Julia fractals in PostScript

Fractal Geometry II

In memory of Hans Lauwerier

Abstract
Lauwerier's BASIC codes for visualization of the usual Julia fractals: JULIAMC, JULIABS, JULIAF, JULIAD, JULIAP, of the Mandelbrot fractal MANDELx, MANDIS, MANDET and his codes for the advanced circular symmetric Julia fractals JULIAS, JULIASYMm, JULIASYM, FRACSYMm, as well as the classical 1D bifurcation picture Collet, have been converted into PostScript defs. Examples of use are included. A glimpse into Chaos theory, in order to understand the principles and peculiarities underlying Julia sets, is given. Bifurcation diagrams of the Verhulst model of limited growth and of the Julia quadratic dynamical system — M-fractal — have been included. Barnsley's triples: fractal, IFS and equivalent dynamical system are introduced. How to use the beginnings of colours in PostScript is explained. How to obtain Julia fractals via Stuif's Julia fractal viewer, and via the special fractal packages Winfract, XaoS, and Fractalus is dealt with. From BASIC codes to PostScript library defs entails software engineering skills. The paper exhibits experimental fractal geometry, practical use of minimal TEX, as well as ample EPSF programming, and is the result of my next step in acquainting myself with Lauwerier's 10+ years work on fractals.

Keywords
Acrobat Pro, Adobe, art, attractor, backtracking, Barnsley, BASIC, bifurcation, Cauchy convergence, chaos, circle symmetry, dynamical systems, EPSF, escape-time algorithm, Feigenbaum constant, fractal dimension D, Fractalus package, FIFO, fractal geometry, IDE (Integrated development Environment), IFS (Iterated Function System), Julia, Lauwerier, Mandelbrot, mathematical software, minimal encapsulated PostScript, minimal plain TeX, Monte Carlo, μ-geometry, periodic doubling, Photoshop, PSlib, PSView, repeller, self-similarity, software engineering, Stuif's previewer, TEXworks, (adaptable) user space, Verhulst growth model, Winfract package, XaoS fractal package.

Contents

Introduction

Gaston Julia, a French mathematician, in his award-winning paper of 1918 discovered fractals, theoretically, avant la lettre. He studied among others recursions of the type

$$z_{i+1} = z_i^2 + c, \quad z_i, c \in \mathbb{C} \quad i = 0, 1, 2, \ldots$$

Fixed-points: $\{l_{1,2} \mid l = l^2 + c\} \rightarrow l_{1,2} = .5 \pm .5\sqrt{1-4c}$. ∞ is attractor for $|z_0|$ sufficiently large.

The boundary of the domain in $\mathbb{C}$ for which z_0 does not fly away to ∞ is called a Julia fractal.

Julia discerned three classes of fractals: plane-filling (rare), dust-like and connected (contours in contours in ...).

The answer to the question by Mandelbrot[1]

"For which values of c will the Julia fractal, J(c), be line-like and for which dust-like?"

marks the development of fractal geometry.

In the sequel we will

- □ discuss methods for drawing Julia repellers in PostScript
- □ show various Julia fractals
- □ show the use of `Stuif's Julia previewer`
- □ show the use of `WinFract`, `XaoS`, `Fractalus`[2]
- □ explain the various conversions in the appendices, with the `defs` included in my `PSlib.eps`
- □ tell how the beginnings of colouring can be done in PostScript, in the last appendix.

Although the paper is physically thick, I hope that the reader will recognise the logicallly thin parts. Parts can be read independently.
The section Catching-up is for the unwary. The section Barnsley's Triples is food for mathematicians. In the footsteps of Lauwerier the reader is invited to experiment with the PostScript programs and redo Lauwerier's exercises as posed in his booklets. Playing with `Stuif's Julia previewer`, `WinFrac`, `XaoS` or `Fractalus` is fun, and ... fascinating.

The LaTeX Graphics Companion contains some rudimentary classical Math fractals in MetaPost in section 4.4.3. No systematic programming approach à la Lindenmayer enriched with PostScript concepts, nor the use of PostScript's powerful facility to transform User Space, nor minimal PostScript, nor EPSF, are mentioned.[3]

The paper is my next step in acquainting myself with Lauwerier's 10+ years work on fractals.

The XaoS movie is an appetizer to the world of Julia fractals and its associated Mandelbrot map.

Catching up: quadratic dynamical systems

The purpose of this section is to make the unwary aware of the fact that innocent-looking iterative processes might converge, might bifurcate, repeatedly, or finally end-up in chaos.

---

1. Benoît B. Mandelbrot(Warsaw 20 November 1924 – Cambridge 14 October 2010) was a French American mathematician. Born in Poland, he moved to France with his family when he was a child. Mandelbrot spent much of his life living and working in the United States. Barnsley, M.F.& M.Frame(2012): Influence on Benoît Mandelbrot on Mathematics. `http://www.ams.org/notices/201209/index.html`.
2. `WinFract`, `XaoS` and `Fractalus` are fractal packages with rich colour and zoom facilities, for 'free to use' available on the WWW under the GNU public license. Stuif's Julia previewer is a Java Applet accessible via his WWW.
3. Understandable from the viewpoint to refrain from lower-level tools like PostScript.

In my education we practised a lot of iteration processes,[4] for example in the various zero finding processes. The accompanying picture shows the usual convergence for the determination of the intersection point of the parabola $y = x^2 + c$ and the line $y = x$ with $c \in [-.75, 25]$, for example c=-.5.

Julia Model with c=-.5 $x_{i+1} = f(x_i) = x_i^2 - .5, \quad i = 0, 1, 2, \ldots$
Fixed-points of $f(x)$:

$$l = l^2 - .5 \quad \rightarrow \quad \begin{aligned} l_1 &= .5(1+\sqrt{3}) \approx 1.367 \\ l_2 &= .5(1-\sqrt{3}) \approx -.367 \end{aligned}$$

Stability at fixed-points of $f(x)$:

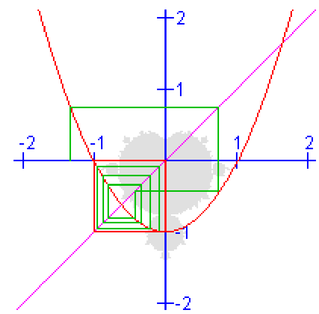
$$|f'(l_{1,2})| = |2l_{1,2}| \quad \rightarrow \quad \begin{aligned} &|f'(l_1)| = |1+\sqrt{3}| \approx 2.7321, \text{ repeller} \\ &|f'(l_2)| = |1-\sqrt{3}| \approx 0.7321, \text{ attractor.} \end{aligned}$$

Critical point of $f(x)$: $\{x \mid f'(x) = 0\} \quad \rightarrow \quad x = 0.$

Julia Model with c=.25, the cusp in the M-fractal: $x_{i+1} = f(x_i) = x_i^2 + .25, \quad i = 0, 1, 2, \ldots$

The intersection point becomes a tangent point. For $x_0 \in [-.5, .5]$ the dynamical system converges; for a starting value outside the interval the dynamical system diverges.

But ... what if we take $c = -1$?



On the World Wide Web, WWW for short, I found a fractal primer,[5] aimed at a broad audience with Math at the high-school level. Julia sets, Fatou domains, (strange) attractors, the Feigenbaum number and bifurcation are introduced. Stuif discusses the intersection point of the parabola, $y = x^2 - 1$, and the straight line $y = x$, supported by (Java) applets for experimentation.
Important concepts in dynamical systems are: fixed-points and stability at the fixed-points, the critical points, and the so-called strange attractors c.q. repellers.

Julia Model with c=-1 $x_{i+1} = f(x_i) = x_i^2 - 1, \quad i = 0, 1, 2, \ldots$[6]
Fixed-points of $f(x)$:

$$l = l^2 - 1 \quad \rightarrow \quad \begin{aligned} l_1 &= .5(1+\sqrt{5}) \approx 1.618, \qquad \phi \\ l_2 &= .5(1-\sqrt{5}) \approx -.618, \quad -1/\phi \end{aligned}$$

Stability at the fixed-points of $f(x)$:

$$|f'(l_{1,2})| = |2l_{1,2}| \quad \rightarrow \quad \begin{aligned} &|f'(l_1)| = |1+\sqrt{5}| > 1, \text{ repeller} \\ &|f'(l_2)| = |1-\sqrt{5}| > 1, \text{ repeller.} \end{aligned}$$

Critical point of $f(x)$: $\{x \mid f'(x) = 0\} \quad \rightarrow \quad x = 0.$
Because of the repeller nature at the fixed-points we don't expect convergence.

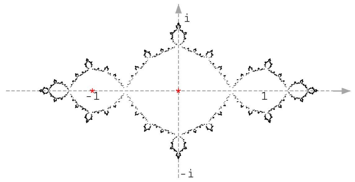
4. Fractals raised a new awareness of, a new insight in, iterative processes as can be witnessed from the title of Mandelbrot's invited paper 'Fractals and the Rebirth of Iteration Theory,' in Peitgen c.s.(1986)
5. `http://www.stuif.com/fractals/index.html`
6. The general case with c complex, will be treated later.

Non-convergence, with a flip-flop behaviour with values 0 and -1, the so-called strange attractors. The splitting-up is called bifurcation.

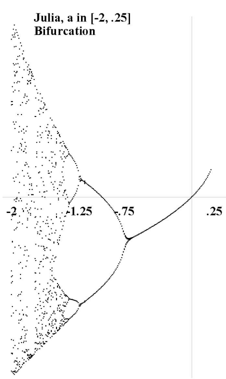
Bifurcation condition for 2-period:

$$\begin{cases} x &= w^2 - 1 \\ w &= x^2 - 1 \end{cases} \quad \rightarrow \quad \begin{matrix} x &= 0 \\ w &= -1 \end{matrix} \quad \text{the flip-flop behaviour.}$$

The associated Julia fractal, J(-1), picture looks like



Starting points within the Julia fractal, J(-1), don't fly away to infinity. They either stay on the fractal in a chaotic way or converge to the strange attractors 0 and -1.



In the accompanying bifurcation diagram, the behaviour of the Julia dynamical system for real parameter, a, is summarized.
For $a \in [-.75, .25]$ there is the usual convergence.
For $a \in [-1.25, -.75]$ 2-periodic bifurcation pops up.
A little further higher bifurcations come into light as well as the chaotic behaviour
Beyond -2 the Julia fracal flies away to ∞ except for the dust Julia fractals.
In literature I did not stumble upon the bifurcation diagram for the Julia system with real parameter.
The general complex parameter bifurcation diagram is the M-fractal.

Verhulst limited growth model is related to the Julia dynamical system. Moreover, Lauwerier has extensively studied this model; his results have been summarized below.

Verhulst model $x_i = f(x_{i-1}) = ax_{i-1}(1 - x_{i-1}) \quad i = 0, 1, 2, \dots, \quad 0 < a \leqslant 4.$
Fixed-points of f(x):

$$l = al(1 - l) \quad \rightarrow \quad \begin{matrix} l_1 = 0 \\ l_2 = 1 - 1/a \end{matrix}$$

Stability at the fixed-points of $f(x)$:

$$|f'(l_{1,2})| = |a(2l_{1,2} - 1)| \rightarrow \begin{matrix} f'(l_1) = |a|, & \text{attractor for } 0 < a < 1 \\ f'(l_2) = |a - 2|, & \text{attractor for } 1 < a < 3 \end{matrix}$$

The last 20 values of $x_n = ax_{n-1}(1 - x_{n-1}), \quad n = 0, 1, 2, \dots, 100$, are shown below for a few values of a; the (non)convergence behaviour is striking.

Bifurcation condition for 2-period:

$$\begin{cases} x = w(aw - (a - 1)) \\ w = x(ax - (a - 1)) \end{cases} \quad \rightarrow \quad \begin{matrix} x = (1 + a + \sqrt{a^2 - 2a - 3})/(2a) \\ w = (1 + a - \sqrt{a^2 - 2a - 3})/(2a) \end{matrix}$$

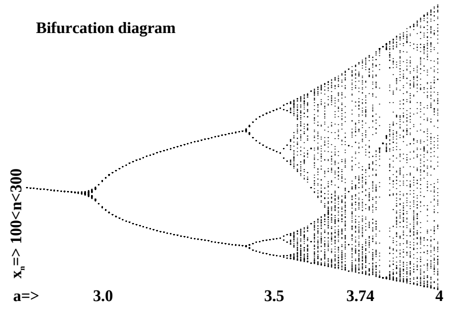
The class of iterations for non-linear functions have earned a place of their own in nowadays Math. The field is called dynamical systems.

Behaviour Verhulst model

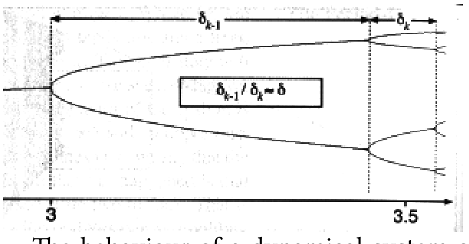
Lauwerier(1987, ch 6, paragraph Getal van Feigenbaum) summarizes the behaviour of the Verhulst model as function of a:[7]

$0 < a <= 1$	$\Rightarrow$	$x_n \rightarrow 0$	super stable equilibrium
$1 < a <= 2$	$\Rightarrow$	$x_n \rightarrow 1 - 1/a \approx .6875$	monotone, stable equilibrium
$2 < a <= 3$	$\Rightarrow$	$x_n \rightarrow 1 - 1/a \approx .6875$	oscillating, stable equilibrium
$a \approx 3.2$	$\Rightarrow$	$x_n \rightarrow \approx .51$ and $\approx .78$	2-cycle bifurcation
$a \approx 3.45$	$\Rightarrow$	$x_n \rightarrow \approx .50, \approx .88, \approx .38, \approx .83$	4-cycle bifurcation
$a \approx 3.54$	$\Rightarrow$	$x_n \rightarrow ...$	8-cycle bifurcation
$a \approx 3.83$	$\Rightarrow$	$x_n \rightarrow ...$	3-cycle bifurcation
$a \approx 3.84$	$\Rightarrow$	$x_n \rightarrow ...$	6-cycle bifurcation
$3.56 < a <= 4$	$\Rightarrow$	$x_n \rightarrow ...$	chaotic or periodic

The bifurcation diagram repeats itself if we zoom in. It behaves like a fractal. Isn't a miracle that we could find zeroes in the past at all?



Lauwerier's tiny bifurcation diagram of 1987, his program `COLLET`, has been improved with better legend and anachronisms with respect to screen-size has been removed in PostScript. It is called `bifurcation` of which the result has been included. The numbers 3.0 etc. indicate for which values of a a periodic doubling, or otherwise, shows up. Lauwerier(1996) discusses in detail the bifurcation diagram. Moreover, programs are provided for showing magnified parts of the diagram. He also treats the case a=4, analytically. The various periodic behaviours and the chaos are not specific for this function. In 2D, and for complex values of z_0, the Julia set gave rise to interesting pictures: fractals!
Informally, the Fatou set of the function consists of values with the property that all nearby values behave similarly under repeated iteration of the function, and the Julia set consists of values such that an arbitrarily small perturbation can cause drastic changes in the sequence of iterated function values.



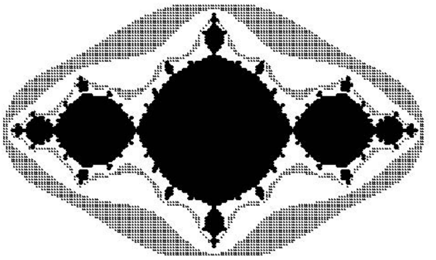
Chaos research yielded
Feigenbaum's universal constant $\delta \approx 4.6692$:
the quotient of successive bifurcation lengths.
Feigenbaum's constant appears in physics, chemistry, ...

The behaviour of a dynamical system on the Fatou set is 'regular', while on the Julia set its behaviour is 'chaotic'. In the case $z^3 = 1$, see the accompanying illustration, courtesy Georg-Johann Lay Wikipedia, when starting with a point on the 'necklace' we can't tell beforehand to which zero the process will converge.
Explanation. The Julia set (in white) is visualized which associates Newton's method for locating the zeroes of the 3^{rd} degree polynomial equation by the iteration $z_{i+1} = \frac{2}{3}z_i + \frac{1}{3z_i^2}$ to the Julia fractal. The colours indicate which starting point converges to which zero. It shows that close-by starting points converge to different zeros in an irregular, intriguing pattern.

7. For a=3.2 substitution in the above formula yielded .8 instead of .78. Maybe 3.2 is not accurate enough?

Barnsley's triples

Barnsley(1988) considers triples: dynamical system, equivalent IFS and attracting fractal. His escape time algorithm is similar to the JULIAP casu quo JULIAF methods. The theoretical, rigorous math approach is food for mathematicians. The accompanying J(-1), by Barnsley's program, has a contour for which the escape number is 5. The broader band are the points with escape number 2. The points with escape number infinity are considered to lie on and within the fractal, the raison d'être of the escape-time algorithm.



