fractint5

COLLABORATORS

	TITLE : fractint5		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 11, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

fractint5

1.1 Fractals and the PC

```
A Little History:

    Before Mandelbrot
    Who Is This Guy, Anyway?

A Little Code:

    Periodicity Logic
    Limitations of Integer Math (And How We Cope)
    Arbitrary Precision and Deep Zooming
    The Fractint Fractal Engine Architecture

A Little Math:
    Summary of Fractal Types
    Inside=bof60|bof61|zmag|period
    Inside=epscross|startrail
    Finite Attractors
    Trig Identities
    Quaternion and Hypercomplex Algebra
```

1.2 Common Problems

Of course, Fractint would never stoop to having a "common" problem. These
notes describe some, ahem, "special situations" which come up occasionally
and which even we haven't the gall to label as "features".

```
Hang during startup:
  There might be a problem with Fractint's video detection logic and your
  particular video adapter. Try running with "fractint adapter=xxx" where
  xxx is cga, ega, egamono, mcga, or vga.  If "adapter=vga" works, and you
  really have a SuperVGA adapter capable of higher video modes, there are
  other "adapter=" options for a number of SuperVGA chipsets – please see
  the full selection in Video Parameters for details.  If this solves the
  problem, create an SSTOOLS.INI file with the "adapter=xxx" command in it
  so that the fix will apply to every run.
```

Another possible cause: If you install the latest Fractint in say
directory "newfrac", then run it from another directory with the command
"/newfrac/fractint", *and* you have an older version of fractint.exe
somewhere in your DOS PATH, a silent hang is all you'll get. See the
notes under the "Cannot find FRACTINT.EXE message" problem for the reason.
Another possibility: try one of the "textsafe" parameter choices
described in Video Parameters.

Scrambled image when returning from a text mode display:
  If an image which has been partly or completely generated gets partly
  destroyed when you return to it from the menu, help, or the information
  display, please try the various "textsafe" parameter options – see
  Video Parameters for details. If this cures the problem, create
  an SSTOOLS.INI file with the "textsafe=xxx" command so that the fix will
  apply to every run.

"Holes" in an image while it is being drawn:
  Little squares colored in your "inside" color, in a pattern of every
  second square of that size, in solid guessing mode, both across and down
  (i.e., 1 out of 4), are
  a symptom of an image which should be calculated with more conservative
  periodicity checking than the default. See the Periodicity parameter
  under Image Calculation Parameters.

Black bar at top of screen during color cycling on 8086/8088 machines:
  (This might happen intermittently, not every run.)
  "fractint cyclelimit=10" might cure the problem. If so, increase
  the cyclelimit value (try increasing by 5 or 10 each time) until the
  problem reappears, then back off one step and add that cyclelimit value
  to your SSTOOLS.INI file.

Other video problems:

  If you are using a VESA driver with your video adapter, the first thing
  to try is the "vesadetect=no" parameter. If that fixes the problem, add
  it to your SSTOOLS.INI file to make the fix permanent.

  It may help to explicitly specify your type of adapter – see the
  "adapter=" parameter in Video Parameters.

  We've had one case where a video driver for Windows does not work
  properly with Fractint. If running under Windows, DesqView, or some
  other layered environment, try running Fractint directly from DOS to see
  if that avoids the problem.
  We've also had one case of a problem co-existing with "386 to the Max".

  We've had one report of an EGA adapter which got scrambled images in
  all modes until "textsafe=no" was used (see Video Parameters).

  Also, see Video Adapter Notes for information
  about enhanced video modes – Fractint makes only limited attempts to
  verify that a video mode you request is actually supported by your
  adapter.

Other Hangs and Strange Behavior:
  We've had some problems (hangs and solid beeps) on an FPU equipped

machine when running under Windows 3's enhanced mode. The only ways
around the problem we can find are to either run the Fractint image
involved outside Windows, or to use the DOS command "SET NO87=nofpu"
before running Fractint. (This SET command makes Fractint ignore your
fpu, so things might be a lot slower as a result.)

Insufficient memory:
  Fractint requires a fair bit of memory to run. Most machines with at
  least 640k (ok sticklers, make that "PC-compatible machines") will have
  no problem. Machines with 512k and machines with many TSR utilities
  and/or a LAN interface may have problems. Some Fractint features
  allocate memory when required during a run. If you get a message about
  insufficient memory, or suspect that some problem is due to a memory
  shortage, you could try commenting out some TSR utilities in your
  AUTOEXEC.BAT file, some non-critical drivers in your CONFIG.SYS file, or
  reducing the BUFFERS parameter in your CONFIG.SYS.

"Cannot find FRACTINT.EXE" message:
  Fractint is an overlayed program – some parts of it are brought from
  disk into memory only when used. The overlay manager needs to know
  where to find the program. It must be named FRACTINT.EXE (which it is
  unless somebody renamed it), and you should either be in the directory
  containing it when you start Fractint, or that directory should be in
  your DOS PATH.

"File FRACTINT.CFG is missing or invalid" message:
  You should either start Fractint while you are in the directory
  containing it, or should have that directory in your DOS PATH variable.
  If that isn't the problem, maybe you have a FRACTINT.CFG file from an
  older release of Fractint lying around? If so, best rename or delete it.
  If that isn't the problem either, then the FRACTINT.CFG included in the
  FRAINT.EXE release file has probably been changed or deleted. Best
  reinstall Fractint to get a fresh copy.

Some other program doesn't like GIF files created by Fractint:
  Fractint generates nice clean GIF89A spec files, honest! But telling
  this to the other program isn't likely to change its mind. Instead, try
  an option which might get around the problem: run Fractint with the
  command line option "gif87a=yes" and then save an image. Fractint will
  store the image in the older GIF87A format, without any fractal
  parameters in it (so you won't be able to load the image back into
  Fractint and zoom into it – the fractal type, coordinates, etc. are not
  stored in this older format), and without an "aspect ratio" in the GIF
  header (we've seen one utility which doesn't like that field.)

Disk video mode performance:
  This won't be blindingly fast at the best of times, but there are things
  which can slow it down and can be tuned. See Disk-Video Modes
  for details.

1.3 before mandelbrot

Like new forms of life, new branches of mathematics and science don't
appear from nowhere. The ideas of fractal geometry can be traced to the

late nineteenth century, when mathematicians created shapes -- sets of
points -- that seemed to have no counterpart in nature. By a wonderful
irony, the "abstract" mathematics descended from that work has now turned
out to be MORE appropriate than any other for describing many natural
shapes and processes.

Perhaps we shouldn't be surprised. The Greek geometers worked out the
mathematics of the conic sections for its formal beauty; it was two
thousand years before Copernicus and Brahe, Kepler and Newton overcame the
preconception that all heavenly motions must be circular, and found the
ellipse, parabola, and hyperbola in the paths of planets, comets, and
projectiles.

In the 17th century Newton and Leibniz created calculus, with its
techniques for "differentiating" or finding the derivative of functions --
in geometric terms, finding the tangent of a curve at any given point.
True, some functions were discontinuous, with no tangent at a gap or an
isolated point. Some had singularities: abrupt changes in direction at
which the idea of a tangent becomes meaningless. But these were seen as
exceptional, and attention was focused on the "well-behaved" functions
that worked well in modeling nature.

Beginning in the early 1870s, though, a 50-year crisis transformed
mathematical thinking. Weierstrass described a function that was
continuous but nondifferentiable -- no tangent could be described at any
point. Cantor showed how a simple, repeated procedure could turn a line
into a dust of scattered points, and Peano generated a convoluted curve
that eventually touches every point on a plane. These shapes seemed to
fall "between" the usual categories of one-dimensional lines, two-
dimensional planes and three-dimensional volumes. Most still saw them as
"pathological" cases, but here and there they began to find applications.

In other areas of mathematics, too, strange shapes began to crop up.
Poincare attempted to analyze the stability of the solar system in the
1880s and found that the many-body dynamical problem resisted traditional
methods. Instead, he developed a qualitative approach, a "state space" in
which each point represented a different planetary orbit, and studied what
we would now call the topology -- the "connectedness" -- of whole families
of orbits. This approach revealed that while many initial motions quickly
settled into the familiar curves, there were also strange, "chaotic"
orbits that never became periodic and predictable.

Other investigators trying to understand fluctuating, "noisy" phenomena --
the flooding of the Nile, price series in economics, the jiggling of
molecules in Brownian motion in fluids -- found that traditional models
could not match the data. They had to introduce apparently arbitrary
scaling features, with spikes in the data becoming rarer as they grew
larger, but never disappearing entirely.

For many years these developments seemed unrelated, but there were
tantalizing hints of a common thread. Like the pure mathematicians' curves
and the chaotic orbital motions, the graphs of irregular time series often
had the property of self-similarity: a magnified small section looked very
similar to a large one over a wide range of scales.

1.4   who is this guy, anyway?

While many pure and applied mathematicians advanced these trends, it is
Benoit Mandelbrot above all who saw what they had in common and pulled the
threads together into the new discipline.

He was born in Warsaw in 1924, and moved to France in 1935. In a time when
French mathematical training was strongly analytic, he visualized problems
whenever possible, so that he could attack them in geometric terms. He
attended the Ecole Polytechnique, then Caltech, where he encountered the
tangled motions of fluid turbulence.

In 1958 he joined IBM, where he began a mathematical analysis of
electronic "noise" -- and began to perceive a structure in it, a hierarchy
of fluctuations of all sizes, that could not be explained by existing
statistical methods. Through the years that followed, one seemingly
unrelated problem after another was drawn into the growing body of ideas
he would come to call fractal geometry.

As computers gained more graphic capabilities, the skills of his mind's
eye were reinforced by visualization on display screens and plotters.
Again and again, fractal models produced results -- series of flood
heights, or cotton prices -- that experts said looked like "the real
thing."

Visualization was extended to the physical world as well. In a provocative
essay titled "How Long Is the Coast of Britain?" Mandelbrot noted that the
answer depends on the scale at which one measures: it grows longer and
longer as one takes into account every bay and inlet, every stone, every
grain of sand. And he codified the "self-similarity" characteristic of
many fractal shapes -- the reappearance of geometrically similar features
at all scales.

First in isolated papers and lectures, then in two editions of his seminal
book, he argued that many of science's traditional mathematical models are
ill-suited to natural forms and processes: in fact, that many of the
"pathological" shapes mathematicians had discovered generations before are
useful approximations of tree bark and lung tissue, clouds and galaxies.

Mandelbrot was named an IBM Fellow in 1974, and continues to work at the
IBM Watson Research Center. He has also been a visiting professor and
guest lecturer at many universities.

1.5   Periodicity Logic

The "Mandelbrot Lake" in the center of the M-set images is the traditional
bane of plotting programs. It sucks up the most computer time because it
always reaches the iteration limit -- and yet the most interesting areas
are invariably right at the edge the lake.
(See The Mandelbrot Set for a description of the iteration process.)

Thanks to Mark Peterson for pointing out (well, he more like beat us over
the head until we paid attention) that the iteration values in the middle
of Mandelbrot Lake tend to decay to periodic loops (i.e., Z(n+m) == Z(n),

a fact that is pointed out on pages 58-61 of "The Beauty of Fractals"). An
intelligent program (like the one he wrote) would check for this
periodicity once in a while, recognize that iterations caught in a loop
are going to max out, and bail out early.

For speed purposes, the current version of the program turns this checking
algorithm on only if the last pixel generated was in the lake. (The
checking itself takes a small amount of time, and the pixels on the very
edge of the lake tend to decay to periodic loops very slowly, so this
compromise turned out to be the fastest generic answer).

Try a full M-set plot with a 1000-iteration maximum with any other
program, and then try it on this one for a pretty dramatic proof of the
value of periodicity checking.

You can get a visual display of the periodicity effects if you press
<O>rbits while plotting. This toggles display of the intermediate
iterations during the generation process. It also gives you an idea of
how much work your poor little PC is going through for you! If you use
this toggle, it's best to disable solid-guessing first using <1> or <2>
because in its second pass, solid-guessing bypasses many of the pixel
calculations precisely where the orbits are most interesting.

Mark was also responsible for pointing out that 16-bit integer math was
good enough for the first few levels of M/J images, where the round-off
errors stay well within the area covered by a single pixel. Fractint now
uses 16-bit math where applicable, which makes a big difference on non-32-
bit PCs.

