ELSE}
{$N-}
{$ENDIF}

uses Crt,
     Dos,
     Graph;

{$I GRLIB.PAS}

Type Complex = Record re,im : double end;

var Wmin,Wmax,Wd,a,x0,x1,x2,x3,z1,z2,XNC,C : Complex;
    rr : extended;
    GraphDriver,Graphmode,Detect, Icc : integer;
    Ch1 : Char;
    FileNmRad,TodayDate : String14;

Procedure Ingr;
begin
  GraphDriver := InstallUserDriver('PC98',@Detect);
  GraphDriver := detect;
  InitGraph(Graphdriver,Graphmode,'');
end;

Procedure WindowSet;
begin
  clrscr;
  writeln('Region');
  writeln(' AX : ',Wmin.re:10:7,' : ',Wmax.re:10:7);
  writeln(' AY : ',Wmin.im:10:7,' : ',Wmax.im:10:7);
  writeln(' C : ',a.re:10:7,' : ',a.im:10:7);
  write(' AXmin = ');Readln(Wmin.re);
  write(' AXmax = ');Readln(Wmax.re);
  write(' AYmin = ');Readln(Wmin.im);

```

```

write(' AYmax = ');Readln(Wmax.im);
write(' C.re = ');Readln(a.re);
write(' C.im = ');Readln(a.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 = 1989/');readln(TodayDate);
ClrScr;
end;

Procedure JuliaPicture;
var Ix,Iy,Icol,IYT,IXT,K,IMAX,ia : integer;
    Axd,Ayd,EPSI,zz,EPS,SQN,X : extended;
Label PA;

Procedure CheckInterruption;
begin
if KeyPressed then
begin
Cleardevice;
closegraph;
Halt;
end;
end;

begin
IMAX := 700 ; EPSI :=4.0 ; EPS := 0.00001 ;

gotoxy(1,1);clrscr;
writeln('Julia');writeln('set');
writeln;writeln;writeln('Region');
writeln(' Ax = ');writeln(Wmin.re:10:7);writeln(Wmax.re:10:7);
writeln(' Ay = ');writeln(Wmin.im:10:7);writeln(Wmax.im:10:7);
writeln(' C = ');writeln(a.re:10:7);writeln(a.im:10:7,'i');
writeln;writeln;writeln(FileNmRad);
gotoXY(1,22);writeln('(C)');
writeln('Y,Yonekawa');write('1989/',TodayDate);
gotoXY(1,1);
Axd :=(Wmax.re - Wmin.re) / 500; Ayd := (Wmax.im - Wmin.im) / 400;
x0.re := 0.0 ; x0.im := 0.0 ;
for IYT := 1 to 1000 do
begin
x2.re := x0.re * x0.re - x0.im * x0.im + a.re ;
x0.im := 2.0 * x0.re * x0.im + a.im ; x0.re := x2.re ;
end; XNC := x0 ;
for IYT := 1 to 400 do
begin
x := Wmin.im + Ayd * IYT ; Iy := 400 - IYT;
for IXT := 0 to 500 do
begin
ix := 100 + IXT;
Icol := 7 ; PutPixel(Ix,Iy,Icol);

```

```

x0.re := Wmin.re + Axd * IXT ; x0.im := x ;
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);
  sqn := sqr(xnc.re - x0.re) + sqr(xnc.im -x0.im) ;
  if zz > EPSI then
  begin
    if K <= imax then
    begin
      icol := (k mod 6) + 1;
    end;
    goto PA;
  end;
  if SQN < EPS THEN
  begin
    ia := k div 4;
    icol := (ia mod 6) + 9 ;
    if icol =10 then icol :=12; goto PA;
  end;
  end;
  Icol := 0;
PA:    PutPixel(Ix,Iy,Icol);
      end;
      Checkinterruption;
    end;
end;

begin
  InGr;
  Windowset;
  Fileassign;
  Juliapicture;
  Grsave(FileNmRad);
  Cleardevice;
  CloseGraph;
end.

```

List-2

ジュリア集合 $(f(z)=cz(1-z))$ のメインプログラム

Program JuliaSet;

```

{$IFDEF CPU87}
{$N+}
{$ELSE}
{$N-}
{$ENDIF}

```

```

uses Crt,
     Dos,
     Graph;

```

```

{$I GRLIB.PAS}

```



```

Type Complex = Record re,im : double end;

var Wmin,Wmax,Wd,a,x0,x1,x2,x3,z1,z2,XNC,C : Complex;
    rr : extended;
    GraphDriver,Graphmode,Detect, Icc : integer;
    Ch1 : Char;
    FileNmRad,TodayDate : String14;

Procedure Ingr;
begin
    GraphDriver := InstallUserDriver('PC98',@Detect);
    GraphDriver := detect;
    InitGraph(Graphdriver,Graphmode,'');
end;