Julia fractals

Properties

- ☐ is a point on the Julia set, J, then all its successors and predecessors are on J
- ☐ the Julia set is point symmetric, $z^2 = (ze^{i\pi})^2$, which property is used to save computation time
- ☐ J(c) is x-axis mirror symmetric with J($\bar{c}$), like the Mandelbrot fractal
- ☐ there are 3 kinds of Julia fractals: plane-like, line-like and dust-like
- ☐ if $\lim_{k\to\infty} z_k(a,b) = \infty, z_0 = 0$ then J(a,b) is dust-like
- ☐ the points around the fractal are repelled from the fractal, J(a,b) is a repeller.

Relation Verhulst model and the Julia dynamical system

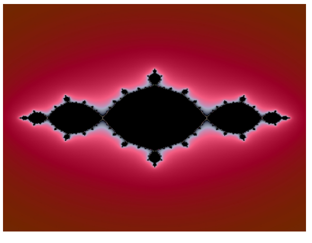
The Julia dynamical system is a sort of 2D extension of the Verhulst growth model, because of working in $\mathbb{C}$, with all the peculiarities of the 1D Verhulst model inherited, Lauwerier(1996, ch8).

$$\text{Verhulst:}\quad \xi_n = a\xi_{n-1}(1-\xi_{n-1}) \quad \xrightarrow{\xi=.5-x/a} \quad \text{Julia:}\quad x_n = x_{n-1}^2 + a/2 - a^2/4$$

$$\text{Julia:}\quad x_n = x_{n-1}^2 + c \quad \xrightarrow{x=a(.5-\xi)\wedge c=a/2-a^2/4} \quad \text{Verhulst:}\quad \xi_n = a\xi_{n-1}(1-\xi_{n-1})$$

Mandelbrot's map of Julia fractals

Mandelbrot was interested in: for what values of c is a Julia fractal line-like and for what values dust-like. The 'map of Julia fractals', the bifurcation diagram, is the famous fractal named after Mandelbrot, M-fractal for short. When e.g. the white + in the M-picture, coordinates (-1, 0), is pointed at by the cursor, with the Julia mode active of `Fractalus`, then the J(-1, 0) fractal is opened in another window.

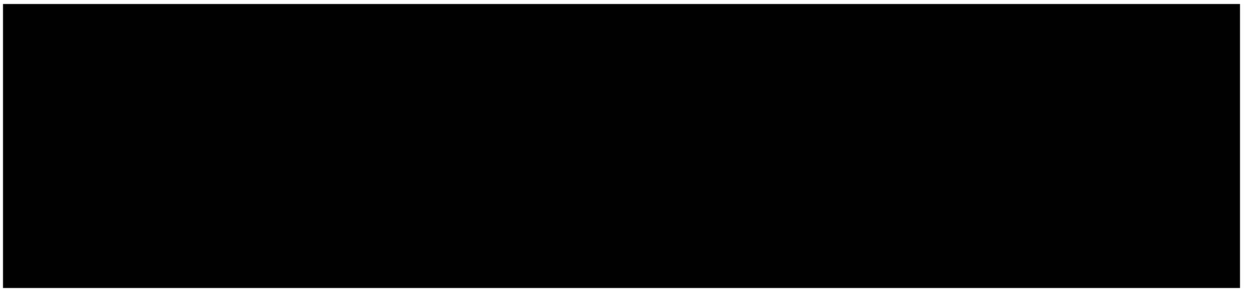
→ 

Peitgen c.s.(1986) considered the 'M-fractal as map' important as can be witnessed from the full-page illustration by Milnor.
Barnsley(1988) contains a similar picture.
It gives a survey of the various interesting Julia fractals and their position in the M-fractal.

Zooming-in, or … going for details

Drawing fractals as such is one thing. Visualization of the endless repetition, the microscopic detail, is another. The (near) repetition of the main picture in the details, this near to self-similarity property, as well as the new forms that arise make fractals intriguing.

Zooming-in the M-fractal, zooming-in again, again and again is shown in the accompanying picture, which is borrowed from the Mandelbrot Wikipedia.



Lauwerier(1994, 1996) contains the programs `MANDET`(ail) and `MANDIS`(tance) for visualization of a detail of the M-fractal. The J-fractals and the M-fractal have their playground on the square inch: μ-geometry!

Lauwerier(1994) also provided for a zoom program, `MZOOML`, in BASIC (3p), which I consider no longer relevant. It is overruled by the easy to use zoom facilities of `Winfract`, `XaoS`, `Fractalus`, … .

Use of the various PostScript defs

The PostScript `defs` expect on the stack: the values for the problem parameter `a`, `b`, the coordinates of the fractal domain specified by the values for the upper-right corner $\mathbf{x}_{\mathrm{ur}}$, $\mathbf{y}_{\mathrm{ur}}$, and the maximum number of iterations. The sharper the specifications of the fractal domain the faster the program. In the header comment of the `EPSF` program the values for the BoundingBox, BB for short, should be supplied in order to get a cropped picture. The BB-values are default 100 (scaling factor) times the fractal domain values.

Fractal	**a**	**b**	$\mathbf{x}_{\mathrm{ur}}$	$\mathbf{y}_{\mathrm{ur}}$
San Marco	-1	0	1.65	0.85
Dragon	-0.7454	0.1103	1.5	0.9
Douady	-0.12	0.74	1.2	1.3
Douady conj	-0.12	-0.74	1.2	1.3
Dendrit	0	1	2.1	1.85
Lightning	-1.029	0.386	1.65	0.85
Leaves	0.11	0.66	1.3	1.3
Cusp	0.25	0	0.85	1.1
4 spirals	0.55	0	0.85	0.85
Cloud	-0.59	-0.34	1.45	1

-1.5 0 5000 JULIAMC
San Marco (flat)

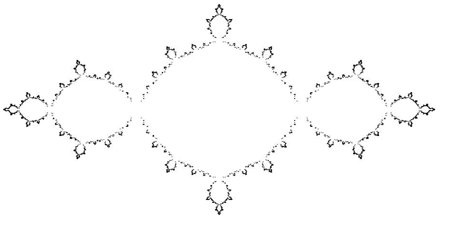
Using JULIAMC

The `def JULIAMC` implements inverse iteration. The parameters of the `def JULIAMC` are: the real and imaginary part of the problem parameter `c`, i.e. `a` and `b`, and the number of points of the fractal to be generated, `n`. It is the simplest Julia fractal generator.

$$z_i = z_{i-1}^2 + c, \quad i = 1, 2, 3, \ldots \qquad \to$$

$$\text{Inverse:} \quad z_{i-1} = \pm\sqrt{z_i - c}, \quad i = n+10, \ldots,\ 1, \quad |z_{n+10}| <= 1.$$

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -165 -85 165 85
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run%PSlib(rary) inclusion
%%EndProlog
%
% Program ---the script---
%
-1 0 5000 JULIAMC         %SanMarco
showpage
%%EOF
```



In order to have the picture cropped for `EPSF` use appropriate `BB`-values have to be inserted.

The Julia set for $z_i = z_{i-1}^2 + c, \quad i = 1, 2, 3, \ldots$ is a repeller; the Julia set for the inverse iteration is an attractor. The inverse operation is a 2-valued function because of the square root. It is enough to use one of the values randomly. This random use explains the suffix `MC`, from the Monte Carlo casino, in the name.

The BASIC code `JULIAMC` and its conversion into a PostScript `def` is discussed in the first appendix. If you, kind reader, don't want to bother with the use of the PostScript library, then just include the PostScript `def`, in place of the library inclusion as follows.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -165 -85 165 85
%%BeginSetup
%%EndSetup
%%BeginProlog
```

```
/JULIAMC{%stack: a, b, maxk.
        %==> Julia set of z^2 + a + ib via Inverse Iteration
...
end}bind def
%also include the corresponding dictionary
/JULIAMCdict 11 dict def
%also centershow for centering the dots
/centershow{%string --> show string centered
gsave dup stringwidth pop 2 div neg 0 rmoveto show grestore} def
%%EndProlog
%
% Program ---the script---
%
-.75 0 5000 JULIAMC%SanMarco
showpage
%%EOF
```

-.75 0 5000 JULIAMC, San Marco, D≈1.27

This inclusion makes clear how handy a `PSlib.eps` is. It keeps a program small and clear, with all `defs` stored together, separately and well-organized. For exact `BB`-values use the values prompted by `pathbbox` and print the values via the library `def showobject`. Finally, insert these values, `llx lly urx ury` in the header comment `%%BoundingBox: llx lly urx ury`, for an exact cropped picture in the next run, where the picture can also be reused as `EPSF`. A 2-pass process.[8]

0 1 5000 JULIAMC
Dendrit, D≈1.2

-.8 .15 5000 JULIAMC
Dragon

.25 0 5000 JULIAMC
Club

.11 .66 5000 JULIAMC
Leaves

-1.03 .386 5000 JULIAMC
Lightning

.55 0 5000 JULIAMC
4 Spirals

In the figures the lines are not of equal thickness, some parts are even blank. To compensate for this there are the boundaryscan method and the distance formula method.

Using JULIABS

`JULIABS` implements the Boundary Scan method. The parameters of the `def JULIABS` are: the real and imaginary part of the problem parameter `c`, i.e. `a` and `b`, the upper-right corner of the rectangle where the fractal is located, i.e. x_{ur}, y_{ur}, and the maximum number of iterations per grid point, i.e. `kmax`.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -180 -90 180 90
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run  %inclusion of PSlib(brary)
%%Endprolog
```

8. How to obtain the same effect in a 1-pass job for EPSF, see Recreational use of TEX&Co. This proceedings

```
%
%   Program ---the script---
%
1 .7 scale 0 0 195 0 360 arc%scale (circle to oval)
lightblue fill             %fill oval by background colour
blue -1 0 1.8 .9 80 JULIABS %blue San Marco
showpage
%%EOF
```

-1 0 1.8 .9 80 JULIABS
San Marco, D≈1.29

-.12 .74 1.2 1.3 80 JULIABS
Rabbit Douady, D≈1.39

-.74 .11 1.5 .9 80 JULIABS
Dragon

0 .65 1.4 1.4 80 JULIABS
Disconnected

.11 .66 2.1 1.85 80 JULIABS
Leaves

.25 0 .9 1.15 80 JULIABS
Cusp

There is no contour as path, just dots. From this collection of points I was not able to construct a contour in PostScript; in Photoshop, yes. Dendrit and Lightning were not obtained by this method.

Using JULIAF

`JULIAF` implements Filling the area of the fractal. The parameters of the `def JULIAF` are: the real and imaginary part of the problem parameter `c`, i.e. `a` and `b`, the upper-right corner of the rectangle where the fractal is located, i.e. x_{ur}, y_{ur},[9] the maximum number of iterations per grid point, i.e. `kmax`.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -150 -90 150 90
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run
%%EndProlog
%
%   Program the script
%
gsave
1 .7 scale 0 0 150 0 360 arc %oval
lightblue fill              %fill oval by background colour
grestore                    %end effect of scaling
blue -.8 .15 1.5 .9 50 JULIAF%blue Dragon
showpage
%%EOF
```

9. Default the BB values are a factor 100 times the values of the rectangle where the fractal is located.

-1 0 1.65 .85 80 JULIAF
San Marco

-.12 .74 1.3 1.2 80 JULIAF
Rabbit of Douady

-.55 0 1.45 1 50 JULIAF
Rounded San Marco

Using JULIAP

JULIAP implements the Pixel method. The parameters of the def JULIAP are: the real and imaginary part of the problem parameter c, i.e. a and b, the upper-right corner of the rectangle where the point-symmetrical fractal is located, i.e. x_{ur}, y_{ur}, and the maximum number of iterations per grid point, i.e. kmax.

Colour-banded Julia fractals are obtained. The number of iterations, the escape number, needed to 'escape to ∞,' is used for the selection of the colour. The closer the initial point z_0 lies to the fractal the longer it takes to pass the threshold of the ∞-attractor, i.e. the larger the escape number. For the points within the fractal the escape number is defined by the number of iterates for which 2 successive iterates are close enough to each other, the Cauchy criterion. For the case of periodic cycles the escape number is the maximum number of iterates, i.e. kmax.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -160 -140 160 140
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run %Inclusion of PSlib(rary)
%%EndProlog
%
%   Program, the script
%
gsave
1 .7 scale 5 0 85 0 360 arc%ellips
lightblue stroke          %background colour ellips
grestore                  %end effect of scale and colour
-.55 0 1.6 1.4 50 JULIAP  %colour-banded Julia fractal
showpage
%%EOF
```

-1.3 0 1.65 .85 80 JULIAP
San Marco flat var.

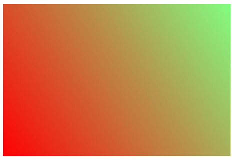
-.12 .74 1.3 1.2 80 JULIAP
Rabbit of Douady

.11 .66 1.3 1.3 80 JULIAP
Leaves

We might ask ourselves the question:
How to colour by gradients?
The primitive way is to scale and colour the scaled-lines by slightly changing colours. Another way is to make a contour of the fractal, in Photoshop for the time being, and apply colouring by gradients. The accompanying cloud example has been obtained in 2009 in this way; the doughnut earlier.

PostScript 3 provides facilities for graceful and high-quality colouring by gradients

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 100 400 400 600
100 400 300 200 rectclip
<</ShadingType 2 /ColorSpace [/DeviceRGB] /Coords [100 400 400 600]
  /Function<</FunctionType 2
             /Domain [0 1]
             /C0 [1 0 0] /C1 [.5 1 .5]%colours at end points
             /N 1 >>
>>
shfill%shadefill
showpage
```



Using the powerful JULIAD

`JULIAD` implements the use of the Distance formula. The parameters of the `def JULIAD`, are: the real and imaginary part of the problem parameter `c`, i.e. `a` and `b`, the upper-right corner of the rectangle where the fractal is located, i.e. x_{ur}, y_{ur}, the maximum number of iterations per grid point, i.e. `kmax`, and the distance formula parameter, e.g. 0.01. The pictures below show the richness of detail obtained.

-.775 .1103 1.5 .9 75 .01 JULIAD

-.7454 .1103 1.5 .9 75 .01 JULIAD

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -150 -90 150 90
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run          %Inclusion of the PSlib(brary)
%%EndProlog
%
%   Program, the script
%
gsave
1 .7 scale 0 0 150 0 360 arc          %oval
lightblue fill                        %fill the oval by background colour
```

```
grestore
blue -.7454 .1103 1.5 .9 75 .01 JULIAD%blue, scaled inner structured Dragon
showpage
%%EOF
```

-.74 .11 1.5 .9 75 .01 JULIAD
Dragon

-1.03 .386 1.65 .85 50 .01 JULIAD
Lightning

-.55 0 2.1 1.85 50 .01 JULIAD
Nesting of clouds

-1.3 0 1.65 .5 50 .01 JULIAD
San Marco flat var.

-.12 .74 1.2 1.3 50 .01 JULIAD
Rabbit of Douady

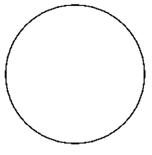
.11 .66 1.3 1.3 50 .01 JULIAD
Leaves

-1 0 1.65 .85 50 0.1 JULIAD
San Marco, f=.1

-1 0 1.65 .85 50 .01 JULIAD
San Marco, f=.01

-1 0 1.65 .85 50 .001 JULIAD
San Marco, f=.001

Compared with earlier methods J(.11, .66) looks different, apparently because of the more accurate method. This raises the question: What does the real fractal look like?



0 0 1.1 1.1 50 0.01 JULIAD

Degenerated cases
$\leftarrow$ J(0), a circle
J(-2), a line $\rightarrow$
$J(0) \xrightarrow{z+\frac{1}{z}} J(-2)$

-2 0 2.1 0.1 50 .01 JULIAD

J(-3.45) is disconnected dust

The invoke `-3.45 0 3 .25 50 .0000001 JULIAD` did not yield more details; the dust reminds me of Cantor. The dust does not contain more details, apparently.

I was curious whether once one point has been found of the dust, the others would be obtained by gambling inverse iteration: 4 values were found $\pm 2.62, \pm 1.98$.

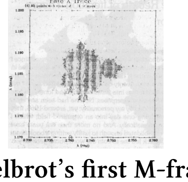
Using MANDEL and variants

Mandelbrot in 1980 answered the question posed in the introduction:

For which values of c will the Julia fractal, J(c),
be line-like and for which values dust-like?

He was surprised, but ... realized the relevance.

The picture consists of a cardioid, some circular bulbs, hairy details and stretching out with an 'antenna'. So nice to find a real-life application where a classical Math contour, cardioid, pops up.



Mandelbrot's first M-fractal

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox:  -210 -135 85 135
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run
%%Endprolog
%
%   Program ---the script---
%
MANDEL
%respectively
%MANDELzw
%MANDELzwcontour
showpage
%%EOF
```

Mandelbrot also elaborated on the fractal dimension notion. The M-fractal curve, and surface, have fractal dimension D=2.

Fixed-points of the Julia quadratic dynamical system: $\{l \mid l = l^2 + c\}$.