1.6   Limitations of Integer Math (And How We Cope)

By default, Fractint uses 16-bit and/or 32-bit integer math to generate
nearly all its fractal types. The advantage of integer math is speed: this
is by far the fastest such plotter that we have ever seen on any PC. The
disadvantage is an accuracy limit. Integer math represents numbers like
1.00 as 32-bit integers of the form [1.00 * (2^29)] (approximately
a range of 500,000,000) for the Mandelbrot and Julia sets. Other integer
fractal types use a bitshift of 24 rather than 29, so 1.0 is stored
internally as [1.00 * (2^24)]. This yields accuracy of better than 8
significant digits, and works fine... until the initial values of the
calculations on consecutive pixels differ only in the ninth decimal place.

At that point, if Fractint has a floating-point algorithm handy for that
particular fractal type (and virtually all of the fractal types have one
these days), it will silently switch over to the floating-point algorithm
and keep right on going. Fair warning - if you don't have an FPU, the
effect is that of a rocket sled hitting a wall of jello, and even if you
do, the slowdown is noticeable.

If it has no floating-point algorithm, Fractint does the best it can: it
switches to its minimal drawing mode, with adjacent pixels having initial
values differing by 1 (really 0.000000002). Attempts to zoom further may
result in moving the image around a bit, but won't actually zoom. If you
are stuck with an integer algorithm, you can reach minimal mode with your
fifth consecutive "maximum zoom", each of which covers about 0.25% of the

previous screen. By then your full-screen image is an area less than
1/(10^13)th [~0.0000000000001] the area of the initial screen. (If your
image is rotated or stretched very slightly, you can run into the wall of
jello as early as the fourth consecutive maximum zoom. Rotating or
stretching by larger amounts has less impact on how soon you run into it.)

Think of it this way: at minimal drawing mode, your VGA display would have
to have a surface area of over one million square miles just to be able to
display the entire M-set using the integer algorithms. Using the
floating-point algorithms, your display would have to be big enough to fit
the entire solar system out to the orbit of Saturn inside it. So there's
a considerable saving on hardware, electricity and desk space involved
here. Also, you don't have to take out asteroid insurance.

32 bit integers also limit the largest number which can be stored. This
doesn't matter much since numbers outside the supported range (which is
between -4 and +4) produce a boring single color. If you try to zoom-out
to reduce the entire Mandelbrot set to a speck, or to squeeze
it to a pancake, you'll find you can't do so in integer math mode.

1.7   Arbitrary Precision and Deep Zooming

The zoom limit of Fractint is approximately 10^15 (10 to the fifteenth
power). This limit is due to the precision possible with the computer
representation of numbers as 64 bit double precision data. To give you an
idea of just how big a magnification 10^15 is, consider this. At the scale
of your computer screen while displaying a tiny part of the Mandelbrot set at
the deepest possible zoom, the entire Mandelbrot set would be many millions
of miles wide, as big as the orbit of Jupiter.

Big as this zoom magnification is, your PC can do better using something
called arbitrary precision math. Instead of using 64 bit double precision to
represent numbers, your computer software allocates as much memory as needed
to create a data type supporting as many decimals of precision as you want.

Incorporation of this feature in Fractint was inspired by Jay Hill and his
DEEPZOOM program which uses the shareware MFLOAT programming library.
Several of the Stone Soup programmers noticed Jay's posts in the Internet
sci.fractals newsgroup and began to investigate adding arbitrary precision
to Fractint. High school math and physics teacher Wes Loewer wrote an
arbitrary precision library in both 80x86 assembler and C, and the Stone Soup
team incorporated Wes's library into Fractint. Initially, support was added
for fractal types mandel, julia, manzpower, and julzpower.

Normally, when you reach Fractint's zoom limit, Fractint simply
refuses to let you zoom any more. When using the fractal types that support
arbitrary precision, you will not reach this limit, but can keep on
zooming. When you pass the threshold between double precision and arbitrary
precision, Fractint will dramatically slow down. The <tab> status screen can
be used to verify that Fractint is indeed using arbitrary precision.

Fractals with arbitrary precision are SLOW, as much as ten times slower than
if the math were done with your math coprocessor, and even slower simply
because the zoom depth is greater. The good news, if you want to call it
that, is that your math coprocessor is not needed; coprocessorless machines

can produce deep zooms with the same glacial slowness as machines with
coprocessors!

Maybe the real point of arbitrary precision math is to prolong the "olden"
days when men were men, women were women, and real fractal programmers spent
weeks generating fractals. One of your Stone Soup authors has a large
monitor that blinks a bit when changing video modes--PCs have gotten so fast
that Fractint finishes the default 320x200 Mandelbrot before the monitor can
even complete its blinking transition to graphics mode! Computers are getting
faster every day, and soon a new generation of fractal lovers might forget
that fractal generation is *supposed* to be slow, just as it was in Grandpa's
day when they only had Pentium chips. The solution to this educational
dilemma is Fractint's arbitrary precision feature. Even the newest sexium and
septium machines are going to have to chug for days or weeks at the extreme
zoom depths now possible ...

So how far can you zoom? How does 10^{1600} sound--roughly 1600 decimal
digits of precision. To put *this* magnification in perspective, the "tiny"
ratio of 10^{61} is the ratio of the entire visible universe to the smallest
quantum effects. With 1600 digits to work with, you can expand an
electron-sized image up to the size of the visible universe, not once but
more than twenty times. So you can examine screen-sized portions of a
Mandelbrot set so large all but a tiny part of it would be vastly farther away
than the billion or so light year limit of our best telescopes.

Lest anyone suppose that we Stone Soupers suffer from an inflated pride over
having thus spanned the Universe, current inflationary cosmological theories
estimate the size of the universe to be unimaginably larger than the
"tiny" part we can see.

Note: many of Fractint's options do not work with arbitrary precision. To
experiment with arbitrary precision at the speedier ordinary magnifications,
start Fractint with the debug=3200 command-line option. With the exception
of mandel and manzpower perturbations, values that would normally be entered
in the Parameters and Coordinates screens need to be entered using the
command-line interface or .par files. Other known things that do not yet
work with arbitrary precision are: biomorph, decomp, distance estimator,
inversion, Julia-Mandel switch, history, orbit-in-window, and the browse
feature.

1.8   The Fractint 'Fractal Engine' Architecture

Several of the authors would never ADMIT this, but Fractint has evolved a
powerful and flexible architecture that makes adding new fractals very
easy. (They would never admit this because they pride themselves on being
the sort that mindlessly but happily hacks away at code and "sees if it
works and doesn't hang the machine".)

Many fractal calculations work by taking a rectangle in the complex plane,
and, point by point, calculating a color corresponding to that point.
Furthermore, the color calculation is often done by iterating a function
over and over until some bailout condition is met.
(See The Mandelbrot Set for a description of the iteration process.)

In implementing such a scheme, there are three fractal-specific

calculations that take place within a framework that is pretty much the
same for them all. Rather than copy the same code over and over, we
created a standard fractal engine that calls three functions that may be
bolted in temporarily to the engine. The "bolting in" process uses the C
language mechanism of variable function pointers.

These three functions are:

   1) a setup function that is run once per image, to do any required
   initialization of variables,

   2) a once-per-pixel function that does whatever initialization has to
   be done to calculate a color for one pixel, and

   3) a once-per-orbit-iteration function, which is the fundamental
   fractal algorithm that is repeatedly iterated in the fractal
   calculation.

The common framework that calls these functions can contain all sorts of
speedups, tricks, and options that the fractal implementor need not worry
about. All that is necessary is to write the three functions in the
correct way, and BINGO! – all options automatically apply. What makes it
even easier is that usually one can re-use functions 1) and 2) written for
other fractals, and therefore only need to write function 3).

Then it occurred to us that there might be more than one sort of fractal
engine, so we even allowed THAT to be bolted in. And we created a data
structure for each fractal that includes pointers to these four functions,
various prompts, a default region of the complex plane, and various
miscellaneous bits of information that allow toggling between Julia and
Mandelbrot or toggling between the various kinds of math used in
implementation.

That sounds pretty flexible, but there is one drawback – you have to be a
C programmer and have a C compiler to make use of it! So we took it a step
further, and designed a built-in high level compiler, so that you can
enter the formulas for the various functions in a formula file in a
straightforward algebra-like language, and Fractint will compile them and
bolt them in for you!

There is a terrible down side to this flexibility. Fractint users
everywhere are going berserk. Fractal-inventing creativity is running
rampant. Proposals for new fractal types are clogging the mail and the
telephones.

All we can say is that non-productivity software has never been so potent,
and we're sorry, it's our fault!

Fractint was compiled using Microsoft C 7.0 and Microsoft Assembler 6.0,
using the "Medium" model. Note that the assembler code uses the "C" model
option added to version 5.1, and must be assembled with the /MX or /ML
switch to link with the "C" code. Because it has become too large to
distribute comfortably as a single compressed file, and because many
downloaders have no intention of ever modifying it, Fractint is now
distributed as two files: one containing FRACTINT.EXE, auxiliary files and
this document, and another containing complete source code (including a

.MAK file and MAKEFRAC.BAT). See Distribution of Fractint.

1.9 Inside=bof60|bof61|zmag|period

INSIDE=BOF60|BOF61|ZMAG|PERIOD

Here is an *ATTEMPTED* explanation of what the inside=bof60 and
inside=bof61 options do. This explanation is hereby dedicated to Adrian
Mariano, who badgered it out of us! For the *REAL* explanation, see
"Beauty of Fractals", page 62.

Let p(z) be the function that is repeatedly iterated to generate a fractal
using the escape-time algorithm. For example, p(z) = z^2+c in the case of
a Julia set. Then let pk(z) be the result of iterating the function p for
k iterations. (The "k" should be shown as a superscript.) We could also
use the notation pkc(z) when the function p has a parameter c, as it does
in our example. Now hold your breath and get your thinking cap on. Define
a(c) = inf{|pkc(0)|:k=1,2,3,...}. In English – a(c) is the greatest lower
bound of the images of zero of as many iterations as you like. Put another
way, a(c) is the closest to the origin any point in the orbit starting
with 0 gets. Then the index (c) is the value of k (the iteration) when
that closest point was achieved. Since there may be more than one,
index(c) is the least such. Got it? Good, because the "Beauty of
Fractals" explanation of this, is, ahhhh, *TERSE* ! Now for the punch
line. Inside=bof60 colors the lake alternating shades according to the
level sets of a(c). Each band represents solid areas of the fractal where
the closest value of the orbit to the origin is the same. Inside=bof61
show domains where index(c) is constant. That is, areas where the
iteration when the orbit swooped closest to the origin has the same value.
Well, folks, that's the best we can do! Improved explanations will be
accepted for the next edition!

In response to this request for lucidity, Herb Savage offers this explanation
the bof60 and bof61 options:

 The picture on page 60 of The Beauty of Fractals shows the distance to
 origin of the closest point to the origin in the sequence of points
 generated from a given X,Y coordinate. The picture on page 61 shows
 the index (or number) in the sequence of the closest point.

inside=zmag is similar. This option colors inside pixels according to
the magnitude of the orbit point when maxiter was reached, using the formula
color = (x^2 + y^2) * maxiter/2 + 1.

inside=period colors pixels according to the length of their eventual cycle.
For example, points that approach a fixed point have color=1. Points that
approach a 2-cycle have color=2. Points that do not approach a cycle during
the iterations performed have color=maxit. This option works best with a
fairly large number of iterations.

1.10 Inside=epscross|startrail

INSIDE=EPSCROSS|STARTRAIL

Kenneth Hooper has written a paper entitled "A Note On Some Internal
Structures Of The Mandelbrot Set" published in "Computers and Graphics", Vol
15, No.2, pp. 295-297. In that article he describes Clifford Pickover's
"epsilon cross" method which creates some mysterious plant-like tendrils in
the Mandelbrot set. The algorithm is this. In the escape-time calculation of a
fractal, if the orbit comes within .01 of the Y-axis, the orbit is terminated
and the pixel is colored green. Similarly, the pixel is colored yellow if it
approaches the X-axis. Strictly speaking, this is not an "inside" option
because a point destined to escape could be caught by this bailout criterion.

Hooper has another coloring scheme called "star trails" that involves
detecting clusters of points being traversed by the orbit. A table of tangents
of each orbit point is built, and the pixel colored according to how many
orbit points are near the first one before the orbit flies out of the cluster.
This option looks fine with maxiter=16, which greatly speeds the calculation.