Procedure WindowSet;
begin
    clrscr;
    writeln('Region');
    writeln(' AX : ',Wmin.re:10:7,' : ',Wmax.re:10:7);
    writeln(' AY : ',Wmin.im:10:7,' : ',Wmax.im:10:7);
    writeln(' C : ',a.re:10:7,' : ',a.im:10:7);
    write(' AXmin = ');Readln(Wmin.re);
    write(' AXmax = ');Readln(Wmax.re);
    write(' AYmin = ');Readln(Wmin.im);
    write(' AYmax = ');Readln(Wmax.im);
    write(' C.re = ');Readln(a.re);
    write(' C.im = ');Readln(a.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 = 1989/');readln(TodayDate);
    ClrScr;
end;

Procedure JuliaPicture;
var lx,ly,Icol,IYT,IXT,K,IMAX,ia : integer;
    Axd,Ayd,EPSI,zz,EPS,SQN,X : extended;
Label PA;

Procedure CheckInterruption;
begin
    if KeyPressed then
        begin
            cleardevice;
            closegraph;
            Halt;
        end;
end;

begin
    IMAX := 700 ; EPSI :=4.0 ; EPS := 0.000001 ;

```

```

gotoxy(1,1);clrscr;
writeln('Julia');writeln('set');
writeln;writeln;writeln('Region');
writeln(' Ax = ');writeln(Wmin.re:10:7);writeln(Wmax.re:10:7);
writeln(' Ay = ');writeln(Wmin.im:10:7);writeln(Wmax.im:10:7);
writeln(' C = ');writeln(a.re:10:7);writeln(a.im:10:7,'i');
writeln;writeln;writeln(FileNmRad);
gotoXY(1,22);writeln('(C)');
writeln('Y,Yonekawa');write('1989/',TodayDate);
gotoXY(1,1);
Axd :=(Wmax.re - Wmin.re) / 500; Axd := (Wmax.im - Wmin.im) / 400;
x0.re := 0.0 ; x0.im := 0.0 ;
for IYT := 1 to 1000 do
begin
x2.re := (sqr(x0.im)-sqr(x0.re))*a.re+2*x0.re*x0.im*a.im+x0.re*a.re-x0.im*
a.im ;
x0.im := (sqr(x0.im)-sqr(x0.re))*a.im-2*x0.re*x0.im*a.re+x0.re*a.im+x0.im*
a.re ;
x0.re := x2.re ;
end; XNC := x0 ;
for IYT := 1 to 400 do
begin
x := Wmin.im + Axd * IYT ; ly := 400 - IYT;
for IXT := 0 to 500 do
begin
IX := 100 + IXT;
lcol := 7 ; PutPixel(Ix,ly,lcol);
x0.re := Wmin.re + Axd * IXT ; x0.im := x ;
for K := 1 to IMAX do
begin
x2.re := (sqr(x0.im)-sqr(x0.re))*a.re+2*x0.re*x0.im*a.im+x0.re*a.r
e-x0.im*a.im ;
x0.im := (sqr(x0.im)-sqr(x0.re))*a.im-2*x0.re*x0.im*a.re+x0.re*a.im
+x0.im*a.re ;
x0.re := x2.re;
zz := sqr(x0.re) + sqr(x0.im);
sqn := sqr(xnc.re - x0.re) + sqr(xnc.im -x0.im) ;
if zz > EPSI then
begin
if k <= imax then
begin
icol := (k mod 6) + 1;
end;
goto PA;
end;
if SQN < EPS THEN
begin
ia := k div 4;
icol := (ia mod 6) + 9 ;
if icol =10 then icol :=12; goto PA;
if icol =12 then icol :=10; goto PA;
end;
end;
icol := 0;
PA: PutPixel(Ix,ly,lcol);
end;
Checkinterruption;
end;
end;
begin

```

```

InGr;
Windowset;
Fileassign;
Juliapicture;
Grsave(FileNmRad);
cleardevice;
closeGraph;
end.

```

List-3

画像データのセーブプログラム

```
{ This file is for PC-9801. }
```

```
Const SegBase1 = $A800; SegBase2 = $B000; SegBase3 = $B800; SegBase4 = $E000;
```

```
Procedure wait;
begin Repeat until KeyPressed; end;
```

```
Type String14 = String[14];
```

```
Procedure GrLoad(var FNameRad : String14);
```

```
var FName : String14; GrFile : File;
```

```
begin
```

```
  clrscr;
```

```
  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);
```

```
  BlockRead(GrFile, Mem[SegBase4:0], 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, Dos.Registers(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];
        Mem[SegBase4:ICN3] :=FontBuffer[ICN2];
    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);
    BlockWrite(GrFile, Mem[SegBase4:0], 255);
    close(GrFile);
end;

```

List-4

画像データの表示プログラム

```

Program Grout;

uses dos, crt, graph;

{$I GRLIB.PAS}

var fileNmRad : string14;
    i, GraphDriver, Graphmode : integer;

begin
    Graphdriver := installuserdriver('PC98', @detect98);
    Graphdriver := detect;
    InitGraph(GraphDriver, Graphmode, '');
    write ('Graphic file name = ');
    readln(FileNmRad);
    if FileNmRad <> '' then Grload(FileNmRad);
    Wait; cleardevice; closeGraph;
end.

```