Stability of the dynamical system: $|f'(z)| < 1$.
$z = e^{i\varphi}/2$ lies on the boundary of $|f'(z)| = 1$, i.e. $|2z| = 1$.
Substitution of $z = e^{i\varphi}/2$ in the equation for the fixed-point, yields

$$c = a + ib = e^{i\varphi}/2 - e^{2i\varphi}/4 \rightarrow \begin{cases} a = \cos(\varphi)/2 - \cos(2\varphi)/4 \\ b = \sin(\varphi)/2 - \sin(2\varphi)/4 \end{cases},$$

the equation of a cardioid.
The c-values for which J(c) has 1 attracting fixed-point lie within the M-cardioid.

2-period bifurcation of $f(z) = z^2 + c \rightarrow \begin{cases} z = w^2 + c \\ w = z^2 + c \end{cases} \quad \wedge \quad |f'(z)f'(w)| < 1.$

From the 2 equations we yield $zw = c + 1$. Together with $|f'(z)f'(w)| = 4zw$ we arrive at $|c + 1| < 1/4$, the circle left of the cardioid. The 2-period bifurcation occurs for c in the circle $C_{(-1,\frac{1}{4})}$.

More, and higher, bifurcations are in the necklace of surrounding circle-shaped bulbs, which themselves have surrounding necklaces in a fractal way.[10]

The most amazing of the Mandelbrot fractal is that by magnification slightly different repetitions of the main picture show up.

Wim W. Wilhelm's Mandelbrot fractal via Asymptote Wim works with Asymptote and LaTeX within the TeXnicCenter IDE, and also uses Mathematica. I have included his program and picture (circular-cropped by Photoshop) to please Asymptote-LaTeX users. Handy are the complex data type `pair`, the Metafont-like `path` datatype and the reflection operator about a line, next to the C++-like language features.

```
size(10cm,0);
real mandelbrot(pair c, real r, int count=100)
{int i=0; pair z=c;
 do { ++i; z=z^2+c;} while (length(z) <= r && i<count);
 return (i<count) ? i/count : 0;}
real r=4, step=.01, xmin=-2.25, xmax=.75, ymin=-1.3, ymax=0;
```

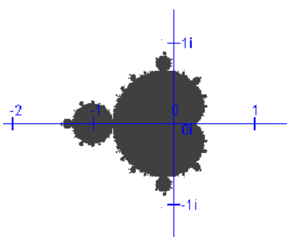
10. Lauwerier(1996, p195).

```
real x=xmin, y=ymin;
int xloop=round((xmax-xmin)/step);
int yloop=round((ymax-ymin)/step);
pen p; path sq=scale(step)*unitsquare;
for(int i=0; i < xloop; ++i)
 {for(int j=0; j < yloop; ++j)
  {p=mandelbrot((x,y),r,20)*red;
   filldraw(shift(x,y)*sq,p,p);
   y += step;
  }x += step; y=ymin;
 }
add(reflect((0,0),(1,0))*currentpicture);
```

Using MANDET

The form and size of the M-fractal is known and depicted in the left picture below, courtesy Stuif. Lauwerier provides programs for viewing details of the M-fractal: `MANDET`(ail) and `MANDIS`(tance). The first 2 parameters of `MANDET` denote the centre of the rectangle of the fractal domain, i.e. `ac` and `bc` (e.g. in 6 decimals). The third parameter, `d`, is half the width, and height,[11] of the fractal rectangle to be viewed. The `4th` parameter is the maximum number of iterations, `kmax`, of the inner-loop. After termination of the inner-loop, the value of the inner-loop variable is used as escape number for determination of the colour.

I chose for `MANDET` and `MANDIS` the rectangular `BoundingBox: -400 -300 400 300`, wired-in, conform the shape of the M-fractal; no deformation by scaling. The grid-points $\{i \mid i = -400, -399, \dots, 399, 400\}$ and $\{j \mid j = -300, -299, \dots, 299, 300\}$ of the `BoundingBox` are scaled down to the grid-points of the fractal square by $\{(a_i, b_j) \mid (ac + d * i/400,\ bc + d * j/400)\}$. The approach is similar to Lauwerier, who maps the full-screen on the fractal area.



Form and size of M-fractal

Mapping BoundingBox onto fractal rectangle

By halving the `d` parameter the magnification is doubled. More details are obtained by increasing `kmax`.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox:  -400 -300 400 300
%%BeginSetup
%%EndSetup
%%BeginProlog
(c:\\PSlib\\PSlib.eps) run
%%Endprolog
%
%   Program ---the script---
%
-1.953712 0 .000049 100 MANDETzw%Black and White variant
% ac      bc d      kmax        %parameter mnemonics
showpage
%%EOF
```

11. The latter is wired-in, scaled in the `def` to 75%, in accordance with the shape of the M-fractal

a	b	d	kmax	time
-1.953712	0	0.000049	200	
-1.927199	0	0.0005	200	31
-1.749057	0.000306	0.000004	300	
-1.28408	0.42726	0.000625	100	75
-1.256362	0.38032	0.008	100	74
-0.91667	0.26667	0.06	200	90
-0.7789	0.1344	0.0032	150	
-0.7777	0.1355	0.0037	150	125
-0.74691	0.10725	0.0008	800	744
-0.746371	0.098641	0.00001	300	
-0.7392	0.1745	0.0028	200	123
-0.698	0.3785	0.004	100	113
-0.160651	1.036793	0.00001	600	
-0.15067	1.04504	0.000006	200	
-0.102324	0.95716	0.000007	200	
-0.1011	0.9563	0.0016	200	75
-0.023262	0.999253	0.003	100	59
-0.01556	1.02071	0.0015	100	63
0.28	0.28	0.00028	400	
0.2812	0.00948	0.00005	300	
0.2813	0.0107	0.002	200	184
0.25	0	0.1	100	
-0.75	0	1	100	
-1.25	0	1	100	

-1.927199 0 .0005 100 MANDET

-1.25636 0.38032 .08 100 MANDET

-1.749057 0.000306 .04 100 MANDET

-0.7489 0.1073 0.004 100 MANDET

Parameter values for MANDET and MANDIS of interesting areas to be magnified have been supplied in Lauwerier(1994, 1996(with timings)) and have been copied in the accompanying table, with some values added by me, prompted by Fractalus.

The timing of Lauwerier's most expensive detail with centre (-0.74691, 0.10725), width d=0.0008 and maximum of iterations kmax=100,[12] took ≈38s by PSView and ≈40s by Acrobat Pro. Nearly a factor 20 faster than Lauwerier's timings. The included pictures are more impressive when viewed full-screen.

A few stepwise magnifications have been given below. Compared to the professional pictures, even with Peitgen c.s.(1986) of 25 years ago, the 'Museum pictures,' the results are just an amateur's[13] beginnings.

-.7454 0 1.5 100 MANDET

-.7454 0 1 100 MANDET

-.7454 0 .5 100 MANDET

-.7454 0 .25 100 MANDET

Using MANDIS

In order to visualize the hairy details the distance formula is used in MANDISm, where the suffix m denotes monochrome. The first 2 parameters denote c, i.e. a and b (e.g. in 6 decimals). The third parameter is half the width of the fractal domain to be viewed. The 4th parameter is the maximum number of iterations, i.e. kmax, of the inner-loop. The value of the loop variable is used as escape number for determining the colour. The fifth parameter, f, is used as threshold for the closeness to the M-fractal, e.g. .00005.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox:  -400 -300 400 300
%%BeginSetup
%%EndSetup
```

12. 100 is sufficient
13. An allude to G.E. Forsythe's 'A professional starts where an amateur ends.'

```
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run
%%Endprolog
%
%   Program ---the script---
%
-.72 0  1.4 50 .0005 MANDISm
%for the right picture the invoke reads
/Courier 100 selectfont
begintime
-.16 1.03 .025 100 0.00005 MANDISm
-390 -295 moveto (Time:) show
  usertime begintime sub showobject
                 (ms.) show
showpage
%%EOF
```

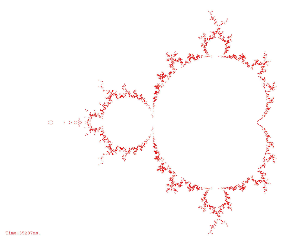
-.72 0 1.4 50 .00005 MANDISm

-.16 1.03 .025 150 .00005 MANDISm

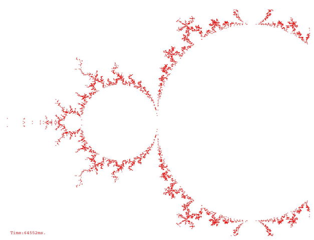
The same mapping as in `MANDET` from the wired-in `BoundingBox -400 -300 400 300` onto the domain of the M-fractal has been used.

Lauwerier(1990, p132–134) discusses the relationship of the quadratic Julia dynamical system with the Verhulst bifurcation diagram. For real c the a-axis reflects the Verhulst behaviour, which stretches out into the full cardioid and the full circle because of complex values of c.

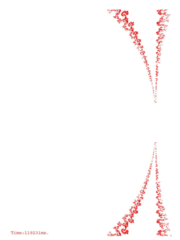
A few stepwise magnifications have been given below, similar to those for `MANDET`.



`-.74 0` 1.5 `100 .00005 MANDISm`



`-.74 0` 1 `100 .00005 MANDISm`



`-.74 0` .25 `100 .00005 MANDISm`

M-cardioid In Courant(1937, p267)[14] the cardioid is mentioned as a special case of the epicycloid, which results when a circle rotates around a circle. The equations for the cardioid, parametric in $\varphi \in [0, 2\pi]$, read

$$x = \cos(\varphi)/2 - \cos(2\varphi)/4$$

$$y = \sin(\varphi)/2 - \sin(2\varphi)/4$$

The cardioid has been draw, with the main circle centre at (-1,0) and radius .25, next to it; see accompanying picture. The cardioid is of the same size as the M-fractal cardioid.

In order to have an idea of the scale in the M-fractal picture the relevant numbers are shown underneath the drawing. The a- and b-axis have been drawn dashed.

If $|4a^2 + 8a + b^2| < 3.75$ then (a,b) lies inside the cardioid.

If $|a + ib + 1| < .25$ then (a,b) lies inside the circle.

Lauwerier(1995) contains a BASIC program for drawing a cardiod, `CARDIO`. My PostScript program for the accompanying M-Cardioid picture is straightforward and has been included in `PSlib.eps`.

Wim W. Wilhelm communicated his compact specification for drawing the cardioid

```
ParametricPlot[{Cos(fi)/2-Cos(2 fi)/4, Sin(fi)/2-Sin(2 fi)/4}, (fi, 0, 2 pi)]
```

I'm not sure whether Lauwerier would have loved these features. I do like the concise Math specifications, progress has been made since the 90s.

14. Courant, R(1937, sec.ed.): Differential and Integral Calculus.

Circle symmetric Julia fractals

In Lauwerier(1996) the research of Field and Golubitsky(1992) is mentioned with respect to circle symmetric fractals. They chose the 2D dynamical system, $z_{n+1} = F(z_n, \bar{z}_n)$ which must obey

$$F(z, \bar{z}) = \bar{c}F(cz, \bar{c}\bar{z})$$

in order that the resulting pseudo-Julia fractal will be circle symmetric, Lauwerier(1996, p122).

Use of JULIAS

The dynamical system used by Field c.s. in `JULIAS` for an m-fold rotation symmetric pseudo-Julia fractal reads, Lauwerier(1996, p129)

$$z_{n+1} = z_n(a + b/z_n^m + c)) \quad \text{with} \quad 1 = a + b/(1 + c).$$

The parameters are `a`, `b`, `N` the number of points, `kmax`, the maximum number of iterations per grid point.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -180 -140 180 140
%%BeginSetup
%%EndSetup
%%BeginProlog
(c:\\PSlib\\PSlib.eps) run%Inclusion of PSlib(rary)
%%EndProlog
%
%   Program, the script
%
1 .75 scale 3 0 105 0 360 arc%ellips
clip                         %clip rest of picture to ellips
.5 .3 200 100 JULIAS         %Circle Symmetric Julia fractal, scaled elliptically
showpage
%%EOF
```

Use of JULIASYMm

Another dynamical system used by Field c.s. in `JULIASYMM` for an m-fold rotation symmetric pseudo-Julia fractal reads, Lauwerier(1996, p124),

$$z_{n+1} = z_n(a + bz_n\bar{z}_n + c\ \mathrm{Re}(z_n^m)) + d\bar{z}_n^{m-1} \quad \text{with} \quad 1 = a + b/(1 + c).$$

The parameters are `a`, `b`, `N` the number of points, `kmax`, the maximum number of iterations per grid point,

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 125 40 515 440
%%BeginSetup
%%EndSetup
%%DocumentFonts: Courier
%%BeginProlog
(c:\\PSlib\\PSlib.eps) run
%%EndProlog
%
%   Program the script
%
-2.08 1 -.1 .167 150 7 5000 JULIASYMm%Mayan bracelet
showpage
%%EOF
```

Example values for the parameters are summarized in the table with the corresponding figures underneath.

Nickname	**a**	**b**	**c**	**d**	**h**	**m**	**col**	**kmax**
Halloween	-2.7	5	1.5	1	250	6	9	7000
Mayan bracelet	-2.08	1	-0.1	0.167	150	7	8	7000
Emperor's cloth	-1.806	0.1806	0	1	250	5	5	7000
Trampoline	1.56	-1	0.1	-0.82	150	3	6	7000
Pentagon	2.6	-2	0	-0.5	150	5	5	10000
Kachina dolls	2.409	-2.5	0	0.9	200	23	8	7000
Pentacle	-2.32	2.32	0	0.75	200	5	8	7000
Golden flintstones	2.5	-2.5	0	0.9	175	3	6	7000

Halloween Mayan bracelet Emperor's cloth Trampoline

Pentagon Kachina dolls Pentacle Golden flintstones

Use of FRACSYMm

The dynamical system used by Field c.s. in FRACSYMm is a linear contraction after which each iterate, z_k, is rotated over $2\pi j/m, j = 0, 1, ...m - 1$.

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad \text{and rotations} \quad z^j_{n+1} = e^{2\pi i\, j/m} z_{n+1} \quad j = 0, 1, ...m-1$$

which leads to a rotation symmetric pseudo-Julia fractal, Lauwerier(1996, p124).