Both of these options should be tried with the outside color fixed
(outside=<nnn>) so that the "lake" structure revealed by the algorithms can be
more clearly seen. Epsilon Cross is fun to watch with boundary tracing turned
on - even though the result is incorrect it is interesting! Shucks - what
does "incorrect" mean in chaos theory anyway?!

1.11   Finite Attractors

FINITE ATTRACTORS

Many of Fractint's fractals involve the iteration of functions of complex
numbers until some "bailout" value is exceeded, then coloring the
associated pixel according to the number of iterations performed. This
process identifies which values tend to infinity when iterated, and gives
us a rough measure of how "quickly" they get there.

In dynamical terms, we say that "Infinity is an Attractor", as many
initial values get "attracted" to it when iterated. The set of all points
that are attracted to infinity is termed The Basin of Attraction of
Infinity. The coloring algorithm used divides this Basin of Attraction
into many distinct sets, each a single band of one color, representing all
the points that are "attracted" to Infinity at the same "rate". These
sets (bands of color) are termed "Level Sets" - all points in such a set
are at the same "Level" away from the attractor, in terms of numbers of
iterations required to exceed the bailout value.

Thus, Fractint produces colored images of the Level Sets of the Basin of
Attraction of Infinity, for all fractals that iterate functions of Complex
numbers, at least. Now we have a sound mathematical definition of what
Fractint's "bailout" processing generates, and we have formally introduced
the terms Attractor, Basin of Attraction, and Level Set, so you should
have little trouble following the rest of this section!

For certain Julia-type fractals, Fractint can also display the Level Sets
of Basins of Attraction of Finite Attractors. This capability is a by-
product of the implementation of the MAGNETic fractal types, which always

have at least one Finite Attractor.

This option can be invoked by setting the "Look for finite attractor"
option on the <Y> options screen, or by giving the "finattract=yes"
command-line option.

Most Julia-types that have a "lake" (normally colored blue by default)
have a Finite Attractor within this lake, and the lake turns out to be,
quite appropriately, the Basin of Attraction of this Attractor.

The "finattract=yes" option (command-line or <Y> options screen)
instructs Fractint to seek out and identify a possible Finite Attractor
and, if found, to display the Level Sets of its Basin of Attraction, in
addition to those of the Basin of Attraction of Infinity. In many cases
this results in a "lake" with colored "waves" in it; in other cases there
may be little change in the lake's appearance.

For a quick demonstration, select a fractal type of LAMBDA, with a
parameter of 0.5 + 0.5i. You will obtain an image with a large blue lake.
Now set "Look for finite attractor" to 1 with the "Y" menu.
The image will be re-drawn
with a much more colorful lake. A Finite Attractor lives in the center of
one of the resulting "ripple" patterns in the lake - turn the <O>rbits
display on to see where it is - the orbits of all initial points that are
in the lake converge there.

Fractint tests for the presence of a Finite Attractor by iterating a
Critical Value of the fractal's function. If the iteration doesn't bail
out before exceeding twice the iteration limit, it is almost certain that
we have a Finite Attractor - we assume that we have.

Next we define a small circle around it and, after each iteration, as well
as testing for the usual bailout value being exceeded, we test to see if
we've hit the circle. If so, we bail out and color our pixels according to
the number of iterations performed. Result - a nicely colored-in lake
that displays the Level Sets of the Basin of Attraction of the Finite
Attractor. Sometimes !

First exception: This does not work for the lakes of Mandel-types. Every
point in a Mandel-type is, in effect, a single point plucked from one of
its related Julia-types. A Mandel-type's lake has an infinite number of
points, and thus an infinite number of related Julia-type sets, and
consequently an infinite number of finite attractors too. It *MAY* be
possible to color in such a lake, by determining the attractor for EVERY
pixel, but this would probably treble (at least) the number of iterations
needed to draw the image. Due to this overhead, Finite Attractor logic
has not been implemented for Mandel-types.

Secondly, certain Julia-types with lakes may not respond to this
treatment, depending on the parameter value used. E.g., the Lambda Set
for 0.5 + 0.5i responds well; the Lambda Set for 0.0 + 1.0i does not - its
lake stays blue. Attractors that consist of single points, or a cycle of
a finite number of points are ok. Others are not. If you're into fractal
technospeak, the implemented approach fails if the Julia-type is a
Parabolic case, or has Siegel Disks, or has Herman Rings.

However, all the difficult cases have one thing in common - they all have

a parameter value that falls exactly on the edge of the related Mandel-
type's lake. You can avoid them by intelligent use of the Mandel-Julia
Space-Bar toggle: Pick a view of the related Mandel-type where the center
of the screen is inside the lake, but not too close to its edge, then use
the space-bar toggle. You should obtain a usable Julia-type with a lake,
if you follow this guideline.

Thirdly, the initial implementation only works for Julia-types that use
the "Standard" fractal engine in Fractint. Fractals with their own
special algorithms are not affected by Finite Attractor logic, as yet.

Finally, the finite attractor code will not work if it fails to detect
a finite attractor. If the number of iterations is set too low, the finite
attractor may be missed.

Despite these restrictions, the Finite Attractor logic can produce
interesting results. Just bear in mind that it is principally a bonus
off-shoot from the development of the MAGNETic fractal types, and is not
specifically tuned for optimal performance for other Julia types.

(Thanks to Kevin Allen for the above).

There is a second type of finite attractor coloring, which is selected
by setting "Look for Finite Attractor" to a negative value. This colors
points by the phase of the convergence to the finite attractor,
instead of by the speed of convergence.

For example, consider the Julia set for -0.1 + 0.7i, which is the three-lobed
"rabbit" set. The Finite Attractor is an orbit of length three; call these
values a, b, and c. Then, the Julia set iteration can converge to one of
three sequences: a,b,c,a,b,c,..., or b,c,a,b,c,..., or c,a,b,c,a,b,...
The Finite Attractor phase option colors the interior of the Julia set with
three colors, depending on which of the three sequences the orbit converges
to. Internally, the code determines one point of the orbit, say "a", and
the length of the orbit cycle, say 3. It then iterates until the sequence
converges to a, and then uses the iteration number modulo 3 to determine the
color.

1.12 Trig Identities

TRIG IDENTITIES

The following trig identities are invaluable for coding fractals that use
complex-valued transcendental functions of a complex variable in terms of
real-valued functions of a real variable, which are usually found in
compiler math libraries. In what follows, we sometimes use "*" for
multiplication, but leave it out when clarity is not lost. We use "^" for
exponentiation; x^y is x to the y power.

```
   (u+iv)  + (x+iy) = (u+x) + i(v+y)
   (u+iv)  - (x+iy) = (u-x) + i(v-y)
   (u+iv)  * (x+iy) = (ux - vy) + i(vx + uy)
   (u+iv)  / (x+iy) = ((ux + vy) + i(vx - uy)) / (x^2 + y^2)

   e^(x+iy)   = (e^x) (cos(y) + i sin(y))
```

```
   log(x+iy) = (1/2)log(x*x + y*y) + i(atan(y/x) + 2kPi)
      for k = 0, -1, 1, -2, 2, ...
       (Fractint generally uses only the principle value, k=0. The log
        function refers to log base e, or ln.)

   z^w = e^(w*log(z))

   sin(x+iy)  = sin(x)cosh(y) + i cos(x)sinh(y)
   cos(x+iy)  = cos(x)cosh(y) - i sin(x)sinh(y)
   tan(x+iy)  = sin(x+iy) / cos(x+iy)
   sinh(x+iy) = sinh(x)cos(y) + i cosh(x)sin(y)
   cosh(x+iy) = cosh(x)cos(y) + i sinh(x)sin(y)
   tanh(x+iy) = sinh(x+iy) / cosh(x+iy)
   cosxx(x+iy) = cos(x)cosh(y) + i sin(x)sinh(y)
      (cosxx is present in Fractint to provide compatibility with a bug
      which was in its cos calculation before version 16)

                    sin(2x)              sinh(2y)
   tan(x+iy) = ------------------  + i------------------
                cos(2x) + cosh(2y)    cos(2x) + cosh(2y)

                 sin(2x) - i*sinh(2y)
   cotan(x+iy) = --------------------
                  cosh(2y) - cos(2x)

                    sinh(2x)             sin(2y)
   tanh(x+iy) = ----------------- + i------------------
                cosh(2x) + cos(2y)   cosh(2x) + cos(2y)

                  sinh(2x) - i*sin(2y)
   cotanh(x+iy) = --------------------
                   cosh(2x) - cos(2y)

   asin(z) = -i * log(i*z+sqrt(1-z*z))
   acos(z) = -i * log(z+sqrt(z*z-1))
   atan(z) = i/2* log((1-i*z)/(1+i*z))

   asinh(z) = log(z+sqrt(z*z+1))
   acosh(z) = log(z+sqrt(z*z-1))
   atanh(z) = 1/2*log((1+z)/(1-z))

   sqr(x+iy) = (x^2-y^2) + i*2xy
   sqrt(x+iy) = sqrt(sqrt(x^2+y^2)) * (cos(atan(y/x)/2) + i sin(atan(y/x)/2))

   ident(x+iy) = x+iy
   conj(x+iy) = x-iy
   recip(x+iy) = (x-iy)/(x^2+y^2)
   flip(x+iy) = y+ix
   zero(x+iy) = 0
   cabs(x+iy) = sqrt(x^2 + y^2)
```

Fractint's definitions of abs(x+iy) and |x+iy| below are non-standard.
Math texts define both absolute value and modulus of a complex number to
be the same thing. They are both equal to cabs(x+iy) as defined above.

```
   |x+iy| = x^2 + y^2
```

```
abs(x+iy) = sqrt(x^2) + i sqrt(y^2)
```

1.13 Quaternion and Hypercomplex Algebra

Quaternions are four dimensional generalizations of complex numbers.
They almost obey the familiar field properties of real numbers, but
fail the commutative law of multiplication, since x*y is not generally
equal to y*x.

Quaternion algebra is most compactly described by specifying the rules
for multiplying the basis vectors 1, i, j, and k. Quaternions form a
superset of the complex numbers, and the basis vectors 1 and i are the
familiar basis vectors for the complex algebra. Any quaternion q can be
represented as a linear combination q = x + yi + zj + wk of the basis
vectors just as any complex number can be written in the form z = a + bi.

Multiplication rules for quaternion basis vectors:
ij = k jk = i ki = j
ji = -k kj = -i ik = -j
ii = jj = kk = -1
ijk = -1

Note that ij = k but ji = -k, showing the failure of the commutative law.
The rules for multiplying any two quaternions follow from the behavior
of the basis vectors just described. However, for your convenience, the
following formula works out the details.

Let q1 = x1 + y1i + z1j + w1k and q2 = x2 + y2i + z2j + w2k.
Then q1q2 = 1(x1x2 - y1y2 - z1z2 - w1w2) +
            i(y1x2 + x1y2 + w1z2 - z1w2) +
            j(z1x2 - w1y2 + x1z2 + y1w2) +
            k(w1x2 + z1y2 - y1z2 + x1w2)

Quaternions are not the only possible four dimensional supersets of the
complex numbers. William Hamilton, the discoverer of quaternions in the
1830's, considered the alternative called the hypercomplex number system.
Unlike quaternions, the hypercomplex numbers satisfy the commutative law of
multiplication. The law which fails is the field property that states that
all non-zero elements of a field have a multiplicative inverse. For a non-zero
hypercomplex number h, the multiplicative inverse 1/h does not always exist.

As with quaternions, we will define multiplication in terms of the basis
vectors 1, i, j, and k, but with subtly different rules.

Multiplication rules for hypercomplex basis vectors:
ij = k jk = -i ki = -j
ji = k kj = -i ik = -j
ii = jj = -kk = -1
ijk = 1

Note that now ij = k and ji = k, and similarly for other products of pairs
of basis vectors, so the commutative law holds.

Hypercomplex multiplication formula:
Let h1 = x1 + y1i + z1j + w1k and h2 = x2 + y2i + z2j + w2k.

```
Then  h1h2 =  1(x1x2 - y1y2 - z1z2 + w1w2) +
              i(y1x2 + x1y2 - w1z2 - z1w2) +
              j(z1x2 - w1y2 + x1z2 - y1w2) +
              k(w1x2 + z1y2 + y1z2 + x1w2)
```

As an added bonus, we'll give you the formula for the reciprocal.

```
Let det = [((x-w)^2+(y+z)^2)((x+w)^2+(y-z)^2)]
Then 1/h =   1[ x(x^2+y^2+z^2+w^2)-2w(xw-yz)]/det +
             i[-y(x^2+y^2+z^2+w^2)-2z(xw-yz)]/det +
             j[-z(x^2+y^2+z^2+w^2)-2y(xw-yz)]/det +
             k[ w(x^2+y^2+z^2+w^2)-2x(xw-yz)]/det
```

A look at this formula shows the difficulty with hypercomplex numbers.
In order to calculate 1/h, you have to divide by the quantity
det = [((x-w)^2+(y+z)^2)((x+w)^2+(y-z)^2)]. So when this quantity is zero,
the multiplicative inverse will not exist.

Hypercomplex numbers numbers have an elegant generalization of any unary
complex valued function defined on the complex numbers. First, note that
hypercomplex numbers can be represented as a pair of complex numbers in the
following way.
```
Let h = x + yi + zj + wk.
    a = (x-w) + i(y+z)
    b = (x+w) + i(y-z)
```
The numbers a and b are complex numbers. We can represent h as the pair of
complex numbers (a,b). Conversely, if we have a hypercomplex number given
to us in the form (a,b), we can solve for x, y, z, and w. The solution to
```
   c = (x-w) + i(y+z)
   d = (x+w) + i(y-z)
is
   x = (real(c) + real(d))/2
   y = (imag(c) + imag(d))/2
   z = (imag(c) - imag(d))/2
   x = (real(d) - real(c))/2
```
We can now, for example, define sin(h) as (sin(a),sin(b)). We know how to
compute sin(a) and sin(b) (see trig identities above).

Let c = sin(a) and d = sin(b). Now use the equations above to solve for
x, y, z, and w in terms of c and d. The beauty of this is that it really
doesn't make any difference what function we use. Instead of sin, we could
have used cos, sinh, ln, or z^2. Using this technique, Fractint can create
3-D fractals using the formula h' = fn(h) + c, where "fn" is any of the
built-in functions. Where fn is sqr(), this is the famous mandelbrot formula,
generalized to four dimensions.

For more information, see _Fractal Creations, Second Edition_ by Tim Wegner
and Bert Tyler, Waite Group Press, 1993.

1.14 GIF Save File Format

Since version 5.0, Fractint has had the <S>ave-to-disk command, which
stores screen images in the extremely compact, flexible .GIF (Graphics
Interchange Format) widely supported on CompuServe. Version 7.0 added the
<R>estore-from-disk capability.

Until version 14, Fractint saved images as .FRA files, which were a
non-standard extension of the then-current GIF87a specification. The
reason was that GIF87a did not offer a place to store the extra
information needed by Fractint to implement the
<R> feature -- i.e., the parameters that let you keep zooming, etc.
as if the restored file had just been created in this session.
The .FRA format worked with all of the popular GIF decoders that we
tested, but these were not true GIF files. For one thing,
information after the GIF terminator (which is where we put the extra info)
has the potential to confuse the
online GIF viewers used on CompuServe. For another, it is the opinion of
some GIF developers that the addition of this extra information violates
the GIF87a spec. That's why we used the default filetype .FRA instead.

Since version 14, Fractint has used a genuine .GIF format, using the
GIF89a spec - an upwardly compatible extension of GIF87a, released by
CompuServe on August 1 1990.
This new spec allows the
placement of application data within "extension blocks".
In version 14 we changed our default savename extension from .FRA
to .GIF.

There is one significant advantage to the new GIF89a format compared to
the old GIF87a-based .FRA format for Fractint purposes: the new .GIF
files may be uploaded to the CompuServe graphics forums fractal information
intact. Therefore anyone downloading a Fractint image from CompuServe will
also be downloading all the information needed to regenerate the image.

Fractint can still read .FRA files generated by
earlier versions. If for some reason you wish to save files in the older
GIF87a format, for example because your favorite GIF decoder has not yet
been upgraded to GIF89a, use the command-line parameter "GIF87a=yes".
Then any saved files will use the original GIF87a format without any
application-specific information.

An easy way to convert an older .FRA file into true .GIF format suitable
for uploading is something like this at the DOS prompt:
    FRACTINT MYFILE.FRA SAVENAME=MYFILE.GIF BATCH=YES
Fractint will load MYFILE.FRA, save it in true .GIF format as MYFILE.GIF,
and return to DOS.

GIF and "Graphics Interchange Format" are trademarks of CompuServe
Incorporated, an H&R Block Company.

1.15 Using Fractint With a Mouse

Left Button: Brings up and sizes the Zoom Box.   While holding down the
               left button, push the mouse forward to shrink the Zoom Box,
               and pull it back to expand it.
               Double-clicking the left button performs the Zoom.

Right Button: While holding the right button held down, move the mouse
               from side to side to 'rotate' the Zoom Box. Move the mouse
               forward or back to change the Zoom Box color.

            Double-clicking the right button performs a 'Zoom-Out'.

Both Buttons: (or the middle button, if you have three of them) While
               holding down both buttons, move the mouse up and down to
               stretch/shrink the height of the Zoom Box, or side to side
               to 'squish' the Zoom Box into a non-rectangular shape.

Zoom and Pan using the mouse typically consists of pushing in the left
button, sizing the zoom box, letting go of the button, panning to the
general area, then double-clicking the left button to perform the Zoom.

1.16   selecting a video mode when loading a file

The most suitable video modes for the file are listed first.

The 'err' column in the video mode information indicates:
  blank mode seems perfect for this image
  v   image smaller than screen, will be loaded in a <v>iew window
  c   mode has more colors than image needs
  *   a major problem, one or more of the following is also shown:
  C   mode has too few colors
  R   image larger than screen, Fractint will reduce the image, possibly
         into a <v>iew window, and maybe with aspect ratio a bit wrong
  A   mode has the wrong shape of pixels for this image

1.17   Distribution of Fractint

DISTRIBUTION OF FRACTINT

New versions of FRACTINT are uploaded to the CompuServe network, and make
their way to other systems from that point. FRACTINT is available as
two self-extracting archive files - FRAINT.EXE (executable & documentation)
and FRASRC.EXE (source code).

The latest version can always be found in one of CompuServe's GO GRAPHICS
forums. Alas, the GO GRAPHICS Group is growing so fast that we get moved
around from periodically, and rumor has it that yet another move is imminent.
The current location of Fractint is the "Fractal Sources" library of the GO
GRAPHDEV forum. The forum staff will leave pointers to our new home if we are
moved again.

If you're not a CompuServe subscriber, but
wish to get more information about CompuServe and its graphics forums,
feel free to call their 800 number (800-848-8199) and ask for operator
number 229.

If you don't have access to CompuServe, many other sites tend to carry
these files shortly after their initial release (although sometimes
using different naming conventions). For instance...

If you speak Internet and FTP, SIMTEL20 and its various mirror sites
tend to carry new versions of Fractint shortly after they are released.

look in the /SimTel/msdos/graphics directory for files named FRA*.*. Then
again, if you don't speak Internet and FTP...

Your favorite local BBS probably carries these files as well (although
perhaps not the latest versions) using naming conventions like FRA*.ZIP.
One BBS that *does* carry the latest version is the "Ideal Studies BBS"
(508)757-1806, 1200/2400/9600HST. Peter Longo is the SYSOP and a true
fractal fanatic. There is a very short registration, and thereafter the
entire board is open to callers on the first call. Then again, if you
don't even have a modem...

Many Shareware/Freeware library services will ship you diskettes containing
the latest versions of Fractint for a nominal fee that basically covers
their cost of packaging and a small profit that we don't mind them making.
One in particular is the Public (Software) Library, PO Box 35705,
Houston, TX 77235-5705, USA. Their phone number is 800-242-4775 (outside
the US, dial 713-524-6394). Ask for item #9112 for five 5.25" disks, #9113
for three 3.5" disks. Cost is $6.99 plus $4 S&H in the U.S./Canada, $11
S&H overseas.

In Europe, the latest versions are available from another Fractint enthusiast,
Jon Horner - Editor of FRAC'Cetera, a disk-based fractal/chaos resource.
Disk prices for UK/Europe are: 5.25" HD BP4.00/4.50 : 3.5" HD BP (British
Pounds) 4.00/4.50. Prices include p&p (airmail to Europe). Contact:
Jon Horner, FRAC'Cetera, Le Mont Ardaine, Rue des Ardaines, St. Peters,
Guernsey GY7 9EU, CI, UK. Phone (44) 01481 63689. CIS 100112,1700

The X Windows port of Fractint maintained by Ken Shirriff is available
via FTP from sprite.berkeley.edu.

1.18 Contacting the Authors

CONTACTING THE AUTHORS
Communication between the authors for development of the next version of
Fractint takes place in a CompuServe (CIS) GO GRAPHICS GROUP (GGG) forum.
This forum changes from time to time as as the GGG grows. You can always
find it using the CompuServe GO GRAPHICS command. Currently we are located
in GRAPHDEV (Graphics Developers) forum, Section 4 (Fractal Sources).

Most of the authors have never met except on CompuServe. Access to the
GRAPHDEV forum is open to any and all interested in computer generated
fractals. New members are always welcome! Stop on by if you have any
questions or just want to take a peek at what's getting tossed into the soup.
This is by far the best way to have your questions answered or participate
in discussion. Also, you'll find many GIF image files generated by fellow
Fractint fans and many fractal programs as well in the GRAPHDEV forum's data
library 5.

If you're not a CompuServe subscriber, but wish to get more information
about CompuServe and its graphics forums, feel free to call their 800
number (800-848-8199) and ask for operator number 229.

The following authors have agreed to the distribution of their addresses.
Usenet/Internet/Bitnet/Whatevernet users can reach CIS users directly if
they know the user ID (i.e., Bert Tyler's ID is 73477.433@compuserve.com).

Just remember that CIS charges by the minute, so it costs us a little bit
to read a message -- don't kill us with kindness. And don't send all your
mail to Bert -- spread it around a little! Postal addresses are listed below
so that you have a way to send bug reports
and ideas to the Stone Soup team.

Please understand that we receive a lot of
mail, and because of the demands of volunteer work on Fractint as well as
our professional responsibilities, we are generally unable to answer it all.
Several of us have reached the point where we can't answer any conventional
mail. We *do* read and enjoy all the mail we receive, however. If you need a
reply, the best thing to do is use email, which we are generally able to
answer, or better yet, leave a message in CompuServe's GRAPHDEV.

(This address list is getting
seriously out of date. We have updated information from those folks who have
contacted us. The next release of Fractint will contain the addresses of
only those people who have explicitly told us that their address is
correct and they want it listed. Please contact one of the main authors with
this information.)

Current main authors:

Bert Tyler   [73477,433] on CIS
Tyler Software   (which is also 73477.433@compuserve.com, if
124 Wooded Lane   you're on the Internet - see above)
Villanova, PA 19085
(610) 525-5478

Timothy Wegner   [71320,675] on CIS
4714 Rockwood   twegner@phoenix.net (Internet)
Houston, TX 77004
(713) 747-7543

Jonathan Osuch   [73277,1432] on CIS
2110 Northview Drive
Marion, IA 52302

Wesley Loewer   loewer@tenet.edu on INTERNET
78 S. Circlewood Glen
The Woodlands, TX 77381
(713) 292-3449

Contributing authors' addresses (in alphabetic order).