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -250 -250 250 250
%%BeginSetup
%%EndSetup
%%BeginProlog
(c:\\PSlib\\PSlib.eps) run
%%EndProlog
%
%   Program the script
%
%a   b    c  d  e  f sc m  kmax  def-name
-.1 .35  .2 .5 .5 .4 6 175 4000 FRACSYMm
showpage
%%EOF
```

The parameters are the similarity transformation constants a, b, c, d, e, f, the scaling factor, the rotation symmetry, and kmax, the maximum number of iterations per grid point,

The following pictures have been created by FRACSYMm with the parameters given underneath.

-.4 .75 .2 -.3 0 .4 200 55 2000   -.15 .75 .2 -.3 .075 .4 250 50 2000   -.25 -.3 .14 -.26 .5 .5 175 12 3000   .45 -.1 -.31 .45 .1 .2 500 11 3000

.4 -.1 -.35 .4 .01 .2 500 9 3000   .45 -.1 .3 -.4 .15 .1 500 8 3000   -.1 .35 .2 .5 .5 .4 150 6 4000   .5 0 0 .5 .5 0 200 5 4000

Use of some packages from the WWW

"It is my philosophy that not only we should show in our User Group publications what TEX&Co can do for us, but also mention programs, or tools, which perform the same task, in order to choose the best tool for the purpose."

Use of Stuif's Julia previewer Stuif's WWW contains a button for the Julia Mandelbrot applet.

Moving with the cursor over the M-fractal, left, and clicking the mouse yields the corresponding Julia fractal. The pictures at right show J(.325, .417), options: details and the fractal.

Use of Winfract It opens with a window displaying the M-fractal with a menu bar.

Within the context of this note the Mandelbrot/Julia toggling is interesting. It allows to go from a point in the M-fractal to show the corresponding Julia fractal.

Pictures are stored in `.par` format, whatever that means. Pictures can also be saved on the clipboard and opened in Photoshop te be stored in a format of choice.

The help file contains the topics

- ☐ Whats New?
- ☐ File Menu
- ☐ Fractals Menu
- ☐ View Menu
- ☐ Colors Menu
- ☐ Help Menu
- ☐ Fractal Formula Selection
- ☐ Zooming in on an Image
- ☐ Mandelbrot/Julia Toggling
- ☐ Color-Cycling
- ☐ Fractint-Style Help and Prompts
- ☐ Coordinate Box options
- ☐ Limitations in Winfract
- ☐ Distribution Policy, Contacting the Authors, The Book
- ☐ A list of Winfract and Fractint Authors

Use of the XaoS real-time fractal zoomer `XaoS` has been made by Jan Hubicka&Thomas Marsch[15] and has been put in the public domain under the GNU license.

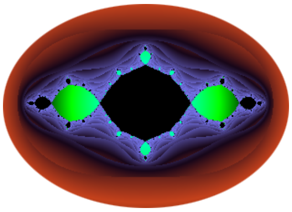
A lot of prefab fractals are available via the menus to experiment with or just to watch in amazement. A picture is saved as `.png` file. There is a tutorial and a help file. Jan Hubicka's movie 'An introduction to Fractals' is fascinating, and gives more than a book ever can provide. I did not find out yet how to circularly crop a picture. Manually altering the window size crops the picture. Zooming goes by the left and right mouse button. In order to obtain the circle-inversion of the Mandelbrot fractal (right picture) one has to select $1/\mu$ through the hierarchy of menus `Fractal` → `Plane` → $1/\mu$. The inverse Mandelbrot has been coloured inside via `incolouring mode`.[16]

Sierpinski Mandelbrot Mandelbrot detail Mandelbrot4 Invers Mandelbrot

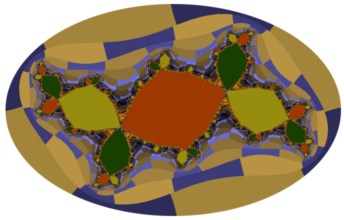
For the variation of colour, second, fourth and fifth picture, there are various options under `incolouring` mode, i.e. inside the fractal, and `outcolouring` mode, i.e. outside the fractal. For example `Fractal` → `Incouloring Mode` → `real/imag`. There is also a pseudo 3D option. In short, a lot of options to experiment with and to obtain coloured fractals, which are not so easy to realize via PostScript, to put it mildly.

Julia fractals via XaoS Within the context of this note, I was interested to specify my own Julia fractals. For fractals to be specified by the user there is the sub-menu `User formula`, see accompanying picture, where the Julia fractal `J(-1, 0)` can be specified by `z^2-1`, and in general the Julia fractal `J(a, b)` by `z^2+{a;b}`, with a the real part and b the imaginary part of $c \in \mathbb{C}$ in $z_{n+1} = z_n^2 + c$.

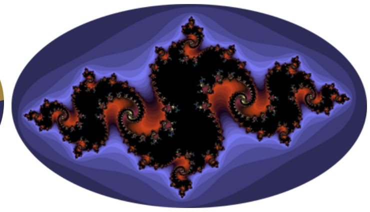
15. Since 1996–2008, Version 3.5
16. An inverse Mandelbrot picture in blue and yellow is used on the cover of Lauwerier(1990).



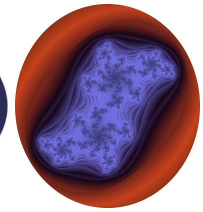
J(.8, 0) San Marco



J(-.12, .74) Douady

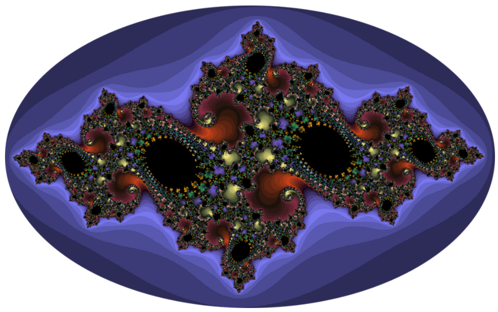


J(-.8, .15) Dragon



J(.11, .66) Dendrit

Without previous experience with `XaoS` it took me roughly 5 hours to produce this section, where the pictures have been cropped by PhotoShop.



J(-.7454, .1103)

Use of the Fractalus package `Fractalus` is a Julia fractal generator for Windows 32- and 64bits, made by Kari Korkeila and available[17] in the public domain under the GNU license.

Features

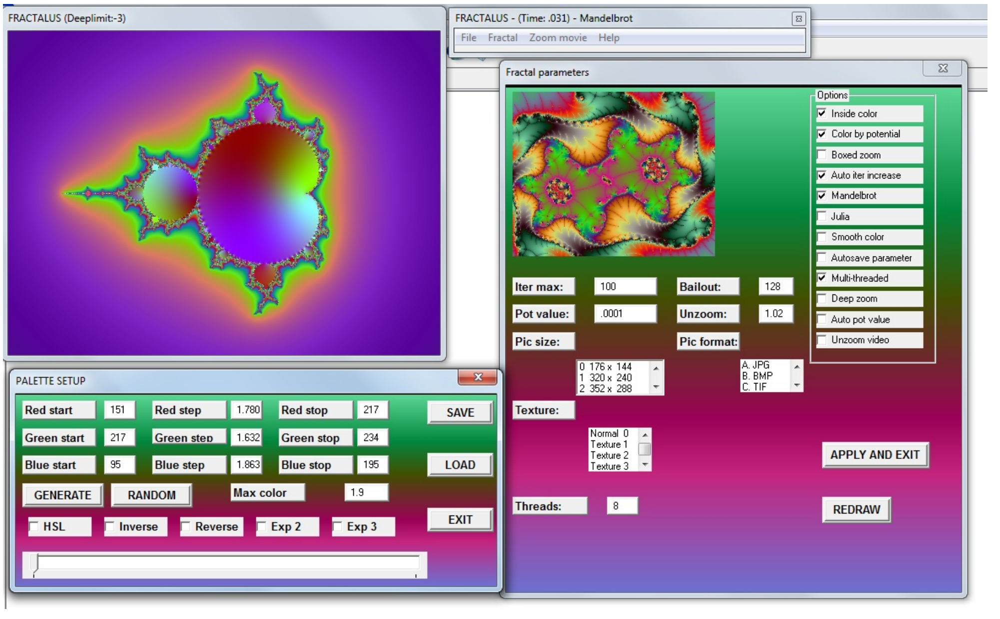
- ☐ fast and small optimized executable (only 760 kb)
- ☐ truecolor rendering
- ☐ saves and loads parameter-files (deep zoom parameters can be loaded with image without re-calculating)
- ☐ saves and loads palettes
- ☐ saves pictures as `.png`, `.tiff`, `.bmp` and `.jpg`; selectable image size from 176x144 to 8192x6140
- ☐ largest image size may not work on all systems - default is `.jpg`
- ☐ Mandelbrot and Julia fractal types currently supported
- ☐ 28 different drawing styles or textures
- ☐ palette manipulation
- ☐ browse fractal parameters with image preview
- ☐ easy zooming and unzooming with right and left mouse-button
- ☐ make zoom movies from fractal parameter files (max resolution 1920x1080) deep zoom also supported
- ☐ make zoom movies from `.jpg` image sequences
- ☐ real-time zoom & unzoom movie (only 80 bit accuracy supported and quite fast machine required)
- ☐ unlimited deep zooming for Mandelbrot and Julia fractals (more than 80 bits used - tested with over 800bit accuracy or over 200 decimals)
- ☐ multi-thread support with multi-core processors
- ☐ portable application (download zip file)
- ☐ graphical Julia finder

17. Since 2011. Current Version 5.751.

The last feature is very interesting within the context of this note: visualization of Julia fractals.

Menus There are 4 menus shown in the picture below. The most relevant for direct use are

- FRACTALUS - Mandelbrot, main menu, with roll-down selection menus.
- FRACTALUS (Deeplimit:-3): displays the fractals, from which the picture can be saved. The parameter value c can requested via the Fractal roll-down menu by the selection of the Location-item.



Quick-start guide from the WWW

- Zoom with left mouse click. Unzoom with right mouse click
- Select palette from menu - click random button until you find good palette
- Save picture and/or as parameter-file.
- For deep zooms some tips
  Use small image size with deep zooms and use boxed zoom.
  Rendering deep zooms can be quite slow.
  Turn on auto-save option when rendering deep zooms.

There is no user guide. Apparently, one just has to play with it.

Generation of Jullia fractals Select in the Fractal roll-down menu the item Julia finder. For the position c of the cursor in the main Mandelbrot window, the Julia fractal $z^2 + c$ will be shown real-time, on-the-fly, in a second window. If you right-mouse click, the Julia fractal will appear in the main window after which it can be saved as .jpg, .bmp, .tif or .png at various sizes, to be selected in the fractal parameters window. In the accompanying figure the white cross denotes the position of the cursor and the small window shows the corresponding Julia fractal. For those who just like the beauty of fractals, like my sister, there is an extensive Gallery of prefab fractals.

Without previous knowledge of `Fractalus` it took me roughly two days to produce this section. Definitely a tool to have at hand.

Annotated References

□ Introductory surveys:
http://en.wikipedia.org/wiki/Dragon_curve
http://www.stuif.com/fractals/index.html
http://en.wikipedia.org/wiki/Julia_set
http://en.wikipedia.org/wiki/Mandelbrot_set
http://en.wikipedia.org/wiki/List_of_fractals_by_Hausdorff_dimension.

□ Adobe Red, Green and Blue Books. The musts for PostScript programmers.

□ Biography of H.A. Lauwerier: http://bwnw.cwi-incubator.nl/cgi-bin/uncgi/alf.

□ Deubert, J: Acumen Journal. http://www.acumentraining.com/acumenjournal.html. Highly educative, with respect to PostScript, PDF and XPS. (From the november2011 issue: PostScript Tech - Transparency in PostScript Using pdfmark PostScript implements a strictly opaque imaging model; objects painted on the page completely obscure anything on the page beneath them. That is, unless you are handing your PostScript file to Distiller; in that case your PostScript code can draw translucent objects on the final PDF page using the Distiller-only pdfmark operator.)

□ Fractal Foundation. http://www.fractalfoundation.org.

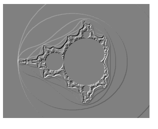
□ Gleisk, J(1987): CHAOS — making a new science. Penguin.
(An introduction to and survey of the world of non-linearity, strange attractors and fractals. The accompanying picture has the following legend: ...The complex boundaries of Newton's method. The attracting pull of four points — in the four dark holes — creates 'basins of attraction,' each of different color, with a complicated fractal boundary. The image represents the way Newton's method for solving equations leads from different starting points to one of four possible solutions (In this case the equation is $z^4 - 1 = 0$) ...)

□ Goossens, M(2007, sec.ed.) c.s.: LATEX Graphics Companion. ISBN 978 0 321 50892 8.

□ Heck, A(2005): Learning MetaPost by Doing. MAPS 32. (A tutorial with nice practical examples.)

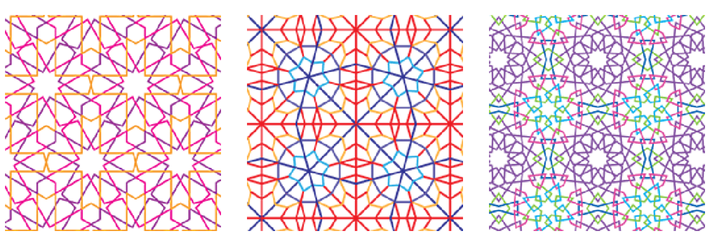
□ Hobby, J.D(1995): Drawing Graphs with MetaPost. CSTR Report 16.

□ Hubicka J, T. Marsch(1996–2008) XaoS3.5 — a real-time fractal zoomer.
(Along with the package comes a fascinating and well-done movie 'An introduction to Fractals.')



□ Jackowski, B, P. Strelczyk, P. Pianowski(1995-2008): PSView5.12.
bop@bop.com.pl. (Extremely fast previewer for .eps and .pdf.
Allows PSlib(rary) inclusion via the run command.
Error messages appear in the pop-up GhostScript window.)

- □ Kari Korkeila(2011): Fractulus — a Fractal generator for Windows. `http:/personal.inet.fi/koti/fractulus`.
- □ Knuth, D.E, T. Larrabee, P.M. Roberts(1989): Mathematical Writing. MAA notes 14. The Mathematical Association of America.
- □ Knuth, D.E(1990, 10th printing): The TEXbook. Addison-Wesley. ISBN 0-201-13447-0. (A must for plain TEXies. I never read a manual so many times. Well ...Adobe's RGB-books come close.)
- □ Kroonenberg, S(2007): Epspdf, easy conversion between PostScript and PDF. EuroBachoTEX.
- □ Lauwerier, H.A(1987): Analyse met de Microcomputer. Epsilon 7.
- □ Lauwerier, H.A(1987): FRACTALS — meetkundige figuren in eindeloze herhaling. Aramith. (Contains BASIC programs. Lauwerier, H.A (1991): Fractals: Endlessly Repeated Geometrical Figures. Translated by Sophia Gill-Hoffstadt, Princeton University Press, Princeton NJ. ISBN 0-691-08551-X, cloth. ISBN 0-691-02445-6 paperback. "This book has been written for a wide audience ... " Includes sample BASIC programs in an appendix. With respect to Julia fractals it contains only the programs JULIAB(acktracking) and MANDEL, for the black-and-white banded Mandelbrot fractal. Intended for instructors, (high-school) students,) and the educated layman.)
- □ Lauwerier, H.A(1988): Meetkunde met de Microcomputer. Epsilon 8.
- □ Lauwerier, H.A(1989): Oneindigheid — een onbereikbaar ideaal. Aramith. ISBN 90 6834 055 7.
  (With respect to Julia fractals it contains the BASIC programs `JULIAS`, `JULIAP` and `MANDELP` the latter is more efficient than the one in Lauwerier(1987). This booklet filled up holes in my knowledge of number systems. The regular continued fraction expansion for ϕ, the golden ratio, with only 1-s, I'll never forget. Audience: Instructors, (high-school) students, and the educated layman.)
- □ Lauwerier, H.A(1990): Een wereld van FRACTALS. Aramith. ISBN 90 6834 076 X. (A sequel and updated version of Lauwerier(1987). Intended for instructors, (high-school) students, and the educated layman.)
- □ Lauwerier, H.A(1992): Computer Simulaties — De wereld als model. Aramith. ISBN 90 6834 106 5. (The last chapter is called Orde en Chaos, and applies to this paper.)
- □ Lauwerier, H.A(1994): Spelen met Graphics and Fractals. Academic Service. ISBN 90 395 0092 4. (An inspiring book with Math at the high-school level for a wide audience. The BASIC programs I consider outdated for direct use. Intended for instructors, (high-school) students, and the educated layman.)
- □ Lauwerier, H.A(1995): Symmetrie, Kunst en Computers. Aramith. (The permutation group P_n is used to classify various symmetries. The possible tilings of the plane by regular polygons are discussed and their mathematical classification is explained. Tilings of Duat and Penrose are included. The Platonic and Archimedic polyhedra are treated. The symmetries in the (pseudo) Julia and Mandelbrot fractal are mentioned. Intended for instructors, (high-school) students, and the educated layman. The BASIC codes are available from the fileserver of the THE.)



□ Lauwerier, H.A(1996): Chaos met de Computer. Epsilon Uitgaven, Utrecht. ISBN 90 5041 043 X4. (Treats the restricted growth model in detail and elaborates on extensions into 2D, among others. Intended for instructors, (high-school) students, and the educated layman.)

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox:  -440 -90 430 345
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run
%%EndProlog
%
% Program
%
lorenzclipped%Lorentz attractor
showpage
%%EOF
```

□ Minsky, M(1969): Form and Content in Computer Science. Turing Award lecture. In: ACM Turing Award Lectures. The first twenty years 1966-1985. ACM Press. (With TEX and similar quality tools for typesetting, I get the impression that pre-press publications within the TEX-community are out of balance: too much attention is paid to form.)
□ Peitgen, H.O, H.Jürgens, D. Saupe(2004 sec.ed.): Chaos and Fractals. New frontiers of Science. (Images of the fourteen chapters of this book cover the central ideas and concepts of chaos and fractals as well as many related topics including: the Mandelbrot set, Julia sets, cellular automata, L-systems, percolation and strange attractors. This new edition has been thoroughly revised throughout. The appendices of the original edition were taken out since more recent publications cover this material in more depth.[18] Instead of the focused computer programs in BASIC, the authors provide 10 interactive JAVA-applets for this second edition via `http://www.cevis.uni-bremen.de/fractals`. An encyclopaedic work. Multiple Reduction Copying Machines, as model for feedback systems, are associated with IFSystems, The Barnsley fern has been elaborated upon and the chaos game is explained. Audience: No mathematical sophistication is required, so a broad audience is aimed at. It portrays the new fields: Chaos and fractals, in an authentic manner. The first edition has been typeset by (La)TEX&Co, the second edition as well, I presume.)
□ Stone Soup Group(1990–2008): `Fractint` and `Winfract` fractal packages. Released in the Public Domain under GNU license.
□ Swanson, E(1986, rev.ed.): Mathematics into Type. American Mathematical Society.
□ Szabó, P(2009): PDF output size of TEX documents. Proceedings EuroTEX2009/ConTEXt, p57–74. (Various tools have been compared for the purpose.)
□ Van der Laan, C.G(1995): Publishing with TEX. Public Domain. (See TEX archives. `BLUe.tex` comes with `pic.dat` the database of pictures in TEX-alone. No Julia fractals, but some strange attractors have been done by TEX-alone. The advantage of TEX alone approach is portability in place and time. All chapters

18. Curious is that the data compression appendix has been taken out while the source still contains blind alleys to the absent appendix; the source has not been adapted, nor are wavelets mentioned, which are used in `.png` and `.jpg` compression. Although the BASIC codes have been taken out, the text is still BASIC biased as can be whitnessed from the attention paid to Turtle Graphics; no recursion which BASIC lacks. The material on the history of the calculation of the digits of π is nice and informative, but a bit out of context.

still compile under pdfTeX without adaptation. The inelegance of the font aspects is inherited from the bitmap-fonts plain TeX-engine. A successor of this work should be based on an OTF-TeX-engine.)