Joseph A Albrecht
9250 Old Cedar Ave #215
Bloomington, Mn 55425
(612) 884-3286

Kevin C Allen   kevina@microsoft.com on Internet
9 Bowen Place
Seven Hills
NSW 2147
Australia
+61-2-870-2297 (Work)

```
(02) 831-4821 (Home)

Rob Beyer              [71021,2074] on CIS
23 Briarwood Lane
Laguna Hills, CA, 92656
(714) 957-0227
(7-12pm PST & weekends)

John W. Bridges        (Author GRASP/Pictor, Imagetools, PICEM, VGAKIT)
2810 Serang Place Costa Mesa
California 92626-4827   [75300,2137] on CIS, GENIE:JBRIDGES

Juan J Buhler          jbuhler@usina.org.ar
Santa Fe 2227 1P "E"
(54-1) 84 3528
Buenos Aires, Argentina

Michael D. Burkey      burkey@sun9.math.utk.edu on Internet
6600 Crossgate Rd.
Knoxville, TN 37912

Robin Bussell
13 Bayswater Rd
Horfield
Bristol
Avon, England
(044)-0272-514451

Prof Jm Collard-Richard jmc@math.ethz.ch

Monte Davis            [71450,3542] on CIS
223 Vose Avenue
South Orange, NJ 07079
(201) 378-3327

Paul de Leeuw
50 Henry Street
Five Dock
New South Wales
2046
Australia
+61-2-396-2246 (Work)
+61-2-713-6064 (Home)

David Guenther         [70531,3525] on CIS
50 Rockview Drive
Irvine, CA 92715

Michael L. Kaufman     kaufman@eecs.nwu.edu on INTERNET
2247 Ridge Ave, #2K    (also accessible via EXEC-PC bbs)
Evanston, IL, 60201
(708) 864-7916
Joe McLain             [75066,1257] on CIS
McLain Imaging
2417 Venier
Costa Mesa, CA 92627
(714) 642-5219
```

```
Bob Montgomery            [73357,3140] on CIS
(Author of VPIC)
132 Parsons Road
Longwood, Fl  32779


Roy Murphy                [76376,721] on CIS
9050 Ewing Ave.
Evanston, IL 60203


Ethan Nagel               [71062,3677] on CIS
4209 San Pedro NE #308
Albuquerque, NM 87109
(505) 884-7442


Mark Peterson             [73642,1775] on CIS
The Yankee Programmer
405-C Queen St., Suite #181
Southington, CT 06489
(203) 276-9721


Marc Reinig               [72410,77] on CIS
3415 Merrill Rd.          72410.77@compuserve.com.
Aptos, CA. 95003
(408) 475-2132


Lee H. Skinner            [75450,3631] on CIS
P.O. Box 14944
Albuquerque, NM  87191
(505) 293-5723


Dean Souleles             [75115,1671] on CIS
8840 Collett Ave.
Sepulveda, CA  91343
(818) 893-7558


Chris J Lusby Taylor
32 Turnpike Road
Newbury, England
Tel 011 44 635 33270


Scott Taylor                 [72401,410] on CIS
2913 Somerville Drive Apt #1  scott@bohemia.metronet.org on Internet
Ft. Collins, Co  80526       DGWM18A on Prodigy
(303) 221-1206


Paul Varner               [73237,441] on CIS
PO Box 930
Shepherdstown, WV 25443
(304) 876-2011


Phil Wilson               [76247,3145] on CIS
410 State St., #55
Brooklyn, NY 11217
(718) 624-5272
```

1.19 The Stone Soup Story

THE STONE SOUP STORY

Once upon a time, somewhere in Eastern Europe, there was a great famine.
People jealously hoarded whatever food they could find, hiding it even
from their friends and neighbors. One day a peddler drove his wagon into a
village, sold a few of his wares, and began asking questions as if he
planned to stay for the night.

[No! No! It was three Russian Soldiers! – Lee Crocker]
[Wait! I heard it was a Wandering Confessor! – Doug Quinn]
[Well *my* kids have a book that uses Russian Soldiers! – Bert]
[Look, who's writing this documentation, anyway? – Monte]
[Ah, but who gets it *last* and gets to upload it? – Bert]

"There's not a bite to eat in the whole province," he was told. "Better
keep moving on."

"Oh, I have everything I need," he said. "In fact, I was thinking of
making some stone soup to share with all of you." He pulled an iron
cauldron from his wagon, filled it with water, and built a fire under it.
Then, with great ceremony, he drew an ordinary-looking stone from a velvet
bag and dropped it into the water.

By now, hearing the rumor of food, most of the villagers had come to the
square or watched from their windows. As the peddler sniffed the "broth"
and licked his lips in anticipation, hunger began to overcome their
skepticism.

"Ahh," the peddler said to himself rather loudly, "I do like a tasty stone
soup. Of course, stone soup with CABBAGE -- that's hard to beat."

Soon a villager approached hesitantly, holding a cabbage he'd retrieved
from its hiding place, and added it to the pot. "Capital!" cried the
peddler. "You know, I once had stone soup with cabbage and a bit of salt
beef as well, and it was fit for a king."

The village butcher managed to find some salt beef...and so it went,
through potatoes, onions, carrots, mushrooms, and so on, until there was
indeed a delicious meal for all. The villagers offered the peddler a great
deal of money for the magic stone, but he refused to sell and traveled on
the next day. And from that time on, long after the famine had ended, they
reminisced about the finest soup they'd ever had.

                                    ***

That's the way Fractint has grown, with quite a bit of magic, although
without the element of deception. (You don't have to deceive programmers
to make them think that hours of painstaking, often frustrating work is
fun... they do it to themselves.)

It wouldn't have happened, of course, without Benoit Mandelbrot and the
explosion of interest in fractal graphics that has grown from his work at
IBM. Or without the example of other Mandelplotters for the PC. Or without
those wizards who first realized you could perform Mandelbrot calculations

using integer math (it wasn't us – we just recognize good algorithms when
we steal--uhh--see them). Or those graphics experts who hang around the
CompuServe PICS forum and keep adding video modes to the program. Or...
(continued in A Word About the Authors)

1.20   A Word About the Authors

A WORD ABOUT THE AUTHORS

Fractint is the result of a synergy between the main authors, many
contributors, and published sources. All four of the main authors have
had a hand in many aspects of the code. However, each author has certain
areas of greater contribution and creativity. Since there is not room in
the credits screen for the contributions of the main authors, we list these
here to facilitate those who would like to communicate with us on
particular subjects.

Main Authors of Version 19.

BERT TYLER is the original author of Fractint. He wrote the "blindingly
fast" 386-specific 32 bit integer math code and the original video mode
logic. Bert made Stone Soup possible, and provides a sense of direction when
we need it. His forte is writing fast 80x86 assembler, his knowledge of a
variety of video hardware, and his skill at hacking up the code we send him!

Bert has a BA in mathematics from Cornell University. He has been in
programming since he got a job at the computer center in his sophomore
year at college – in other words, he hasn't done an honest day's work in
his life. He has been known to pass himself off as a PC expert, a UNIX
expert, a statistician, and even a financial modeling expert. He is
currently masquerading as an independent PC consultant, supporting the PC-
to-Mainframe communications environment at NIH. If you sent mail from the
Internet to an NIH staffer on his 3+Mail system, it was probably Bert's
code that mangled it during the Internet-to-3+Mail conversion. He also
claims to support the MS-Kermit environment at NIH. Fractint is Bert's
first effort at building a graphics program.

TIM WEGNER contributed the original implementation of palette animation,
and is responsible for most of the 3D mechanisms. He provided
the main outlines of the "StandardFractal" engine and data structures, and is
accused by his cohorts of being "obsessed with options". One of Tim's main
interests is the use of four dimensional algebras to produce fractals.
Tim served as team coordinator for version 19, and integrated Wes Loewer's
arbitrary precision library into Fractint.

Tim has BA and MA degrees in mathematics from Carleton College and the
University of California Berkeley. He worked for 7 years overseas as a
volunteer, doing things like working with Egyptian villagers building
water systems. Since returning to the US in 1982, he has written shuttle
navigation software, a software support environment prototype, and
supported strategic information planning, all at NASA's Johnson Space
Center. Tim has started his own business, and now writes and programs full
time.

JONATHAN OSUCH started throwing pebbles into the soup around version 15.0

with a method for simulating an if-then-else structure using the formula
parser. He has contributed the fn||fn fractal types, the built-in bailout
tests, the increase in both the maximum iteration count and bailout value,
and bug fixes too numerous to count. Jonathan worked closely with Robin
Bussell to implement Robin's browser mechanism in Fractint.

Jonathan has a B.S. in Physics from the University of Dubuque and a B.S. in
Computer Science from Mount Mercy College, both in Iowa. He is currently
working as a consultant in the nuclear power industry.

WES LOEWER first got his foot in the Stone Soup door by writing fast
floating point assembler routines for Mandelbrot, Julia, and Lyapunov
fractals. He also rewrote the boundary trace algorithms and added the
frothybasin fractal. His most significant contribution is the addition of
the arbitrary precision library which allows Fractint to perform
incredibly deep zooms.

Wes has a B.S. in Physics from Wheaton College in Illinois. He also holds
an M.S. in Physics and an M.Ed. in Education from Texas A&M University.
Wes teaches physics and math at McCullough High School in The Woodlands,
Texas where his pupils inspire him to keep that sense of amazement that
students get when they understand a physical or mathematical principle for
the first time. Since he uses Fractint to help teach certain mathematical
principles, he's one of the few folks who actually gets to use Fractint on
the job. Besides his involvement with Fractint, Wes is the author of
WL-Plot, an equation graphing program, and MatCalc, a matrix calculator
program.

1.21 Other Fractal Products

(Forgive us, but we just *have* to begin this section with a plug for
our fractal products...)

Several of Fractint's programmers have written books about fractals,
Fractint, and Winfract (the Windows version of Fractint).
The book about Fractint is Fractal Creations Second Edition (1994 Waite
Group Press, ISBN # 1-878739-34-4). The book about Winfract is The Waite
Group's Fractals for Windows (1992 Waite Group Press, ISBN # 1-878739-25-5).

Fractal Creations Second Edition includes:
 o A guided tour of Fractint.
 o A detailed manual and reference section of commands.
 o A tutorial on fractals.
 o A reference containing tips, explanations, and examples of parameters
   for all the Fractals generated by Fractint/Winfract.
 o Secrets on how the programs work internally.
 o Spectacular color plate section.
 o A CD containing Fractint and Xfract source and executable, and over a
   thousand spectacular fractal images.
 o A complete copy of the source code with a chapter explaining how the
   program works.

If you enjoy Fractint, you're sure to enjoy Fractal Creations. The book
includes Fractint and is an excellent companion to the program. If you
use the Windows environment, be sure to pick up a copy of Fractals for

Windows as well.

A great fractals newsletter is "Amygdala" published by Rollo Silver.
You'll find equal parts fractal algorithms, humor, reviews, and ideas.
Write to:
    Amygdala
    Box 219
    San Cristobal, NM 87564
    USA
    Email:rsilver@lanl.gov
    Phone: 505-586-0197

Another great fractals newsletter (this one based in the UK) is
"FRAC'Cetera", a disk-based fractal/chaos resource, for PCs and
compatibles, distributed on 3.5" HD disk, published by Jon Horner.
Contact:

    Jon Horner
    FRAC'Cetera
    Le Mont Ardaine
    Rue des Ardaines
    ST Peters
    Guernsey GY7 9EU, CI, UK
    Email: 100112.1700@compuserve.com
    PH: (44) 01481 63689

Several Fractint enthusiasts are selling Fractal CDs. Two of the best are
called "Fractal Frenzy" by Lee Skinner, and "Fractography" by Caren Park.
Highly recommended original artwork in a variety of graphics formats.

You can receive the "Fractal Frenzy CD" by sending $39.95US + $5.00 S&H to
    Walnut Creek CDROM
    1537 Palos Verdes Mall, Suite 260
    Walnut Creek, CA 94596

and the "Fractography" CD by sending $30.00US + $5.00 S&H (in US/Canada) to
    Lost and Found Books
    485 Front Street N, Suite A
    Issaquah, WA 98027-2900

Michael Peters (author of PARTOBAT) and Randall Scott have written a fractal
program called HOP based on the Martin orbit fractals. This program is much
narrower than Fractint in the kind of thing that it does, but has many more
animation effects and makes a great screen saver. Michael sent us the
algorithms for the chip, quadruptwo, and threeply fractal types to give us
a taste. The file is called HOPZIP.EXE in LIB 4 of CompuServe's GRAPHDEV
forum.

1.22 Bibliography

BARNSLEY, Michael: "Fractals Everywhere," Academic Press, 1988.

DAVENPORT, Clyde: "A Hypercomplex Calculus with Applications to Relativity",
    ISBN 0-9623837-0-8. This self-published expansion of Mr. Davenport's
    Master's thesis makes the case for using hypercomplex numbers rather than

quaternions. This book provided the background for Fractint's
implementation of hypercomplex fractals.

DEWDNEY, A. K., "Computer Recreations" columns in "Scientific American" --
8/85, 7/87, 11/87, 12/88, 7/89.

FEDER, Jens: "Fractals," Plenum, 1988.
Quite technical, with good coverage of applications in fluid
percolation, game theory, and other areas.

GLEICK, James: "Chaos: Making a New Science," Viking Press, 1987.
The best non-technical account of the revolution in our understanding
of dynamical systems and its connections with fractal geometry.

MANDELBROT, Benoit: "The Fractal Geometry of Nature," W. H. Freeman & Co.,
1982.
An even more revised and expanded version of the 1977 work. A rich and
sometimes confusing stew of formal and informal mathematics, the
prehistory of fractal geometry, and everything else. Best taken in
small doses.

MANDELBROT, Benoit: "Fractals: Form, Chance, and Dimension," W. H. Freeman
& Co., 1977.
A much revised translation of "Les objets fractals: forme, hasard, et
dimension," Flammarion, 1975.

PEITGEN, Heinz-Otto & RICHTER, Peter: "The Beauty of Fractals," Springer-
Verlag, 1986.
THE coffee-table book of fractal images, knowledgeable on computer
graphics as well as the mathematics they portray.

PEITGEN, Heinz-Otto & SAUPE, Ditmar: "The Science of Fractal Images,"
Springer-Verlag, 1988.
A fantastic work, with a few nice pictures, but mostly filled with
equations!!!

PICKOVER, Clifford: "Computers, Pattern, Chaos, and Beauty," St. Martin's
Press, 1990.

SCHROEDER, Manfred: "Fractals, Chaos, Power Laws," W. H. Freeman
& Co., 1991.

WEGNER, Timothy: "Image Lab, Second Edition", Waite Group Press, to be
released in 1995. Learn how to create fractal animations, fractal
RDS stereo images, and how to use Fractint with other image creation
and processing tools such as Piclab, POV-Ray and Polyray ray tracers.

WEGNER, Timothy & TYLER, Bert: "Fractal Creations, Second Edition" Waite
Group Press, 1993
This is the definitive Fractint book. Spectacular color plate section,
totally new and expanded fractal type descriptions, annotated PAR files,
source code secrets, and a CD filled to the brim with spectacular fractals.

WEGNER, Timothy, TYLER, Bert, PETERSON, Mark, and Branderhorst, Pieter:
"Fractals for Windows," Waite Group Press, 1992.
This book is to Winfract (the Windows version of Fractint) what

    "Fractal Creations" is to Fractint.

1.23   Other Programs

WINFRACT. Bert Tyler has ported Fractint to run under Windows 3! The same
underlying code is used, with a Windows user interface. Winfract has
almost all the functionality of Fractint – the biggest difference is the
absence of a zillion weird video modes. Fractint for DOS will continue to
be the definitive version. Winfract is available from CompuServe in
GRAPHDEV Lib 4, as WINFRA.ZIP (executable) and WINSRC.ZIP (source).

PICLAB, by Lee Crocker – a freeware image manipulation utility available
from CompuServe in PICS Lib 10, as PICLAB.EXE. PICLAB can do very
sophisticated resizing and color manipulation of GIF and TGA files. It
can be used to reduce 24 bit TGA files generated with the Fractint
"lightname" option to GIF files.

FDESIGN, by Doug Nelson (CIS ID 70431,3374) – a freeware IFS fractal
generator available from CompuServe in GRAPHDEV Lib 4, and probably on your
local BBS. This program requires a VGA adapter and a Microsoft-compatible
mouse, and a floating point coprocessor is highly recommended. It
generates IFS fractals in a *much* more intuitive fashion than Fractint.
It can also (beginning with version 3.0) save its IFS formulas in
Fractint-style .IFS files.

ACROSPIN, by David Parker – An inexpensive commercial program that reads
an object definition file and creates images that can be rapidly rotated
in three dimensions. The Fractint "orbitsave=yes" option creates files that
this program can read for orbit-type fractals and IFS fractals. Contact:
   David Parker   801-966-2580
   P O Box 26871   800-227-6248
   Salt Lake City, UT 84126-0871
;
;

1.24   Revision History

   Please select one of:

      Version 18

      Version 17

      Version 16

      Version 15

      Versions 12 through 14

      Versions 1 through 11

;

1.25   version 18

Versions 18.1 and 18.2 are bug-fix releases for version 18.0. Changes from
18.1 to 18.2 include:

 The <b> command now causes filenames only to be written in PAR files.

 Fractint will now search directories in the PATH for files not found in the
 requested the requested directory or the current directory. If you place
 .MAP, .FRM, etc. in directories in your PATH, then Fractint will find them.

 Fixed bug that caused fractals using PI symmetry to fail at high resolution.

 Fractals interrupted with <3> or <r> can now resume.

 The palette editor's <u> (undo) now works.

 The <s> command in orbit/Julia window mode is no longer case sensitive.

 Added warnings that the POV-Ray output is obsolete (but has been left in).
 Use POV-Ray's height field facility instead or create and convert RAW files.

 Fixed several IFS bugs.

Changes from 18.0 to 18.1 include:

 Overlay tuning – the Mandelbrot/Julia Set fractals are now back up
 to 17.x speeds

 Disk Video modes now work correctly with VESA video adapters (they
 used to use the same array for different purposes, confusing each other)

 1024x768x256 and 2048x2048x256 disk video modes work again

 Parameter-file processing no longer crashes Fractint if it attempts to
 run a formula requiring access to a non-existent FRM file

 IFS arrays no longer overrun their array space

 type=cellular fixes

 "autologmap=2" now correctly picks up the minimum color

 The use of disk-video mode with random-access fractal types is now
 legal (it generates a warning message but lets you proceed if you
 really want to)

 The Lsystems "spinning-wheel" now spins slower (removing needless overhead)

 Changes to contributors' addresses in the Help screens

(The remainder of this "new features" section is from version 18.0)

New fractal types:

 19 new fractal types, including:

 New fractal types - 'lambda(fn||fn)', 'julia(fn||fn)', 'manlam(fn||fn)',
 'mandel(fn||fn)', 'halley', 'phoenix', 'mandphoenix', 'cellular',
 generalized bifurcation, and 'bifmay' - from Jonathan Osuch.

 New Mandelcloud, Quaternion, Dynamic System, Cellular Automata fractal
 types from Ken Shirriff.

 New HyperComplex fractal types from Timothy Wegner

 New ICON type from Dan Farmer, including a PAR file of examples.

 New Frothy Basin fractal types (and PAR entries) by Wesley Loewer

 MIIM (Modified Inverse Iteration Method) implementation of Inverse Julia
 from Michael Snyder.

 New Inverse Julia fractal type from Juan Buhler.

 New floating-point versions of Markslambda, Marksmandel, Mandel4,
 and Julia4 types (chosen automatically if the floating-point option
 is enabled).

New options/features:

 New assembler-based parser logic from Chuck Ebbert - significantly
 faster than the C-based code it replaces!

 New assembler-based Lyapunov logic from Nicholas Wilt and Wes Loewer.
 Roughly six times faster than the old version!

 New Orbits-on-a-window / Julia-in-a-window options:
  1) The old Overlay option is now '#' (Shift-3).
  2) During generation, 'O' brings up orbits (as before) - after
     generation, 'O' brings up new orbits Windows mode.
  3) Control-O brings up new orbits Windows mode at any time.
  4) Spacebar toggles between Inverse Julia mode and the Julia set and
     back to the Mandelbrot set.
 These new "in-a-window" modes are really neat! See Orbits Window
 and Julia Toggle Spacebar Commands for details.

 New multi-image GIF support in the <B> command. You can now generate
 65535x65535x256 fractal images using Fractint (if you have the disk
 space, of course). This option builds special PAR entries and a
 MAKEMIG.BAT file that you later use to invoke Fractint multiple times
 to generate individual sections of the image and (in a final step)
 stitch them all together. If your other software can't handle
 Multiple-image GIFs, a SIMPLGIF program is also supplied that converts
 MIGS into simple-image GIFs. Press F1 at the <B> prompts screen for
 details.

 Fractint's decoder now handles Multi-Image Gifs.

 New SuperVGA/VESA Autodetect logic from the latest version of

VGAKIT. Sure hope we didn't break anything.

New register-compatible 8514/A code from Jonathan Osuch. By default,
Fractint now looks first for the presence of an 8514/A register-compatible
adapter and then (and only if it doesn't find one) the presence of the
8514/A API (IE, HDILOAD is no longer necessary for register-compatible
"8514/a" adapters). Fractint can be forced to use the 8514/A API by using
a new command-line option, "afi=yes". Jonathan also added ATI's
"8514/a-style" 800x600x256 and 1280x1024x16 modes.

New XGA-detection logic for ISA-based XGA-2 systems.

The palette editor now has a "freestyle" editing option. See
Palette Editing Commands for details.

Fractint is now more "batch file" friendly. When running Fractint from
a batch file, pressing any key will cause Fractint to exit with an
errorlevel = 2. Any error that interrupts an image save to disk will
cause an exit with errorlevel = 2. Any error that prevents an
image from being generated will cause an exit with errorlevel = 1.

New Control-X, Control-Y, and Control-Z options flip a fractal image
along the X-axis, Y-axis, and Origin, respectively.

New area calculation mode in TAB screen from Ken Shirriff
(for accuracy use inside=0).

The TAB screen now indicates when the Integer Math algorithms are in use.

The palette must now be explicitly changed, it will not reset to the default
unexpectedly when doing things like switching video modes.

The Julibrot type has been generalized.
Julibrot fractals can now be generated from PAR files.

Added <b> command support for viewwindows.

Added room for two additional PAR comments in the <B> command

New coloring method for IFS shows which parts of fractal came from
which transform.

Added attractor basin phase plotting for Julia sets from Ken Shirriff.

Improved finite attractor code to find more attractors from Ken Shirriff.

New zero function, to be used in PAR files to replace old integer tan, tanh

Debugflag=10000 now reports video chipset in use as well as CPU/FPU
type and available memory

Added 6 additional parameters for params= for those fractal types that
need them.

New 'matherr()' logic lets Fractint get more aggressive when these errors
happen.

New autologmap option (log=+-2) from Robin Bussell that ensures that
all palette values are used by searching the screen border for the lowest
value and then setting log= to +- that color.

Two new diffusion options - falling and square cavity.

Three new Editpal commands: '!', '@' and '#' commands (that's
<shift-1>, <shift-2>, and <shift-3>) to swap R<->G, G<->B, R<->B.

Parameter files now use a slightly shorter maximum line length, making
them a bit more readable when stuffed into messages on CompuServe.

Plasma now has 16-bit .POT output for use with Ray tracers. The "old"
algorithm has been modified so that the plasma effect is independent
of resolution.

Slight modification to the Raytrace code to make it compatible with
Rayshade 4.0 patch level 6.

Improved boundary-tracing logic from Wesley Loewer.

Command-line parameters can now be entered on-the-fly using the <g> key
thanks to Ken Shirriff.

Dithered gif images can now be loaded onto a b/w display.
Thanks to Ken Shirriff.

Pictures can now be output as compressed PostScript.
Thanks to Ken Shirriff.

Periodicity is a new inside coloring option.
Thanks to Ken Shirriff.

Fixes: symmetry values for the SQR functions, bailout for the floating-pt
versions of 'lambdafn' and 'mandelfn' fractals from Jonathan Osuch.

"Flip", "conj" operators are now selectable in the parser

New DXF Raytracing option from Dennis Bragg.

Improved boundary-tracing logic from Wesley Loewer.

New MSC7-style overlay structure is used if MAKEFRAC.BAT specifies MSC7.
(with new FRACTINT.DEF and FRACTINT.LNK files for MSC7 users). Several
modules have been re-organized to take advantage of this new overlay
capability if compiled under MSC7.

Fractint now looks first any embedded help inside FRACTINT.EXE, and then
for an external FRACTINT.HLP file before giving up. Previous releases
required that the help text be embedded inside FRACTINT.EXE.

Bug fixes:

Corrected formulas displayed for Marksmandel, Cmplxmarksmandel, and
  associated julia types.

BTM and precision fixes.

Symmetry logic changed for various "outside=" options

Symmetry value for EXP function in lambdafn and lambda(fn||fn) fixed.

Fixed bug where math errors prevented save in batch mode.

The <3> and <r> commands no longer destroy image -- user can back out
with ESC and image is still there.

Fixed display of correct number of Julibrot parameters, and Julibrot
relaxes and doesn't constantly force ALTERN.MAP.

Fixed tesseral type for condition when border is all one color but center
contains image.