□ Van der Laan, C.G(unpublished, BachoTeX workshop): TeXing Paradigms. (A plea is made for standardized macro writing in TeX to enhance readability and correctness. Topics: Plain's `items` extended, Headache (about headings), Two-part macros. Parametrization I — Options, The wind and hallfwinds (macros for turtle graphics in TeX), It's all in the game — Dialogue with TeX, Loops, Searching, Sorting, Just a little bit of PostScript, FIFO and LIFO sing the BLUes, Syntactic Sugar.)

□ Van der Laan, C.G(2009): TeX Education — an overlooked approach. EuroTeX2009-3rdConTeXt proceedings. (Launched my `PSlib.eps` library.)

□ Van der Laan, C.G(2010): Circle Inversions —with a serious undertone—. MAPS 42. (Contains the solution of Apollonius problem as a PS `def.` `PSlib.eps` is introduced.)

□ Van der Laan, C.G(2011): Gabo's Torsion. MAPS 42. (Contains also a summary of the PostScript language and its developments.)

□ Van der Laan, C.G(2012): Pythagoras Trees in PostScript — Fractal Geometry 0. EuroBachoTeX2012-proceedings (MAPS 44). (Submitted Informatsionnie Texnologii i Matematicheskoe Modelirovanie.)

□ Van der Laan, C.G(2012): Classical Math Fractals in PostScript — Fractal Geometry I. EuroBachoTeX2012-proceedings (MAPS 44). (Submitted Informatsionnie Texnologii i Matematicheskoe Modelirovanie.)

□ Veith, U(2009): Experiences typesetting mathematical physics. Proceedings EuroTeX2009/ConTeXt, p31–43. (Practical examples where we need to adjust TeX's automatic typesetting.)

I don't own any of the following Classical Fractal books but pick up Barnsley(1988, 2006) and Mandelbrot(1982) from the library:[19]

□ Barnsley, M.F(1988): Fractals Everywhere. Academic Service.

(Famous from this book is the IFS for a fern. IFS-s, equivalent dynamical systems as well as the associated fractals are treated within a theoretical framework for Fractal Geometry. The accompanying illustration is used in ch. 3 to illuminate the idea of a contractive transformation on a compact metric space.)

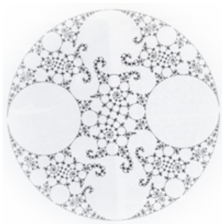
Chapter 7 Julia sets. Chapter 8 Parameter Spaces and Mandelbrot sets. BASIC programs are included: 3.8.1 Example of Deterministic Algorithm; 3.8.2 Random Iteration Algorithm; 6.2.1 Fractal Interpolation; 7.1.1 Example of the Escape Time Algorithm. (No efficiency short-cuts such as use of symmetry in the codes. I could not spot the data reduction process of the Andes Indian girl picture.)

□ Barnsley, M.F(2006): SuperFractals. Patterns of Nature. Cambridge University Press. 464p. (.....Superfractals would be a superb addition to the bookshelves of any scientists who use fractal analysis techniques in their research, be they physicist, biologist or economist. The author concludes by promising that the introduction of superfractals will revolutionize the way mathematics, physics, biology and art are combined, to produce a unified description of the complex world in which we live. After reading this book, I have no doubt that he is correct. From NATURE Vol 445 18 January 2007.)

□ Falconer, K(1997): Techniques in Fractal Geometry. Wiley. ISBN 978-0-471-95724-9. (Peitgen c.s. mention that this book contains an adequate technicaL discussion of the fractal dimension.)

19. The problem with the classics like Barnsley(1988, 2006) is that they contain too much in too advanced Math. It is easier for me to read Lauwerier, who understands the matter and with his magical intuition summarizes the relevant issues in not too sophisticated Math. But ... the pictures in those classics are usually breathtaking.

□ Falconer, K.J(2003, sec.ed.): Fractal Geometry. Mathematical Foundations and Applications. Digital and ordinary book.
□ Field, M, M.Golubitsky(1992): Symmetry in Chaos. Oxford University Press.
□ Field, M, M.Golubitsky(2009): A search for Pattern in Mathematics, Art and Nature, (Symmetry suggests order and regularity whilst chaos suggests disorder and randomness. Symmetry in Chaos is an exploration of how combining the seemingly contradictory symmetry and chaos can lead to the construction of striking and beautiful images. This book is an engaging look at the interplay of art and mathematics, and between symmetry and chaos. The underlying mathematics involved in the generation of the images is described. This second edition has been updated to include the Faraday experiment, a classical experiment from fluid dynamics which illustrates that increasing the vibration amplitude of a container of liquid causes the liquid to form surface waves, instead of moving as a solid body. This second edition also includes updated methods for numerically determining the symmetry of higher dimensional analogues of the images. As well as this, it contains new and improved quality images.)
□ Golubitsky, M(1997): Introduction to nonlinear Dynamical Systems and Chaos. Springer-Verlag.
□ Fischer, Y (): Fractal Image Compression Theory and Applications.
□ Mandelbrot, B(1982); The Fractal structure of Nature. ISBN 07 167 11869. (This essay of erudition, originality and insight, is about the fundamentals of Math: Euclidean geometry and the dimension concept are widened up into fractal geometry and fractal dimension. A plea is made for fractals to describe natural phenomena: clouds, waves, landscapes, length of coasts with the ill-posed question of length answered by: each coastline has a fractal dimension, The 'montruous' curves of 19th century Math find a honourable niche in fractal geometry, with a fractal dimension added.)



□ Peitgen, H.O, P.H. Richter(1986): The Beauty of Fractals. Images of complex dynamical systems. Springer-Verlag. (Frontiers of Chaos. Verhulst Dynamics. Julia Sets and their Computergraphical generation. Sullivan's Classification and Critical Points. The Mandelbrot Set. External Angles and the Hubbard Trees. Newton's method for Complex Polynomials: Cayley's Problem. Newton's method for Real Equations. A discrete Volterra-Lotka System. Magnetism and Complex boundaries. Yang-Lee Zeros. Invited contributions:
Fractals and the Rebirth of Iteration Theory by B. Mandelbrot;
Julia Sets and the Mandelbrot Set by A. Douady;
Freedom, Science, and Aesthetics by G. Eilinger;
Refractions on Science into Art by H.W. Franke.
The book ends with DO IT YOURSELF and DOCUMENTATION.)

Conclusions

It was pleasure, educative and inspiring to read Lauwerier's booklets. Some of his algorithms have found a wider audience by conversion of his BASIC codes into PostScript `defs`, hopefully.

I have no experience in running BASIC programs nor do I know how to include the resulting pictures elegantly in my publications. The EPSF results of PostScript programs can easily and time-proven, be included in my pdf(La)TeX[20], Word, ... documents, or in other PostScript programs.

20. After conversion, alas. `\psfig` functionality has been lost. Happily, ConTEXt and LuaTEX allow EPSF inclusion.

Lauwerier(1990) mentions a lot of applications of fractals in the world around us, to which I did not pay attention in this note, but demonstrates the relevance of studying fractals.

A half year ago, I didn't know how to draw fractals on my PC. Now I have a suite of PostScript `def`s available in my `PSlib.eps` library, and I'm aware of several fractal packages, which also allow drawing user specified Julia fractals, with zoom, colour and transformation facilities.
The relation between the M-fractal and the various Julia fractals is

- □ The M-fractal is a map of Julia fractals. To each point `(a, b)` of the map belongs a Julia fractal `J(a,b)`.
- □ The M-fractal is a bifurcation diagram of Julia fractals.

In the past 15 years nice fractal software has been released in the public domain.

I get once more the impression that I should learn JAVA (Stuif, Peitgen) in order to create animated Fractals, in pursuit of the animated Cabri software for hyperbolic geometry.

Lauwerier, 25 years ago, was more than right in his vision that the PC would become a household tool and could be used for playing with fractals and for experimenting the world around us, as witnessed in Lauwerier(1992).

'Het Wiskunde boek' states that fractals have renewed and raised interest in Mathematics.

Before publishing consult the Wikipedia on aspects of the subject as well as Wolfram's knowledge base `http://www.wolframalpha.com`.

It would be nice to have fractal contours for the line fractals and colouring by gradients.

It is curious how the adherence to structured programming — preferring the MetaPost preprocessor instead of PostScript, casu quo preferring LaTeX instead of minimal plain TeX ...— is overdone in the TeX-community. In the case of neglecting PostScript the champagne is thrown away with the cork,[21] i.e. the powerful transformation of User Space of PostScript is passed by, so is PostScript's more accurate arithmetic, which is of higher accuracy than the arithmetic in MetaPost.[22] The often praised virtue of formal, symbolic solutions of equations in MetaPost, neglects the numerically more stable pivoting strategies.
However ... what about TeX&Java or LaTeX&Asymptote&TeXnicCenter?

Happily, I was not aware of `Fractalus` when I started working on this note, because otherwise I might not have converted Lauwerier's BASIC codes into PostScript.

I don't expect the double precision BASIC codes to translate easily in MetaPost, because of the limited accuracy of the current MetaPost.

Working on this note has widened my horizon of creating pictures by PostScript; I also became aware of the usefulness of the `PSView` previewer, for previewing `.eps` as well as `.pdf`.

21. I drink Moldavian champagne and reuse the corks ànd bottles for my home-made wine and cider.
22. PostScript adopted the accuracy and arithmetic of the underlying computer architecture. The PostScript LRM 3 table B.1 states as accuracy for reals: 8 significant decimal digits. I verified it experimentally and found 10^{-7}.

TEX mark-up For the symbols of the number systems $\mathbb{I}$, $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, which curiously are not provided for in plain TEX, I used the AMS (blackboard) font msbm10. I look out for the day that OpenTypeFonts will be universally available, to start with availability in TEX and PostScript.

In-line verbatim is marked up by starting and ending the verbatim text by a vertical bar, |,borrowed from BLUe.tex, which on its turn learned from the mark-up of the TEXbook source. However, the use of the |-symbol is lost for the mark-up of the absolute value; happily, there is still the more verbose alias \vert.

I have used in the appendices vbox-s next to each other for parallel listings of the program texts — BASIC and its converted PostScript def — which inhibits proper page breaks, but doesn't harm in the appendices. I don't know how to provide macros for local elegantly marked up multi-column texts in \vbox-s, which allow for proper page breaks. How to cope with page-breaks and for example pictures is a well-known problem. Usually the picture is floated on the page or placed on the next page. TEX provides insertion commands to support floating, e.g. the 2-part macro \topinsert and \endinsert, TEXbook p115–116. Ill-placement of figures is taken for granted. Another approach is to bundle the (colour) pictures in a special quire, as applied by Lauwerier.

Inserting (large) .pdf pictures lead to problems with the Acrobat viewer. As far as I understand it the .pdf figure has to be processed by Acrobat before insertion. Beforehand converting the pictures into reasonable-sized .jpg solved the problems, slowness, with viewing by Acrobat. Processing by TEXworks became faster as well.

The table where as function of the Malthus parameter α the various kinds of bifurcations are summarized is typeset by straightforward use of \halign. The tables of pictures are not typeset by \halign nor by the tabbing mechanism, because I found it more flexible just to use box-s. For typesetting of the tables with vertically aligned decimal dots, I had to refresh my knowledge of \halign. For the typesetting of vertically aligned decimal dots I did not follow the approach of the TEXbook p241, where each leading non-significant zero is hidden by the ?-mark-up. I used 2 columns: one for the integer part and the other for the fractional part, which is a bit of error-prone mark-up, however, … once used to it … . As a consequence the text in the heading over these 2 columns is centred by using \span instead of &. The separation of the heading from the table date I did by \noalign{<verticalmaterial>}, the TEXbook, p237, or using a \strut. The \halign template has \sevenstruts for more pleasing lines in the data part. The font used for the tables is 7pt.

The picture of the grid is not created by use of TEX's \leaders, but created in simple straight minimal PostScript, because then the picture can be more easily reused. The labels form an integral part of the picture: an all-in-one self-contained picture. The picture of the Cardioid is created in PostScript, where the arrow heads of the coordinate axes are done by arrow from Adobe's Blue Book, p138–139.

Because \eqalign allows only '2 columns' I modified \eqalign for BLUe.tex into a variant which allows more 'columns,' and used that variant as well.

The simple bar-chart-like graphs for the convergence behaviour of the restricted growth model have been created in straight minimal PostScript as well; no MetaPost Graph macros needed.

The $\stackrel{\rm L}{=}$ composed relational operator is marked up by \mathrel {\mathop =^{\rm L}} and not by $\buildrel\rm L \over=$, TEXbook p437, which is OK when no surrounding Math-relational-operator space is needed.

I don't like the ill-placing of floated pictures. My inserted pictures suffer from the same inconvenience as in Word: changing the text might disturb the layout, such that the pictures will become ill-placed. I use 2 methods for placing pictures: the first one is a \line with 2 \vbox-s, one for the text and the other for the picture. The other creates white-space at the right margin via the use of \hangindent and \hangafter, TEXbook p102, which adheres more to minimal mark-up and is more general than my earlier adapting the value of `\hsize` within a group ended by `\par`. The pictures are inserted in the blank space via the TEXnique mentioned in TEXbook Appendix D p389, by my macros `\insertpdf`, casu quo `\insertjpg`. The mark-up is highly similar to the mark-up of the dangerous bends in the TEXbook. These dangerous bends are just marked up by `\danger`, which takes care of providing the open space by using `\hangindent` and `\hangafter`, and places at the open space the picture, which is stored as character in the `manual` font (See theTEXbook Appendix B for the macro `\danger`).

The non-automatically font scaling within context of TEX is a nuisance. I was caught by that Math symbols in the abstract and footnotes don't comply with 7pt. I had to switch explicitly to `\seveni`. Maybe, I should write a Math scaling, to be applied when needed.

For the circle symmetric fractals Acrobat 7 performed slow, very slow. `PSView` previewed just by a fillip, but ... without yielding a `.pdf`-file. `MANDET` gave an overflow message in Acrobat, while `PSView` still showed the result.

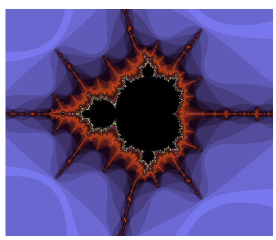
TEXworks2010 fell silent when working on this note, crashed an unacceptable number of times with the message: `Assertion failed.`?!? Installation of TEXworks 2011 solved the problem.

Pictures for the slides are in `.png` format with transparent background; I could not obtain transparency via `.jpg` format.

For the slides with blue-coloured Math text I had to insert when using pdfTEX `\pdfliteral{0 0 0 0 k}` for blue-coloured text as well as `\pdfliteral{0 0 0 0 K}` for blue-coloured lines, e.g. to get for example the horizontal line in the enlarged sqrt-symbol or the horizontal bar in `\over` blue-coloured. Weird, very weird.[23] 21st-century software unworthy.

Finally, as an aside a plea for the developers of the successor of TEX

> "Successors of TEX should be freed from the wired-in discrete fonts, and embrace scalable, context-aware fonts."



PostScript Working on this note has deepened my knowledge of PostScript, though there is still much to learn. For example, I don't know as yet how to use PostScript's `pathbbox`, which prompts the actual BB, such that an exact cropped picture can be obtained, in a 1-pass job.

It is a pity that pdfTEX does not allow for pictures in PostScript.[24]

Advanced use of colour-gradients is on my mind, maybe some time, some day ...

My PostScript programming reflects the procedural style of programming of the 60s of the last century; 21st Century programming wraps procedural codes into a package structure, with access by interactive menus, abstracting from the various computer operating systems.

---

23. In 2009 I already experienced this inconvenience with commutative diagrams.
24. 15 Years ago I used psfig to include my PS-pictures in TEX documents. It is a pity that pdfTEX did not build on this tradition. Happily, LuaTEX and ConTEXt allow .eps inclusion.

Acknowledgements

Thank you Adobe for your maintained, adapted to LanguageLevel 3 since 1997, good old, industrial standard PostScript and Acrobat Pro (actually DISTILLER) to view it, Don Knuth for your stable plain TEX, Jonathan Kew for the TEXworks IDE, Hàn Thế Thành for pdf(La)TEX, Hans Lauwerier for your nice, educational booklets with so many inspiring examples and clear Math exposé.

Thank you Nice Temmme for the reference to the AMS notices of 2012.

Thank you Jos Winnink for proofing a near-finished version and for your as usual valuable comments, such as the mark-up of tables in smaller corps, and providing for the first example of use a variant with the library def, and what is needed more, included. Wim W. Wilhelm for your comments and for your example of the M-fractal in Asymptote, your modern vector-graphics tool as successor of MetaPost. Wim's parametric-plots tool looks handy to me. Henk Jansen for detailed proofing and his suggestion for the word quire, as well as the phrase 'throw away the champagne with the cork.' My sister, Martha van der Laan, for drawing my attention to XaoS, which she uses from an artistic viewpoint. MAPS editors for improving my use of English and last but not least Taco Hoekwater for procrusting my plain TEX note into MAPS format.

IDE My PC runs 32 bits Vista, with Intel Quad CPU Q8300 2.5GHz assisted by 8GB RAM. I visualize PostScript with PSView and convert into .pdf via Acrobat Pro 7.[25] My PostScript editor is just Windows 'kladblok (notepad).' I use the EPSF-feature to crop pictures to their BoundingBox, ready for inclusion in documents. For document production I use TEXworks IDE with the plain TEX engine, pdfTEX, with as few as possible structuring macros taken from BLUe.tex — adhering minimal TEX mark-up. I use the Terminal font in the edit window with the pleasing effect that comments remain vertically aligned in the .pdf window.

For checking the spelling I use the public domain en_GB dictionary and hyphenation patterns en_GB.aff in TEXworks.[26]

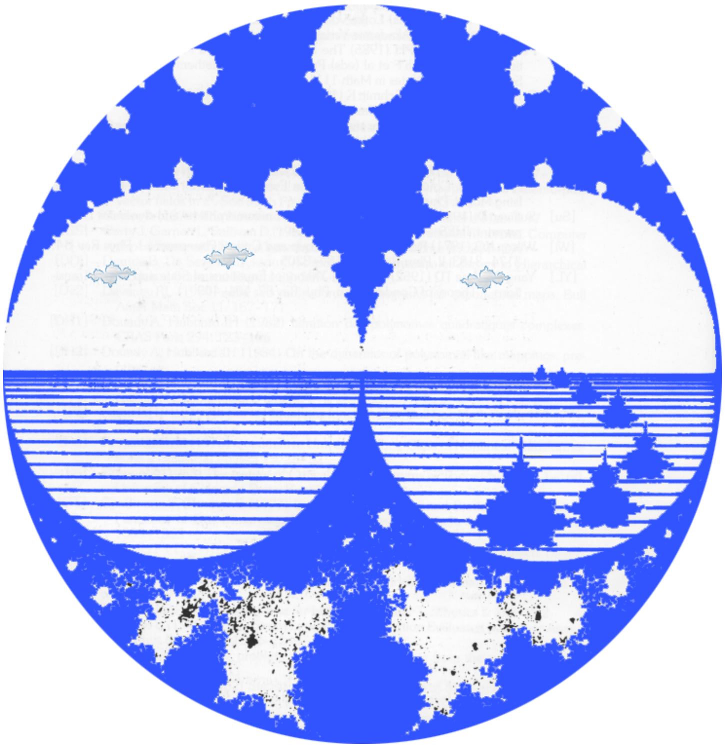
Prior to sending my PDF's by email the files are optimized towards size by Acrobat Pro.

The bad news with respect to .eps into .pdf conversion is, that the newest Acrobat 10 Pro X does not allow for the run command for library inclusion. Photoshop did not accept .eps M-cardiod (Could not ...). AI-CS5 showed the picture full-page, not cropped to the BB. The result could be saved as .pdf, .svg. Maybe I should buy Jaws PDFcreator, which is advised by John Deubert. The trial version seems to do what I want, also handles library inclusion. But ... it does not crop. For the time being I'll try it out, taking their watermark stamp for granted.

My case rests, have fun and all the best.

Kees van der Laan
kisa1@xs4all.nl

25. PSView is extremely fast as previewer, allows PS library inclusion via the run command as well, gives error message via a GhostScript window, but ... doesn't provide for .pdf output, alas.
26. Why not deliver TEXworks on the TEX Collection DVD with the dictionaries already inserted? Agreed, inserting them is easy.



Mandelbrot's view of 'Breskens'

Appendix: Lauwerier's BASIC codes into PostScript

Lauwerier used overtime various variants of BASIC: starting with the GW-BASIC interpreter in the 80s and ending with the PowerBASIC compiler in the 90s.

BASIC screen settings and interrupting commands have been neglected in the conversion, because PostScript is batch-oriented and abstracts from the various screens and printer devices. The artefacts `XM=320` and `YM=200` in the BASIC codes reflect the screen size of those days 640×400 pixels.

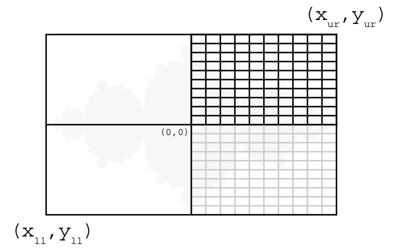
Single precision is used in PostScript throughout.

The real part and imaginary part of the problem constant $c \in \mathbb{C}$ are called `A` and `B` in BASIC and provided as parameters to the `defs` in PostScript; they are stored in the local dictionary as `a` and `b`.

BASIC uses `SIN` and `COS` with the angle given in radians; PostScript assumes angles in degrees.

`SQR` converts into `sqrt`.

Instead of `DELH` and `DELV` (halfwidths) the equivalent x_{ur} and y_{ur}, coordinates of right-upper corner of the fractal domain, are used as variables for the point-symmetric rectangular which contains the fractal associated with c, because these quantities are more easily associated with PostScript's `BoundingBox`, which is just the scaled fractal domain. In PostScript these values are provided as arguments to the `defs` and stored locally as x_{ur} and y_{ur}. However, in `MANDET` and `MANDIS`, for details of the M-fractals, I have chosen for a fixed-sized `BoundingBox`, similar to Lauwerier's screen-size, where the various magnifications are shown.



The nodes of the grid, which cover the 'first quadrant' of the rectangular which contains the fractal, are tested in nested loops on whether they belong to the JULIA set or not. If $b \neq 0$ the nodes of the 4^{th} quadrant of the rectangle also have to be tested. The other nodes of the fractal domain are point-symmetric with the tested nodes.

From the inner-loop Lauwerier escapes pragmatically by an escape `GOTO`, for testing the next node, when the iterated value is of no longer use. The escape translates directly into `exit` from the (loop)procedure as argument of PostScript's `loop`. The maximum number of iterations of the inner loop is stored in the variable `KMAX` in BASIC, which is an argument to the `def` in PostScript, and stored locally as `kmax`.

The test criterion is method-dependent.

For colours Lauwerier used a permutation array `COL`, such that he could select a few colours, e.g. 10, in a convenient order from the 16 standard colours in BASIC.

Standard in BASIC: 0=black, 1=blue, ... , 14=yellow, 15=bright white

Lauwerier's order of selection: 0=black, 1=blue, 2=light blue, ... , 8=brown, 9=yellow.

```
REM ***Colours***
DIM COL(8) : DATA 0,0,1,9,4,12,6,14,2
FORKI=0 TO 8 : READ COL(K) : NEXT K
REM Colours are selected by assignment of an integer to L
PSET( I,J), L ; PSET( I,-J), L
```

```
/Courier 14 selectfont
/colours [/black /black /blue /lightblue
         /red /lightred /brown /yellow /green] def
%Colours are selected by assignment of an integer to l
colours l get cvx exec
i j moveto (.) centershow i j neg moveto (.) centershow
```

For `PSET(i,j)` the dots of the `Courier` font at 12pt have been used and are centered.

In the BASIC codes for $j = 0$, the x-axis, the pixels are printed twice; in PostScript this yields too black results. In the converted codes this double printing is avoided. For printing the pixels `centershow` from `PSlib.eps` has been used.

Appendix: JULIAMC into a PostScript library def

The program performs `kmax+10` inverse iterations of which the first 10 are neglected.

Inverse Iteration: $x_{k-1} = (x_k - a)/2 \pm \sqrt{((x_k - a)/2)^2 + ((y_k - b)/2)^2}$ $k = \text{kmax+10}, \ldots, 1$

$y_{k-1} = (y_k - b)/(2x_{k-1})$

`RND < .5` is translated into `rand 1073741823 gt`, because PostScript's random generator, `rnd`, yields an integer in the range 0 to $2^{31} - 1$, maximum integer (Mersienne's number), while BASIC's `RND` yields a number in the range 0 to 1.

```
A=.3 : B=0
KMAX=40000: K=0 : X=RND : Y=RND
DO WHILE K<KMAX AND INKEY$="" 'interrupt possibility
   X1=(X-A)/2 : Y1=(Y-B)/2 : R=SQR(X1*X1+Y1*Y1)
   IF RND < .5 THEN
      X=SQR(R+X1) : Y=SQR(R-X1)
      IF Y1<0 THEN Y=-Y
   ELSE
      X=-SQR(R+X1) : Y=-SQR(R-X1)
      IF Y1<0 THEN Y=-Y
   END IF
   IF K>10         THEN PSET( X,Y) : PSET(-X,-Y)
   IF K>10 AND B=0 THEN PSET(-X,Y) : PSET( X,-Y)
   K=K+1
LOOP
END
```

```
/JULIAMC{%stack: a, b, maxk.
         %a+ib complex constant c,  maxk maximal iterations
         % ==> Julia set of z^2 + a + ib
JULIAMCdict begin /Courier 12 selectfont
/kmax exch def /b exch def /a exch def
/nextpoint{/x1 x a sub 2 div def /y1 y b sub 2 div def
            x1 dup mul y1 dup mul add sqrt dup%R
           /y exch x1 sub sqrt y1 0 lt{neg}if def
           /x exch x1 add sqrt def
}bind def
/printxy{x s y s        moveto (.) centershow
    x s neg y s neg     moveto (.) centershow %point symmetry
    b 0 eq y 0 ne and
       {x s y s neg moveto (.) centershow
        x s neg y s moveto (.) centershow}if  %symmetry x- and y-axes
}bind def
/s {100 mul} def                              %scaling
/nrand rand 2147483647 div def                %random number in [0,1]
/x nrand def /y nrand def                     %start values in [0,1]
10{nextpoint}repeat                           %discard 10 iterations
kmax{nextpoint
     rand 1073741823 gt {/x x neg def /y y neg def}if
     printxy
    }repeat
end}bind def
%
/JULIAMCdict 11 dict def                      %local dictionary
```

In his first book on Fractals Lauwerier also provided for the backtracking method, by pursuing both points of the inverse iteration. Not needed.

Appendix: JULIABS into a PostScript library def

The program tests the size of $z_k^2 + c$ for $k = 0, 1, 2, 3, \ldots, kmax$ at the 4 corners of a (small) square around each node of the 'grid.' When not at all 4 cornerpoints the sequence z_n goes to ∞, (4 in the program) i.e. the sequence z_n stays finite, then the grid point is considered to belong to the boundary of the fractal, because on the boundary (and inside the contour) the sequence of points remain on the fractal (or stay finite).

```
REM***JULBS of z*z+c, Boundary Scan
A=0 : B=.65
KMAX=80 : DELV=1.3 : DELH=1.3
N1=200: N2=INT(N1*DELV/DELH)
IF B=0 THEN N3=0 ELSE N3=N2
FOR I=0 TO N1 : FOR J=-N1 TO N2
   T=0 : IF INKEYS$<>"" THEN END
   X=(I+.5)*DELH/N1 : Y=(j-.5)*DELV/N2 : GOSUB jcycle
   X=(I+.5)*DELH/N1 : Y=(j+.5)*DELV/N2 : GOSUB jcycle
   X=(I-.5)*DELH/N1 : Y=(j-.5)*DELV/N2 : GOSUB jcycle
   X=(I-.5)*DELH/N1 : Y=(j+.5)*DELV/N2 : GOSUB jcycle
   IF T=0 OR T=4 THEN GOTO repeat
   PSET (I,J), 14 ; PSET (-I,-J), 14
   IF B=0 THEN PSET (I,-J), 14 ; PSET (-I,J), 14
repeat:
NEXT J : NEXT I
END
jcycle:
FOR K-1 TO KMAX
   X1=X*X-Y*Y+A : Y1=2*X*Y+B
   IF X1*X1+Y1*Y1>4 THEN T=T+1 : EXIT FOR
   END IF
   X=X1 : Y=Y1
NEXT K
RETURN
END
```

```
/JULIABS{%Stack: a b (a+ib in C) x_ur y_ur kmax
         %==> fractal
JULIABSdict begin /Courier 12 selectfont
/kmax exch def /y_ur exch def /x_ur exch def
/b exch def /a exch def
/step .01 def /hstep step 2 div def /sc {100 mul} def
/print-xsc-ysc{xsc     ysc     moveto (.) centershow
         xsc neg ysc neg moveto (.) centershow%point symmetry
         b 0 eq {xsc     ysc neg  moveto (.) centershow
                 xsc neg ysc      moveto (.) centershow}if
        }def
/jcycle{/k 0 def
        {x dup mul y dup mul add 4 gt{/t t 1 add def exit}if
         /xn x dup mul y dup mul sub a add def
         /y 2 x mul y mul b add def /x xn def
         /k k 1 add def
         k kmax eq {exit}if
        }loop
       } bind def
0 step x_ur{/xgrid exch def /xsc xgrid sc def  %for i, loop over grid
   b 0 eq{0}{y_ur neg}ifelse step y_ur
     {/ygrid exch def /ysc ygrid sc def%for j
      /t 0 def
      /x xgrid hstep add def
      /y ygrid hstep sub def jcycle
      /x xgrid hstep add def
      /y ygrid hstep add def jcycle
      /x xgrid hstep sub def
      /y ygrid hstep add def jcycle
      /x xgrid hstep sub def
      /y ygrid hstep sub def jcycle
      t 0 ne t 4 ne and {print-xsc-ysc}if
   }for %j
}for %i
end}bind def
%
/JULIABSdict 40 dict def
```

Appendix: JULIAF into a PostScript library def
The forward iteration is implemented. When the iterate goes to ∞ then the iterate is skipped; when after kmax iterations the iterate stays finite, < 100 say, the point is considered to be part of the JULIA fractal or inside the fractal.

```
REM***JULIAFill of z*z+c, Pixel method***
REM**Lauwerier(1994, p142)***
A=-.8 : B=.15 'Dragon filled
KMAX=200 : DELV=1 : DELH=1.6
N1=250: N2=INT(N1*DELV/DELH)
IF B=0 THEN N3=0 ELSE N3=-N2
FOR I=0 TO N1 : FOR J=N3 TO N2
   X=I*DELH/N1 : Y=J*DELV/N2
   FOR K=1 TO KMAX
      X1=X*X-Y*Y+A: Y1=2*X*Y+B : S=X*X+Y*Y
      IF S>1000 THEN GOTO repeat
      X-X1 : Y=Y1
   NEXT K
   PSET (I,J) ; PSET (-I,-J)
   IF B=0 THEN PSET (I,-J) ; PSET (-I,J)
repeat: NEXT J : NEXT I
END
```

```
/JULIAF{%Stack: a b x_u y_u kmax
        %       c=a+ib, window of fractal, maximum iterates
        %==> Filled JULIA fractal
JULIAFdict begin /Courier 12 selectfont
/kmax exch def /y_u exch def /x_u exch def
/b exch def /a exch def
/pr{x sc y sc moveto (.) centershow
    x sc neg y sc neg moveto (.) centershow
    b 0 eq{x sc y sc neg moveto (.) centershow
           x sc neg y sc moveto (.) centershow}if}def
/sc {100 mul} def
/step 0.01 def
0 step x_u{/x exch def                                   %grid
   b 0 eq{0}{y_u neg}ifelse step y_u{/y exch def /z_i y def /z_r x def
      /k 0 def
      {/aux z_r dup mul z_i dup mul sub a add def%new iterate
       /zn_i 2 z_r mul z_i mul b add def /zn_r aux def
       z_r dup mul z_i dup mul add 100 gt{exit}if          %infty, outside
       zn_r z_r sub dup mul zn_i z_i sub dup mul add .0001 lt{pr exit}if
       /k k 1 add def
       k kmax eq{pr exit}if
       /z_r zn_r def /z_i zn_i def                         %inside
      }loop
   }for %j
}for %i
end}bind def
%
/JULIAFdict 22 dict def
```

Appendix: JULIAD into a PostScript library def

A powerful forward iteration program which iterates over $z_{n+1} = z_n^2 + c$ and the derivative $s_{n+1} = 2s_n z_n$. JULIADist can be used advantageously where other programs fall short.

At the heart lies the distance formula, Lauwerier(1990, 1996)[27]

$$d(z_0, J) \approx |z_n| \log |z_n| / |\frac{dz_n}{dz}|.$$

A curious, compact formula where $d(z_0, J)$ is expressed in $|z_n|$ and $|\frac{dz_n}{dz}|$.