Fixed integer mandel and julia when used with parameters > +1.99 and < -1.99

Eliminated recalculation when generating a julia type from a mandelbrot
type when the 'z' screen is viewed for the first time.

Minor logic change to prevent double-clutching into and out of graphics
mode when pressing, say, the 'x' key from a menu screen.

Changed non-US phone number for the Houston Public (Software) Library

The "Y" screen is now "Extended Options" instead of "Extended Doodads"

...and probably a lot more bux-fixes that we've since forgotten that
we've implemented.

;

1.26 version 17

Version 17.2, 3/92

  - Fixed a bug which caused Fractint to hang when a Continuous Potential
    Bailout value was set (using the 'Y') screen and then the 'Z' screen
    was activated.
  - fixed a bug which caused "batch=yes" runs to abort whenever any
    key was pressed.
  - bug-fixes in the Stereo3D/Targa logic from Marc Reinig.
  - Fractint now works correctly again on FPU-less 8088s when
    zoomed deeply into the Mandelbrot/Julia sets
  - The current image is no longer marked as "not resumable" on a
    Shell-To-Dos ("D") command.
  - fixed a bug which prevented the "help" functions from working
    properly during fractal-type selection for some fractal types.

Version 17.1, 3/92

  - fixed a bug which caused PCs with no FPU to lock up when they attempted
    to use some fractal types.

  – fixed a color-cycling bug which caused the palette to single-step
    when you pressed ESCAPE to exit color-cycling.
  – fixed the action of the '<' and '>' keys during color-cycling.

Version 17.0, 2/92

– New fractal types (but of course!):

Lyapunov Fractals from Roy Murphy (see Lyapunov Fractals for details)

'BifStewart' (Stewart Map bifurcation) fractal type and new bifurcation
parameters (filter cycles, seed population) from Kevin Allen.

Lorenz3d1, Lorenz3d3, and Lorenz3d4 fractal types from Scott Taylor.
Note that a bug in the Lorenz3d1 fractal prevents zooming-out from
working with it at the moment.

Martin, Circle, and Hopalong (culled from Dewdney's Scientific American
Article)

Lots of new entries in fractint.par.

New ".L" files (TILING.L, PENROSE.L)

New 'rand()' function added to the 'type=formula' parser

– New fractal generation options:

New 'Tesseral' calculation algorithm (use the 'X' option list to
select it) from Chris Lusby Taylor.

New 'Fillcolor=' option shows off Boundary Tracing and Tesseral structure

inside=epscross and inside=startrail options taken from a paper by
Kenneth Hooper, with credit also to Clifford Pickover

New Color Postscript Printer support from Scott Taylor.

Sound= command now works with <O>rbits and <R>ead commands.

New 'orbitdelay' option in X-screen and command-line interface

New "showdot=nn" command-line option that displays the pixel currently
being worked on using the specified color value (useful for those llooooonngg
images being calculated using solid guessing – "where is it now?").

New 'exitnoask=yes' commandline/SSTOOLS.INI option to avoid the final
"are you sure?" screen

New plasma-cloud options. The interface at the moment (documented here and
  here only because it might change later) lets you:
  – use an alternate drawing algorithm that gives you an earlier preview
    of the finished image.
  – re-generate your favorite plasma cloud (say, at a higher resolution)
    by forcing a re-select of the random seed.

New 'N' (negative palette) option from Scott Taylor – the documentation at

this point is: Pressing 'N' while in the palette editor will invert
each color. It will convert only the current color if it is in 'x' mode,
a range if in 'y' mode, and every color if not in either the 'x' or 'y' mode.

- Speedups:

New, faster floating-point Mandelbrot/Julia set code from Wesley Loewer,
Frank Fussenegger and Chris Lusby Taylor (in separate contributions).

Faster non-386 integer Mandelbrot code from Chris Lusby Taylor, Mike Gelvin
and Bill Townsend (in separate contributions)

New integer Lsystems logic from Nicholas Wilt

Finite-Attractor fixups and Lambda/mandellambda speedups from Kevin Allen.

GIF Decoder speedups from Mike Gelvin

- Bug-fixes and other enhancements:

Fractint now works with 8088-based AMSTRAD computers.

The video logic is improved so that (we think) fewer video boards will need
"textsafe=save" for correct operation.

Fixed a bug in the VESA interface which effectively messed up adapters
with unusual VESA-style access, such as STB's S3 chipset.

Fixed a color-cycling bug that would at times restore the wrong colors
to your image if you exited out of color-cycling, displayed a 'help'
screen, and then returned to the image.

Fixed the XGA video logic so that its 256-color modes use the same
default 256 colors as the VGA adapter's 320x200x256 mode.

Fixed the 3D bug that caused bright spots on surfaces
to show as black blotches of color 0 when using a light source.

Fixed an image-generation bug that sometimes caused image regeneration
to restart even if not required if the image had been zoomed in
to the point that floating-point had been automatically activated.

Added autodetection and 640x480x256 support for the Compaq Advanced VGA
Systems board - I wonder if it works?

Added VGA register-compatible 320x240x256 video mode.

Fixed the "logmap=yes" option to (again) take effect for continuous potential
images. This was broken in version 15.x.

The colors for the floating-point algorithm of the Julia fractal
now match the colors for the integer algorithm.

If the GIF Encoder (the "Save" command) runs out of disk space, it now
tells you about it.

If you select both the boundary-tracing algorithm and either "inside=0"

or "outside=0", the algorithm will now give you an error message instead
of silently failing.

Updated 3D logic from Marc Reinig.

Minor changes to permit IFS3D fractal types to be handled properly
using the "B" command.

Minor changes to the "Obtaining the latest Source" section to refer
to BBS access (Peter Longo's) and mailed diskettes (the Public (Software)
Library).

1.27   version 16

Version 16.12, 8/91

  Fix to cure some video problems reported with Amstrad
    8088/8086-based PCs.

Version 16.11, 7/91

  SuperVGA Autodetect fixed for older Tseng 3000 adapters.

  New "adapter=" options to force the selection of specific SuperVGA
    adapter types. See Video Parameters for details.

  Integer/Floating-Point math toggle is changed only temporarily
    if floating-point math is forced due to deep zooming.

  Fractint now survives being modified by McAfee's "SCAN /AV" option.

  Bug Fixes for Acrospin interface, 3D "Light Source Before
    Transformation" fill type, and GIF decoder.

  New options in the <Z> parameters screen allow you to directly
    enter image coordinates.

  New "inside=zmag" and "outside=real|imag|mult|summ" options.

  The GIF Decoder now survives reading GIF files with a local color map.
  Improved IIT Math Coprocessor support.

  New color-cycling single-step options, '<' and '>'.

Version 16.0, 6/91

  Integrated online help / fractint.doc system from Ethan Nagel.
    To create a printable fractint.doc file see Startup Parameters.

  Over 350 screens of online help! Try pressing <F1> just about anywhere!

  New "autokey" feature. Type "demo" to run the included demo.bat and
    demo.key files for a great demonstration of Fractint.
    See Autokey Mode for details.

New <@> command executes a saved set of commands. The <b> command has
  changed to write the current image's parameters as a named set of
  commands in a structured file. Saved sets of commands can subsequently
  be executed with the <@> command.
  See Parameter Save/Restore Commands.
  A default "fractint.par" file is included with the release.

New <z> command allows changing fractal type-specific parameters without
  going back through the <t> (fractal type selection) screen.

Ray tracer interface from Marc Reinig, generates 3d transform output for a
  number of ray tracers; see Interfacing with Ray Tracing Programs

Selection of video modes and structure of "fractint.cfg" have changed. If
  you have a customized fractint.cfg file, you'll have to rebuild it based
  on this release's version. You can customize the assignment of your
  favorite video modes to function keys; see Video Mode Function Keys.
  <delete> is a new command key which goes directly to video mode selection.

New "cyclerange" option (command line and <y> options screen) from Hugh
  Steele. Limits color cycling to a specified range of colors.

Improved Distance Estimator Method algorithm from Phil Wilson.

New "ranges=" option from Norman Hills.
  See Logarithmic Palettes and Color Ranges for details.

type=formula definitions can use "variable functions" to select
  sin, cos, sinh, cosh, exp, log, etc at run time;
  new built-ins tan, tanh, cotan, cotanh, and flip
  are available with type=formula; see Type Formula

New <w> command in palette editing mode to convert image to greyscale

All "fn" fractal types (e.g. fn*fn) can now use new functions tan, tanh,
  cotan, cotanh, recip, and ident; bug in prior cos function fixed, new
  function cosxx (conjugate of cos) is the old erroneous cos calculation

New L-Systems from Herb Savage
New IFS types from Alex Matulich
Many new formulas in fractint.frm, including a large group from
  JM Collard-Richard
Generalized type manzpwr with complex exponent per Lee Skinner's request
Initial orbit parameter added to Gingerbreadman fractal type

New color maps (neon, royal, volcano, blues, headache) from Daniel Egnor

IFS type has changed to use a single file containing named entries
  (instead of a separate xxx.ifs file per type); the <z> command brings up
  IFS editor (used to be <i> command). See Barnsley IFS Fractals.

Much improved support for PaintJet printers; see PaintJet Parameters

From Scott Taylor:
  Support for plotters using HP-GL; see Plotter Parameters
  Lots of new PostScript halftones; see PostScript Parameters
  "printer=PS[L]/0/..." for full page PostScript; see PostScript Parameters

   Option to drive printer ports directly (faster); see Printer Parameters
   Option to change printer end of line control chars; see Printer Parameters

Support for XGA video adapter
Support for Targa+ video adapter
16 color VGA mode enhancements:
   Now use the first 16 colors of .map files to be more predictable
   Palette editor now works with these modes
   Color cycling now works properly with these modes
Targa video adapter fixes; Fractint now uses (and requires) the "targa"
   and "targaset" environment variables for Targa systems
"vesadetect=no" parameter to bypass use of VESA video driver; try
   this if you encounter video problems with a VESA driver
Upgraded video adapter detect and handling from John Bridges; autodetect
   added for NCR, Trident 8900, Tseng 4000, Genoa (this code
   is from a beta release of VGAKIT, we're not sure it all works yet)

Zoom box is included in saved/printed images (but, is not recognized as
   anything special when such an image is restored)

The colors numbers reserved by the palette editor are now selectable with
   the new <v> palette editing mode command

Option to use IIT floating point chip's special matrix arithmetic for
   faster 3D transforms; see "fpu=" in Startup Parameters

Disk video cache increased to 64k; disk video does less seeking when
   running to real disk
Faster floating point code for 287 and higher fpus, for types mandel,
   julia, barnsleyj1/m1/j2/m2, lambda, manowar, from Chuck Ebbert

"filename=.xxx" can be used to set default <r> function file mask

Selection of type formula or lsys now goes directly to entry selection
   (file selection step is now skipped); to change to a different file, use
   <F6> from the entry selection screen

Three new values have been added to the textcolors= parameter; if you use
   this parameter you should update it by inserting values for the new 6th,
   7th, 9th, and 13th positions; see "textcolors=" in Color Parameters

The formula type's imag() function has changed to return the result as
   a real number

Fractal type-specific parameters (entered after selecting a new fractal
   type with <T>) now restart at their default values each time you select
   a new fractal type

Floating point input fields can now be entered in scientific notation (e.g.
   11.234e-20). Entering the letters "e" and "p" in the first column causes
   the numbers e=2.71828... and pi=3.14159... to be entered.

New option "orbitsave=yes" to create files for Acrospin for
   some types (see Barnsley IFS Fractals, Orbit Fractals,
   Acrospin)

Bug fixes:

```
      Problem with Hercules adapter auto-detection repaired.
      Problems with VESA video adapters repaired (we're not sure we've got them
        all yet...)
      3D transforms fixed to work at high resolutions (> 1000 dots).
      3D parameters no longer clobbered when restoring non-3D images.
      L-Systems fixed to not crash when order too high for available memory.
      PostScript EPS file fixes.
      Bad leftmost pixels with floating point at 2048 dot resolution fixed.
      3D transforms fixed to use current <x> screen float/integer setting.
      Restore of images using inversion fixed.
      Error in "cos" function (used with "fn" type fractals) fixed; prior
        incorrect function still available as "cosxx" for compatibility

   Old 3D=nn/nn/nn/... form of 3D transform parameters no longer supported

   Fractint source code now Microsoft C6.00A compatible.
;
;
```

1.28   version 15

Version 15.11, 3/91, companion to Fractal Creations, not for general
release

   Autokey feature, IIT fpu support, and some bug fixes publicly released in
   version 16.