My Complex Analysis skills have become a bit rusty after so many years of non-use, so I take Lauwerier's formula for granted.

```
REM***Lauwerier(1996) p112***
REM*** JULIA with distance formula***
A=-.55 : B= .15 'c=a+ib
DELH=2 : DELV= 1
N1=240 : N2=INT(N1*DELV/DELH) : F=.6
IF B=0 THEN N3=0 ELSE N3=-N2
FOR I=0 TO N1
   FOR J=N3 TO N2
     IF I=0 AND J=0 THEN GOTO nextpoint
     X=I*DELH/N1 : Y= J*DELV/N2 : u=1 : V=0
     FOR K=0 TO 200
     X1=X*X-Y*Y +A : Y1=2*X*Y+B
     U1=2(U*X-V*Y) : V1=2*(U*Y+V*X)
     S1=X1*X1+Y1*Y1+1E-10: S2=LOG(S1)
     S3=SQR(U1*U1+V1*V1+1E-10)
     IF S1>256 OR S3>256 THEN
        DIST=SQR(S1)*S2/S3
        IF DIST<F*DELH/N1 THEN
           PSET(I,J), 14 : PSET(-I,-J), 14
           IF B=0 THEN PSET(I,-J), 14 : PSET(-I,J), 14
           END IF GOTO nextpoint
        END IF
     END IF
     x=X1 : Y=Y1 : U=U1 : V=V1
     NEXT K
nextpoint: NEXT J : NEXT I
END

J(.1,-.8) =>
```

```
/JULIAD{%Stack: a b x_ur y_ur kmax f
        % c, window of fractal, maximum iterates, distance threshold
        %==>JULIA fractal by distance formula
JULIADdic begin /Courier 12 selectfont
/f exch def/kmax exch def /y_ur exch def /x_ur exch def
/b exch def /a exch def
/step .01 def /sc {100 mul} def %/f .001 def%thickness
0 step x_ur{/xl exch def /xsc xl sc def
 b 0 eq {0}{y_ur neg}ifelse step y_ur{/y exch def /x xl def
  /ysc y sc def
  /u 1 def /v 0 def
  /k 0 def
   {/k k 1 add def
    k kmax eq {exit}if
      /x1 x dup mul y dup mul sub a add def
      /y1 2 x mul y mul b add def
      /u1 2 u x mul v y mul sub mul def
      /v1 2 u y mul v x mul add mul def
      /s1 x1 dup mul y1 dup mul add def
      /s2 s1 0.00001 add log def
      /s3 u1 dup mul v1 dup mul add .1 add sqrt def
      s1 256 gt s3 256 gt or
       {/dist s1 sqrt s2 mul s3 div def
        dist f lt
        {xsc ysc moveto (.) centershow
         xsc 0 ne{xsc neg ysc neg moveto (.) centershow}if
         b 0 eq{xsc ysc neg moveto (.) centershow
                xsc neg ysc moveto (.) centershow}if
        }if
        exit
       }if
       /x x1 def /y y1 def /u u1 def /v v1 def
    }loop%k
 }for%j
}for%i
end}bind def
%
/JULIADdict 31 dict def
```

27. Formula in Lauwerier(1996) contains the log of the derivative; an error. Lauwerier(1990) is better.

Appendix: JULIAP into a PostScript library def

JULIAP, appended P stands for Pixel method, performs the forward iteration. The process is based on the property that when z_0 is outside the Julia fractal the iterate goes to ∞ and when z_0 belongs to the fractal it remains on the fractal in a chaotic way. The value of the inner-loop variable is maintained and when the value of the iterate is greater than 100, say, the index of the iterate, called escape number, is associated with a colour and the initial point, z_0, is coloured by this colour. If the initial point is on the Julia fractal or converges to an attractor inside the fractal then the loop control variable is again associated with a colour and the initial point, z_0, coloured by this colour. Either $k = kmax$, and z_0 was on the fractal already, or when $k < kmax$ on the escape fulfilled the test by the Cauchy criterion, the sequence converged to an attractor. The points z_0 are taken systematically from the set of nodes of the $2N_1 - 1 \times 2N_2 + 1$ grid laid over the rectangle with left lower corner (`-delh`, `-delv`) and right upper corner (`delh`, `delv`). The user must know beforehand that the fractal lies within the rectangle and supplies `delh` and `delv` as parameters to the invoke of `JULIAP`.

```
REM***Lauwerier(1996, p148):  Graphics and Fractals***
A=-.55 : B= .15 'c=a+ib
DELH=2 : DELV=1.8'adapted
N1=200 : N2=INT(N1*DELV/DELH)
IF B=0 THEN N3=0 ELSE N3=-N2
FOR I=0 TO N1
   FOR J=N3 TO N2
     X=I*DELH/N1 : Y= J*DELV/N2
     FOR K=0 TO 200
     X1=X*X-Y*Y +A : Y1=2*X*Y+B
     S =X*X+Y*Y : S2=LOG(S1)
     S1=(X-X1)*(X-X1)+(Y-Y1)*(Y-Y1)
     IF S  >100 THEN L=1 +k mod 15 : GOTO graphics
     IF S1<.001 THEN L=1 +k mod 15 : GOTO graphics
     x=X1 : Y=Y1
   NEXT K: L=0
graphics: PSET(I,J), L : PSET(-I,-J), L
          IF B=0 THEN PSET(I,-J), L : PSET(-I,J), L
   NEXT J : NEXT I
END
```

```
/JULIAPdict 25 dict def
%
/JULIAP%On stack: a b, x_ur y_ur kmax
       %          c, right-upper of window of fractal, maximum iterates
       %==>Banded coloured Fractal
{JULIAPdict begin /Courier 12 selectfont
/kmax exch def /y_ur exch def /x_ur exch def /b exch def /a exch def
/sc {100 mul} def /step 0.01 def
/colours [/black /blue /lightblue /green /lightgreen
           /red /lightred /brown /yellow] def
0 step x_ur{/xgrid exch def /xsc xgrid sc def
 b 0 eq{0}{y_ur neg}ifelse step y_ur{/y exch def /ysc y sc def
  /x xgrid def
  /k 0 def /l 0 def
  {/k k 1 add def
    k kmax eq {exit}if
      /x1 x dup mul y dup mul sub a add def
      /y1 2 x mul y mul b add def
      /s x1 dup mul y1 dup mul add def
      /s1 x x1 sub dup mul y y1 sub dup mul add def
      s   100 gt {/l k 9 mod def exit}if
      s1 .001 lt {/l k 9 mod def exit}if
      /x x1 def /y y1 def
   }loop%k
   colours l get cvx exec
   xsc ysc moveto (.) centershow
   xsc neg ysc neg moveto (.) centershow
   b 0 eq {xsc ysc neg moveto (.) centershow
           xsc neg ysc moveto (.) centershow
          }if
 }for%j
}for%i
end}bind def
```

Appendix: JULIAS into a PostScript library def

The dynamical system for an m-fold rotation symmetric fractal reads, Lauwerier(1996, 129), borrowed from Field&Golubitsky(1992))

$$z_{n+1} = z_{n+1}(a + b/(z^m + c)) \quad \text{with} \quad 1 = a + b/(1 + c).$$

The parameters of `JULIAS` are `a`, `b`, `m` and `kmax`.

```
REM ***Kleuren***
DIM COL(8) : DATA 0,0,1,9,4,12,6,14,2
FOR I=0 TO 8 : READ COL(I) : NEXT I
M=8 : A=.5 : B=.35: C=(A+B-1)/(1-A) : N=200
R=3.5 'Radius circle window
FOR I=0 TO N : FOR J=0 TO I
   X=I*R/N : Y=J*R/N
   IF X*X+Y*Y >R*R THEN  GOTO repeat
   FOR K=1 TO 200
      X2=X*X-Y*Y : Y2=2*X*Y
      X4=X2*X2-Y2*Y2 : Y4=2*X2*Y2
      X8=X4*X4-Y4*Y4 : Y8=2*X4*Y4
      XN=X8+C : YN=Y8
      S=XN*XN+YN*YN+1E-8
      XT=XN/s : YT=-YN/S
      X1=A+B*XT :Y1=B*YT
      X0=X*X1-Y*Y1 : Y0=X*Y1+Y*X1
      S0=X0*X0+Y0*Y0 : S1=(X8-1)*(X8-1)+Y8*Y8
        IF S0<.0001 OR S1<1E-08 THEN
           L=COL(1+K mod 8) : GOTO graphics
        END IF
      X=X0 :Y=Y0
   NEXT K : L=0
graphics:
   PSET( I,J), L ; PSET( I,-J), L
   PSET(-I,J), L ; PSET(-I,-J), L
   PSET( J,I), L ; PSET( J,-I), L
   PSET(-J,I), L ; PSET(-J,-I), L
repeat:
NEXT J : NEXT I
END
```

```
/JULIAS{%On stack: a b n kmax, n is n-fold symmetry
        %==> Circle Symmetric Julia fractal
JULIASdict begin /Courier 12 selectfont
/kmax exch def /n exch def /b exch def /a exch def
/colours [/black /black /blue /lightblue
          /red /lightred /brown /yellow /green] def
/c a b add 1 sub 1 a sub div def
/r 3.5 def
0 1 n{/i exch def
 0 1 i{/j exch def
  /x i r mul n div def
  /y j r mul n div def
  /k 0 def /l 0 def
  x dup mul y dup mul add r dup mul lt
  {%\type{z}<\type{r}
  {/k k 1 add def
    k kmax gt {exit}if %exit k-loop
    /x2 x dup mul y dup mul sub def
    /y2 2 x mul y mul def
    /x4 x2 dup mul y2 dup mul sub def
    /y4 2 x2 mul y2 mul def
    /x8 x4 dup mul y4 dup mul sub def
    /y8 2 x4 mul y4 mul def
    /xn x8 c add def
    /yn y8 def
    /s xn dup mul yn dup mul add .00001 add def
    /xt xn s div def
    /yt yn neg s div def
    /x1 a b xt mul add def
    /y1 b yt mul def
    /x0 x x1 mul y y1 mul sub def
    /y0 x y1 mul y x1 mul add def
    /s0 x0 dup mul y0 dup mul add def
    /s1 x8 1 sub dup mul y8 dup mul add def
    s0 .0001 lt s1 .00000001 lt or{/l k cvi 8  mod def exit}if
    /x x0 def
    /y y0 def
   }loop %next k
   colours l get cvx exec
   i j         moveto (.) centershow
   i j neg     moveto (.) centershow
   i neg j     moveto (.) centershow
   i neg j neg moveto (.) centershow
   j     i     moveto (.) centershow
   j     i neg moveto (.) centershow
   j neg i     moveto (.) centershow
   j neg i neg moveto (.) centershow
   }if%\type{z}<\type{r}
}for%j
}for%i
end}bind def
%
/JULIASdict 35 dict def
```

Appendix: JULIASYMm into a PostScript library def

The requirement on the 2D dynamical system $F(z,\bar{z})$ in order that the strange attractors are rotation symmetric, reads

$$F(z,\bar{z}) = c * F(cz, \bar{c}\bar{z}) \quad \text{with} \quad c^m = 1 \quad \bar{c} = 1/c \quad \text{the conjugate.}$$

For $F(z,\bar{z})$ Field&Golubitsky(1992) used

$$F(z,\bar{z}) = z(a + bz\bar{z} + cz^m) + d\bar{z}^{m-1}.$$

```
A=2.5 : B=-2.5: C=0 : D=.9 : H=100 : N=6 :
XM=320 :  YM=240  : KMAX=2000 :
K=0 :  L=14 : PI=4*ATN(1)
DIM C(N), S(N)
FOR I=0 TO N-1
   C(I)=COS(2*PI*I/N) : S(I)=SIN(2*PI*I/N)
NEXT I
X=RND/10 : Y= RND/10             'start
DO WHILE K<KMAX
FOR I=0 TO N-1 : FOR J=0 TO I
   S=X*X+Y*Y : X1=X : Y1=Y
   FOR I=1 TO N-2                'Z^N
      X2=X*X1-Y*Y1 : Y2=X*Y1+Y*X1
      X1=X2 : Y1=Y2
   NEXT I
   XN=X*X1-Y*Y1
   P=A+B*S+C*XN
   XNEW=P*X+D*X : YNEW=P*Y-DY1:
   XH=H*XNEW : YH=H*YNEW         'scaling
   FOR J=1 TO N-1    'rotation
      XHJ=C(J)*XH-S(J)*YH
      XHJ=S(J)*XH+C(J)*YH
      PSET(XM+XHJ, YM+YHJ), L
   NEXT J
   X=XNEW : Y=YNEW
   K=K+1
LOOP: BEEP
END
```

```
/JULIASYMm{%On stack: a b c d h n xm ym kmax
          %=> circle symmteric fractal
JULIASYMmdict begin /Courier 12 selectfont
/kmax exch def /ym exch def /xm exch def
/n exch def /h exch def
/d exch def /c exch def /b exch def /a exch def
/co n array def /si n array def
0 1 n 1 sub{/i exch def
   co i 360 i mul n div cos put
   si i 360 i mul n div sin put
   }for
22121943 srand
/x rand 21474836470 div def
/y rand 21474836470 div def
/k 0 def
{/k k 1 add def
 k kmax gt{exit}if
 /s x dup mul y dup mul add def
 /x1 x def /y1 y def
 1 1 n 2 sub{/i exch def
    /x2 x x1 mul y y1 mul sub def
    /y2 x y1 mul y x1 mul add def
    /x1 x2 def /y1 y2 def
    }for%i
 /xn x x1 mul y y1 mul sub def
 /p a b s mul add c xn mul add def
 /xnew p x mul d x1 mul add def
 /ynew p y mul d y1 mul sub def
 /xh h xnew mul def
 /yh h ynew mul def
 0 1 n 1 sub{/j exch def
    /xhj co j get xh mul si j get  yh mul sub def
    /yhj si j get xh mul co j get  yh mul add def
    xm xhj add ym yhj add moveto (.) centershow
    }for%j
 /x xnew def
 /y ynew def
 }loop%k
end}bind def
%
/JULIASYMmdict 35 dict def
```

Appendix: FRACSYMm into a PostScript library def

```
A=-.1 : B=.4: C=.2 : D=.5 : E=.5 F=.4
SC=175 : M=7 : KMAX=4000
K=0 : : L=14
DIM C(M), S(M)
FOR I=0 TO M-1
   C(I)=COS(2*PI*I/N) : S(I)=SIN(2*PI*I/N)
NEXT I
X=RND/10 : Y= RND/10             'start
DO WHILE K<KMAX
  Z=X : X=A*X+B*Y+E : Y=C*Z+C*Y+F
  FOR J=0 TO M-1
    X1=C(J)*X-S(J)*Y : Y1=S(J)*X+C(J)*Y
    IF K>32THEN PSET(SC*X1, SC*Y1), L
  NEXT J
  N=INT(M*RND) : K=K+1
  X=C(N)*X1-S(N)*Y1
  Y=S(N)*X1+C(N)*Y1
LOOP
END
```

```
/FRACSSYMm{%Stack: a b c d e f m sc kmax
          %==> Cirle Symmetric Jula fractal by ITS
FRACSYMmdict begin /Courier 12 selectfont
/kmax exch def /sc exch def /m exch def
/f exch def /e exch def /d exch def
/c exch def /b exch def /a exch def
/co m array def /si m array def
0 1 m 1 sub{/i exch def
   co i 360 i mul m div cos put
   si i 360 i mul m div sin put
   }for
22121943 srand
/x rand 21474836470 div def
/y rand 21474836470 div def
/k 0 def
{/k k 1 add def
 k kmax gt{exit}if
 /z x def
 /x a x mul b y mul add e add def%transform
 /y c z mul d y mul add f add def
 0 1 m 1 sub{/j exch def         %rotate
    /x1 co j get x mul si j get y mul sub def
    /y1 si j get x mul co j get y mul add def
    k 32 gt{ x1 sc mul y1 sc mul moveto (.) centershow}if
    }for %j
/n m rand 2147483647 div mul cvi def
/x co n get x1 mul si n get y1 mul sub def
/y si n get x1 mul co n get y1 mul add def
}loop%k
end}def
%
/FRACSYMmdict 25 dict def
```

Appendix: Lauwerier(1987, p156) Mandelbrot program

This section contains the program which in fact yields the answer to the question posed in the introduction:

> "For which values of c will the Julia fractal, J(c), be line-like and for which dust-like?"

I not only 'converted' his first code for the Mandelbrot set, but also improved his coding, made it straight and better readable, IMHO.

From the Mandelbrot Wikipedia I borrowed the following pseudo-code, which clearly exhibits the structure of an M-fractal code,[28]without efficiency short-cuts.

```
For each pixel on the screen do:
x0 = scaled x coordinate of pixel (must be scaled to lie somewhere in the mandelbrot X scale (-2.5 to 1)
y0 = scaled y coordinate of pixel (must be scaled to lie somewhere in the mandelbrot Y scale (-1, 1)
x = 0  y = 0
iteration = 0   max_iteration = 1000
while ( x*x + y*y < 2*2  AND  iteration < max_iteration )
  {xtemp = x*x - y*y + x0  y = 2*x*y + y0   x = xtemp
   iteration = iteration + 1}
color = iteration  plot(x0, y0, color)
```

Purpose of he program Visualize in the (a,b)-plane the points (a,b) for which J(a,b) is connected.

Property If $\lim_{k\to\infty} z_k(a,b) = \infty$ with $z_0 = (0,0)$ then J(a, b) is dust-like.

```
100 p1= -2.5 : p2=-1.5 : p3=1.5 : p4=1.5
110 N1=250 : N2=INT(.833*N1*(P4-P2)/(P3-P1)}
120 FOR I=-N1 TO N1 : A=(({N1-I)*P1+(N1+I)*P3/(2*N1)
130     FOR J=0 TO N2 : B=(({N2-J)*P1+(N2+J)*P3/(2*N2)
140         X=A : Y=B
150         FOR K=0 TO 50
160         Z=X : X=X*X-Y*Y+A
170               y=2Y*Z+B
180         IF X*X+Y*Y>16 THEN GOTO 200
190        NEXT K
200        PSET(320+I, 200-J), K MOD 2
205        PSET(320+I, 200+J), K MOD 2
210     NEXT J
220 NEXT I
230END
```

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Mandelbrotzw filled, of z^2+a+ib
%%Author: H.A. Lauwerier(1987): Fractals, p156
%%Creator: Kees van der Laan, June 2012
%%BoundingBox:  -400 -300 400 300
%%BeginSetup
%%EndSetup
%%BeginProlog
/man1987{%==> Mandelbrot fractal
man1987dict begin /Courier 12 selectfont
/sc {400 mul} def                                 %scaling for output
/pr{a sc b sc moveto (.) centershow
    a sc b sc neg moveto (.) centershow          %symmetry
    b 0 ne{a sc b sc neg moveto (.) centershow}if}def
/a_l -1.75 def /b_l -1 def /a_u .5 def /b_u 1 def%domain M-fractal
/kmax 50 def
a_l .01  a_u{/a exch def                         %over grid points
   0 .01 b_u{/b exch def
      /x a def /y b def /k 0 def
       {/z x def                                 %loop k; z auxiliary
        /x x dup mul y dup mul sub a add def     %real part z^2+a+ib
        /y 2 y mul z mul b add def               %imag part z^2+a+ib
        x dup mul y dup mul add 16 gt{exit}if    %\type{z_k}>16?(infinite)
        /k k 1 add def
        k kmax eq {pr exit}if                    %z_kmax=finite
       }loop
   }for%j
}for%i
end}bind def
%
/man1987dict 16 dict def
%%Endprolog
%
%   Program ---the script---
%
man1987 showpage
%%EOF
```

28. Or see the code of Wim W. Wilhelm, given earlier.

The black figure denotes the points for which J(a,b) is line-like, also called connected.

Coding Improvements

- □ the naming of the domain of the $M_{a,b}$-fractal, with $a \in [a_l, a_u]$ and $b \in [b_l, b_u]$;
- □ improved bounds for the $M_{a,b}$-fractal domain
- □ the loop for points on the a-axis starts with a_l and ends by a_u, the loop for points on the b-axis starts with 0 and ends by b_u
- □ the coordinates for printing are scaled values of the loop control variables a and b
- □ the dots are centered
- □ no fancy k MODE 2 is used for black-and-white bands, nor coloured bands
- □ `%%BoundingBox:` a_l b_l a_u b_u, each quantity scaled by `sc`
- □ the confusion between z_0 and (a, b) as starting value is avoided; better is to talk about the universal starting value $z_0 = 0$, which entails $z_1(a, b) = a + ib$
- □ the calculation of $\sqrt{a^2 + b^2}$, the 2-norm of (a,b), is protected against unnecessary intermediate overflow
- □ `kmax` varies in the various programs, subjective.

In Lauwerier(1989) the program `MANDELP` is more efficient because the inner (k-)loop is skipped for points inside the cardioid or inside the circle, $C_{(-1,.25)}$.

In Lauwerier(1994, 1996) the Mandelbrot program has been further improved by the use of the distance formula which yields better (hairy) details of the M-fractal. We will come back on both aspects in the next appendix `MANDELx`.

Appendix: MANDISm into a PostScript library def
At th heart lies the distance formula, Lauwerier(1996,116) similar as in JULIAD

$$d(c, M) \approx |p_n| \log |p_n| / |\frac{dp_n}{dc}|.$$

```
XM=320 : YM=240 'half old screen size
DELH=1.35 : DELV=1.1 : AC=-.65
N1=200 : N2=INT(N1*DELV/DELH)
FOR I=-N1 TO N1 : A=AC+I*DELH/N1
 FOR J=0 TO N1 : B=J*DELV/N2
   S1=4*(A*A+B*B) : S2=S1-2A+1/4
   IF S1+8*A+15/4<0 THEN COL=14 : GOTO graphics
   IF S2-SQR(S2)+2*A-1/2<0 THEN COL=14 : GOTO graphics
   X=A : Y=B : U=1 : V=0
   FOR K=0 TO 100
     X1=X*X-Y*Y+A : Y1=2*X*Y+B
     U1=2*(U*X-V*Y)+1 : V1=2(U*Y+V*X)
     W1=X1*X1+Y1*Y1
     X=X1 : Y=Y1 : U=U1 : V=V1
     DIS=SQR(W1)*LOG(W1)/SQR(U1*U1+V1*V1)
     IF W>64 THEN
        IF DIS>.4*DELH/N1 THEN COL=0 ELSE COL=14
              GOTO graphics
        END IF
  NEXT K : COL=14
graphics:
  PSET(XM+I,YM-J), COL : PSET(XM+I,YM+J), COL
 NEXT J
NEXT I
END
```

```
/MANDISm{%stack: ac bc del kmax f
         %(centre of detail, width, max number iterations distance
         % parameter
         %==> mandelbrot fractal by distance formula monochrome
MANDISmdict begin /Courier 12 selectfont
/f exch def /kmax exch def /del exch def /bc exch def /ac exch def
/pr{i j moveto (.) centershow}def
/n 400 def %mimics fixed  BB window
n neg 1 n{/i exch def
  -.75 n mul 1 .75 n mul{/j exch def
     /a ac i n div del mul add def
     /b bc j del mul n div add def
     /x a def /y b def /u 1 def /v 0 def
     /k 0 def%loop over k
     {/x1 x dup mul y dup mul sub a add def
      /y1 2 x mul y mul b add def
      /u1 2 u x mul v y mul sub mul 1 add def
      /v1 2 u y mul v x mul add mul def
      /s1 x1 dup mul y1 dup mul add  .0000001 add def
      s1 128 gt
      {/s2 s1 .0000001 add log def
       v1 abs 1 gt
         {/s3 u1 v1 div dup mul 1 add sqrt v1 abs mul def}
         {/s3 v1 u1 div dup mul 1 add sqrt u1 abs mul def}ifelse
       /dist s1 sqrt s2 mul s3 div def
       dist f lt {red pr}if
       exit
      }if
      /x x1 def /y y1 def /u u1 def /v v1 def
      /k k 1 add def
      k kmax eq {exit}if
     }loop%k
   }for%j
}for%i
end}bind def
%
/MANDISmdict 30 dict def
```

Appendix: MANDELx into a PostScript library def MANDEL and variants

```
DELH=1.6 : DELV=1.34 : AC=-.65
N1=260 : N2=INT(N1*DELV/DELH)
DIM COL(8) : DATA 0,1,9,2,10,4,1,6,14
FOR I=0 TO 8 : READ COL(I) : NEXT I
FOR I=-N1 TO N1 : A=AC+I*DELH/N1
   FOR J=0 TO N1 : B=J*DELV/N2
       U=4*(A*A+B*B) : V=U-2*A+1/4
       IF U+8*A+15/4<0 THEN L=0 : GOTO repeat
       IF V-SQR(V)+2*A-1/2<0 THEN L=0 : GOTO repeat
       X=A : Y=B : K=0
       DO
         Z=X : X=X*X-Y*Y+A : Y=2*Z*Y+B
         S=X*X+Y*Y : K=K+1
       LOOP UNTIL S>100 OR K=50
       IF K<40 THEN L=1+K MOD 8 ELSE L=0
       IF K>3 THEN
          PSET(I,J), COL(L) : PSET(I,-J), COL(L)
       END IF
repeat:
NEXT J
NEXT I
END
```

```
/MANDEL%==>Mandelbrotfractal in coloured bands
{MANDELdict begin /Courier 12 selectfont
/colours [/white /blue /lightblue /green /lightgreen
          /red /lightred /brown /yellow] def
/step .01 def /sc {100 mul} def
-2.1 step .85{/a exch def /asc a sc def
   0 step 1.35 {/b exch def /bsc b sc def
     /u 4 a dup mul b dup mul add mul def
     /v u 2 a mul sub .25 add def
     u 8 a mul add -3.75 le           %exclude cardioid
     v v sqrt sub 2 a mul add .5 le or%exclude circle
     {/l 0 def}%inside white, do nothing
     {/x a def /y b def /k 0 def
         {%loop over k
          /z x def
          /x x dup mul y dup mul sub a add def
          /y 2 z mul y mul b add def
          /s x dup mul y dup mul add def
          /k k 1 add def
           s 100 gt k 50 eq or{exit}if
         }loop%k
       k 40 lt{/l k 8 mod def}{/l 0 def}ifelse
       colours l get cvx exec
       k 3 gt{asc bsc     moveto (.) centershow
              asc bsc neg moveto (.) centershow}if
     }ifelse
   }for%j
}for%i
end}bind def
%
/MANDELdict 20 dict def
```

Variants

```
/MANDELzw%==>Mandelbrot fractal in black and white bands
{MANDELzwdict begin /Courier 12 selectfont /l 0 def
/step .01 def /sc {200 mul} def
-2.1 step .85{/a exch def /asc a sc def
   0 step 1.37 {/b exch def /bsc b sc def
     /u 4 a dup mul b dup mul add mul def
     /v u 2 a mul sub .25 add def
     u 8 a mul add -3.75 ge            %exclude cardioid
     v v sqrt sub 2 a mul add .5 ge and%exclude circle
     {/x a def /y b def /k 0 def
         {%loop over k
          /z x def
          /x x dup mul y dup mul sub a add def
          /y 2 z mul y mul b add def
          /s x dup mul y dup mul add def
          /k k 1 add def
           s 100 gt k 50 eq or{exit}if
         }loop%k
       k 40 lt{/l k 8 mod def}{/l 1 def}ifelse
       l 2 idiv 2 mul l eq{
       k 3 gt{asc bsc     moveto (.) centershow
              asc bsc neg moveto (.) centershow}if}if
     }if
   }for%j
}for%i
end}bind def
%
/MANDELzwdict 20 dict def
```

```
/MANDELzwcontour{%stack:
%==> contour mandelbrot fractal by distance formula in black
%    and white
MANDELzwcontourdict begin /Courier 12 selectfont
/pr{i j moveto (.) centershow i j neg moveto (.) centershow}def
/f .005 def/kmax 50 def /del 1.4 def /ac -.75 def
/n1 400 def /n2 300 def                  %size BB window
/thickness .8 def
n1 neg 1 n1{/i exch def
    /a ac del i n1 div mul add def
   0 1 n2{/j exch def
     /b j n2 div .85 del mul mul def
     /s1 4 a dup mul b dup mul add mul def
     /s2 s1 2 a mul sub .25 add def
     s1 8 a mul add -3.75 ge             %exclude cardioid
     s2 s2 sqrt sub 2 a mul add .5 ge and%exclude circle
     {/x a def /y b def /u 1 def /v 0 def
       /k 0 def                          %loop over k
       {/x1 x dup mul y dup mul sub a add def
        /y1 2 x mul y mul b add def
        /u1 2 u x mul v y mul sub mul 1 add def
        /v1 2 u y mul v x mul add mul def
        /w1 x1 dup mul y1 dup mul add def
        /x x1 def /y y1 def /u u1 def /v v1 def
        /dis w1 sqrt w1 log mul
            u1 dup mul v1 dup mul add sqrt div def
        w1 64 gt{dis f lt {pr} if exit}if
       /k k 1 add def k kmax eq {exit}if
      }loop%k
     }if
   }for%j
}for%i
end}bind def
%
/MANDELzwcontourdict 30 dict def
```

Appendix: MANDET into a PostScript library def and black-and-white variant

```
INPUT"x coordinate centrum = ", AC '2
INPUT"y coordinate centrum = ", BC '0
INPUT"halve breedte = ", DEL : CLS '000049
N1=200 : N2=INT(.75*N1) : KMAX=100
DIM COL(8) : DATA 0,1,9,2,10,4,12,6,14
FOR I=0 TO 8 : READ COL(I) : NEXT I
FOR I=-N1 TO N1 : A=AC+I*DELH/N1
   FOR J=-N2 TO N2 : B=BC+.75*J*DEL/N2
       X=A : Y=B: K=0
       DO
         Z=X : X=X*X-Y*Y+A : Y=2*Z*Y+B
         S=X*X+Y*Y : K=K+1
       LOOP UNTIL S>100 OR K=KMAX
       IF K<40 THEN L=1+K MOD 8 ELSE L=0
       PSET(I,J), COL(L)
repeat:
   NEXT J
NEXT I
END
```

```
/MANDET{%stack: a b del kmax
        %==> Mandelbrot fractal detail at (a,b)
MANDETdict begin /Courier 12 selectfont
/colours [/black /blue /lightblue /green /lightgreen
          /red /lightred /brown /yellow] def
/kmax exch def /del exch def /bc exch def /ac exch def
/n1 400 def /n2 n1 .75 mul cvi def %Mimics BoundingBox
n1 neg 1 n1{/i exch def
   /a ac i del mul n1 div add def
   n2 neg 1 n2{/j exch def
     /b bc j del mul n1 div add def
     /x a def /y b def /k 0 def
     {%loop over k
       /z x def
       /x x dup mul y dup mul sub a add def
       /y 2 z mul y mul b add def
       /s x dup mul y dup mul add def
       /k k 1 add def
       s 100 gt k kmax eq or{exit}if
     }loop%k
     k 40 lt{/l k 8 mod def}{/l 0 def}ifelse
     colours l get cvx exec
     i j moveto (.) centershow
   }for%j
}for%i
end}bind def
%
/MANDETdict 21 dict def
```

Black-and-white variant

-1.256 .3817 .005 50 MANDETzw

-1.749057 .000306 .05 50 MANDETzw

```
/MANDETzw{%stack: a b del kmax
        %==> mandelbrot fractal detail at (a,b)
MANDETzwdict begin /Courier 12 selectfont
/kmax exch def /del exch def /bc exch def /ac exch def
/n1 400 def /n2 n1 .75 mul cvi def
n1 neg 1 n1{/i exch def
   /a ac i del mul n1 div add def
   n2 neg 1 n2{/j exch def
     /b bc j del mul n1 div add def
     /x a def /y b def /k 0 def%loop over k
     {/z x def
      /x x dup mul y dup mul sub a add def
      /y 2 z mul y mul b add def
      /s x dup mul y dup mul add def
      /k k 1 add def
      s 100 gt k kmax eq or {exit}if
     }loop%k
     k 40 lt{/l k 8 mod def}{/l 0 def}ifelse
     l 2 idiv 2 mul l eq {i j moveto (.) centershow}if
   }for%j
}for%i
end}bind def
%
/MANDETzwdict 21 dict def
```

Appendix: Use of colours in PostScript

Simple use of colours in PostScript is explained. Colours appear opaque, not transparent. For transparent use of colours in PostScript, see Acumen Journal of Nov 2011. For colour gradients see the LRMlevel 3.

RGB-colour tables

In 2010 I found it useful to provide for color tables, such that one can see the result of each combination of colour parameters, despite the discrepancy between what is seen on the screen and what will result in print (Courtesy: Heck A (2005)). In MHO the supporting use of colours should not be that critical. For artists precise colours matter, of course.

It is curious that the LaTeX Graphics Companion does not contain colour tables, nor does mention the standard names for colours.

RGB-colour names

In `PSlib.eps` I have included the pre-settings of the rgb-colours with their names; handy. A snapshot for the parameters for `setrgbcolor` and their standard names is included below, next to an example of how to select colours by number, with irrelevant details omitted.

```
%Defs: selection of pre-settings of rgbcolors from PSlib.eps
/black    {0  0  0 setrgbcolor} def
/white    {1  1  1 setrgbcolor} def
/red      {1  0  0 setrgbcolor} def
/green    {0  1  0 setrgbcolor} def
/blue     {0  0  1 setrgbcolor} def
/greenblue{0  .7 1 setrgbcolor} def
```

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Selecting colours by number
%%Author: Kees van der Laan, May 2012
%%BoundingBox: 0 -5 110 15
(c:\\PSlib\\PSlib.eps) run%invoke library with colour pre-settings
/colours [/red /green /blue] def%array of colour pre-settings
%
0 0 moveto 100 0 lineto
colours 0 get cvx exec stroke showpage%Red line will show up

%%EOF
```

Use: `r g b setrsbcolor`, with each `r g b` a number in the range 0 to 1. Even simpler use: the mnemonic `greenblue`, e.g.

CMYK-colour tables

Standard CMYK-colour names

In `PSlib.eps` I have also included the settings of the cmyk-colours with their names; handy. A snapshot of the pre-settings for `setcmykcolor` and their standard names is included below.

```
% Procedures: colors
% procedures: Cmyk values  for use in PS a la pdfTeX
/cmykGreenYellow{0.15 0 0.69 0}def
...
/cmykGray{0 0 0 0.50}def
/cmykBlack{0 0 0 1}def
/cmykWhite{0 0 0 0}def
% procedures for cmyk colors
/GreenYellow{ cmykGreenYellow setcmykcolor } def
...
/Gray{ cmykGray setcmykcolor } def
/Black{ cmykBlack setcmykcolor } def
/White{ cmykWhite setcmykcolor } def
```

Use: `c m y k setcmykcolor`, with for each `c m y k` a number in the range 0 to 1. Even simpler use: the mnemonic `GreenYellow`, for example.

For my coloured slides, I could not precisely match the background colour in the pictures with the background colour of the slides, otherwise than with transparent pictures.