Version 15 and 15.1, 12/90

   New user interface! Enjoy! Some key assignments have changed and some have
     been removed.
   New palette editing from Ethan Nagel.
   Reduced memory requirements - Fractint now uses overlays and will run on a
     512K machine.
   New <v>iew command: use to get small window for fast preview, or to setup
     an image which will eventually be rendered on hard copy with different
     aspect ratio
   L-System fractal type from Adrian Mariano
   Postscript printer support from Scott Taylor
   Better Tandy video support and faster CGA video from Joseph A Albrecht
   16 bit continuous potential files have changed considerably; see the
     Continuous Potential section for details. Continuous potential is now
     resumable.
   Mandelbrot calculation is faster again (thanks to Mike Gelvin) - double
     speed in 8086 32 bit case
   Compressed log palette and sqrt palette from Chuck Ebbert
   Calculation automatically resumes whenever current image is resumable and
     is not paused for a visible reason.
   Auto increment of savename changed to be more predictable
   New video modes:
     trident 1024x768x256 mode
     320x480x256 tweak mode (good for reduced 640x480 viewing)
     changed NEC GB-1, hopefully it works now
   Integer mandelbrot and julia now work with periodicitycheck

```
Initial zoombox color auto-picked for better contrast (usually)
New adapter=cga|ega|mcga|vga for systems having trouble with auto-detect
New textsafe=no|yes for systems having trouble with garbled text mode
<r> and <3> commands now present list of video modes to pick from; <r> can
  reduce a non-standard or unviewable image size.
Diffusion fractal type is now resumable after interrupt/save
Exitmode=n parameter, sets video mode to n when exiting from fractint
When savetime is used with 1 and 2 pass and solid guessing, saves are
  deferred till the beginning of a new row, so that no calculation time is
  lost.
3d photographer's mode now allows the first image to be saved to disk
textcolors=mono|12/34/56/... -- allows setting user interface colors
Code (again!) compilable under TC++ (we think!)
.TIW files (from v9.3) are no longer supported as input to 3D
  transformations
bug fixes:
  multiple restores (msc 6.0, fixed in 14.0r)
  repeating 3d loads problem; slow 3d loads of images with float=yes
  map= is now a real substitute for default colors
  starfield and julibrot no longer cause permanent color map replacement
  starfield parameters bug fix - if you couldn't get the starfield
  parameters to do anything interesting before, try again with this
  release
  Newton and newtbasin orbit display fixed

Version 15.1:

  Fixed startup and text screen problems on systems with VESA compliant
    video adapters.
  New textsafe=save|bios options.
  Fixes for EGA with monochrome monitor, and for Hercules Graphics Card.
    Both should now be auto-detected and operate correctly in text modes.
    Options adapter=egamono and adapter=hgc added.
  Fixed color L-Systems to not use color 0 (black).
  PostScript printing fix.
;
```

1.29   versions 12 through 14

```
Version 14, 8/90

  LAST MINUTE NEWS FLASH!
    CompuServe announces the GIF89a on August 1, 1990, and Fractint supports
    it on August 2! GIF files can now contain fractal information!  Fractint
    now saves its files in the new GIF89a format by default, and uses .GIF
    rather than .FRA as a default filetype.  Note that Fractint still
    *looks* for a .FRA file on file restores if it can't find a .GIF file,
    and can be coerced into using the old GIF87a format with the new
    'gif87a=yes' command-line option.

  Pieter Branderhorst mounted a major campaign to get his name in lights:
  Mouse interface:  Diagonals, faster movement, improved feel. Mouse button
    assignments have changed - see the online help.
  Zoom box enhancements:  The zoom box can be rotated, stretched, skewed,
    and panned partially offscreen.  See "More Zoom Box Commands".
```

FINALLY!! You asked for it and we (eventually, by talking Pieter into it
  [actually he grabbed it]) did it! Images can be saved before
  completion, for a subsequent restore and continue. See "Interrupting
  and Resuming" and "Batch Mode".
Off-center symmetry: Fractint now takes advantage of x or y axis symmetry
  anywhere on the screen to reduce drawing time.
Panning: If you move an image up, down, left, or right, and don't change
  anything else, only the new edges are calculated.
Disk-video caching – it is now possible, reasonable even, to do most
  things with disk video, including solid guessing, 3d, and plasma.
Logarithmic palette changed to use all colors. It now matches regular
  palette except near the "lake". "logmap=old" gets the old way.
New "savetime=nnn" parameter to save checkpoints during long calculations.
Calculation time is shown in <Tab> display.

Kevin C Allen   Finite Attractor, Bifurcation Engine, Magnetic fractals...
Made Bifurcation/Verhulst into a generalized Fractal Engine (like
  StandardFractal, but for Bifurcation types), and implemented periodicity
  checking for Bifurcation types to speed them up.
Added Integer version of Verhulst Bifurcation (lots faster now). Integer
  is the default. The Floating-Point toggle works, too.
Added NEW Fractal types BIFLAMBDA, BIF+SINPI, and BIF=SINPI. These are
  Bifurcation types that make use of the new Engine. Floating-
  point/Integer toggle is available for BIFLAMBDA. The SINPI types are
  Floating-Point only, at this time.
Corrected the generation of the MandelLambda Set. Sorry, but it's always
  been wrong (up to v 12, at least). Ask Mandelbrot !
Added NEW Fractal types MAGNET1M, MAGNET1J, MAGNET2M, MAGNET2J from "The
  Beauty of Fractals". Floating-Point only, so far, but what do you
  expect with THESE formulae ?!
Added new symmetry types XAXIS NOIMAG and XAXIS NOREAL, required by the
  new MAGNETic Fractal types.
Added Finite Attractor Bailout (FAB) logic to detect when iterations are
  approaching a known finite attractor. This is required by the new
  MAGNETic Fractal types.
Added Finite Attractor Detection (FAD) logic which can be used by *SOME*
  Julia types prior to generating an image, to test for finite attractors,
  and find their values, for use by FAB logic. Can be used by the new
  MAGNETic Fractal Types, Lambda Sets, and some other Julia types too.

Mike Burkey sent us new tweaked video modes:
  VGA   – 400x600x256   376x564x256   400x564x256
  ATI VGA – 832x612x256
New HP Paintjet support from Chris Martin
New "FUNCTION=" command to allow substition of different transcendental
  functions for variables in types (allows one type with four of these
  variables to represent 7*7*7*7 different types!
ALL KINDS of new fractal types, some using "FUNCTION=": fn(z*z), fn*fn,
  fn*z+z, fn+fn, sqr(1/fn), sqr(fn), spider, tetrate, and Manowar. Most of
  these are generalizations of formula fractal types contributed by Scott
  Taylor and Lee Skinner.
Distance Estimator logic can now be applied to many fractal types using
  distest= option. The types "demm" and "demj" have been replaced by
  "type=mandel distest=nnn" and "type=julia distest=nnn"
Added extended memory support for diskvideo thanks to Paul Varner
Added support for "center and magnification" format for corners.
Color 0 is no longer generated except when specifically requested with

       inside= or outside=.
    Formula name is now included in <Tab> display and in <S>aved images.
    Bug fixes - formula type and diskvideo, batch file outside=-1 problem.
    Now you can produce your favorite fractal terrains in full color instead
       of boring old monochrome! Use the fullcolor option in 3d! Along with a
       few new 3D options.
    New "INITORBIT=" command to allow alternate Mandelbrot set orbit
       initialization.

Version 13.0, 5/90

    F1 was made the help key.
       Use F1 for help
       Use F9 for EGA 320x200x16 video mode
       Use CF4 for EGA 640x200x16 mode (if anybody uses that mode)
    Super-Solid-guessing (three or more passes) from Pieter Branderhorst
       (replaces the old solid-guessing mode)
    Boundary Tracing option from David Guenther ("fractint passes=btm", or use
       the new 'x' options screen)
    "outside=nnn" option sets all points not "inside" the fractal to color
       "nnn" (and generates a two-color image).
    'x' option from the main menu brings up a full-screen menu of many popular
       options and toggle switches
    "Speed Key" feature for fractal type selection (either use the cursor keys
       for point-and-shoot, or just start typing the name of your favorite
       fractal type)
    "Attractor" fractals (Henon, Rossler, Pickover, Gingerbread)
    Diffusion fractal type by Adrian Mariano
    "type=formula" formulas from Scott Taylor and Lee H. Skinner.
    "sound=" options for attractor fractals. Sound=x plays speaker tones
       according to the 'x' attractor value Sound=y plays speaker tones
       according to the 'y' attractor value. Sound=z plays speaker tones
       according to the 'z' attractor value (These options are best invoked
       with the floating-point algorithm flag set.)
    "hertz=" option for adjusting the "sound=x/y/z" output.
    Printer support for color printers (printer=color) from Kurt Sowa
    Trident 4000 and Oak Technologies SuperVGA support from John Bridges
    Improved 8514/A support (the zoom-box keeps up with the cursor keys now!)
    Tandy 1000 640x200x16 mode from Brian Corbino (which does not, as yet,
       work with the F1(help) and TAB functions)
    The Julibrot fractal type and the Starmap option now automatically verify
       that they have been selected with a 256-color palette, and search for,
       and use, the appropriate GLASSESn.MAP or ALTERN.MAP palette map when
       invoked. *You* were supposed to be doing that manually all along, but
       you probably never read the docs, huh?
    Bug Fixes:
       TAB key now works after R(estore) commands
       PS/2 Model 30 (MCGA) adapters should be able to select 320x200x256 mode
          again (we think)
       Everex video adapters should work with the Autodetect modes again (we
          think)

Version 12.0, 3/90

    New SuperVGA Autodetecting and VESA Video modes (you tell us the

```
     resolution you want, and we'll figure out how to do it)
  New Full-Screen Entry for most prompting
  New Fractal formula interpreter ('type=formula') - roll your own fractals
    without using a "C" compiler!
  New 'Julibrot' fractal type
  Added floating point option to all remaining fractal types.
  Real (funny glasses) 3D - Now with "real-time" lorenz3D!!
  Non-Destructive <TAB> - Check out what your fractal parameters are without
    stopping the generation of a fractal image
  New Cross-Hair mode for changing individual palette colors (VGA only)
  Zooming beyond the limits of Integer algorithms (with automatic switchover
    to a floating-point algorithm when you zoom in "too far")
  New 'inside=bof60', 'inside=bof61' ("Beauty of Fractals, Page nn") options
  New starmap ('a' - for astrology? astronomy?) transformation option
  Restrictions on the options available when using Expanded Memory
    "Disk/RAM" video mode have been removed
  And a lot of other nice little clean-up features that we've already
    forgotten that we've added...
  Added capability to create 3D projection images (just barely) for people
    with 2 or 4 color video boards.
;
```

1.30   versions 1 through 11

```
Version 11.0, 1/90

  More fractal types
    mandelsinh/lambdasinh        mandelcosh/lambdacosh
    mansinzsqrd/julsinzsqrd      mansinexp/julsinexp
    manzzprw/julzzpwr            manzzpower/julzpower
    lorenz (from Rob Beyer)      lorenz3d
    complexnewton                complexbasin
    dynamic                      popcorn
  Most fractal types given an integer and a floating point algorithm.
    "Float=yes" option now determines whether integer or floating-point
    algorithms are used for most fractal types.  "F" command toggles the use
    of floating-point algorithms, flagged in the <Tab> status display
  8/16/32/../256-Way decomposition option (from Richard Finegold)
  "Biomorph=", "bailout=", "symmetry="  and "askvideo=" options
  "T(ransform)" option in the IFS editor lets you select 3D options (used
    with the Lorenz3D fractal type)
  The "T(ype)" command uses a new "Point-and-Shoot" method of selecting
    fractal types rather than prompting you for a type name
  Bug fixes to continuous-potential algorithm on integer fractals, GIF
    encoder, and IFS editor


Version 10.0, 11/89

  Barnsley IFS type (Rob Beyer)
  Barnsley IFS3D type
  MandelSine/Cos/Exp type
  MandelLambda/MarksLambda/Unity type
  BarnsleyM1/J1/M2/J2/M3/J3 type
  Mandel4/Julia4 type
```

```
  Sierpinski gasket type
  Demm/Demj and bifurcation types (Phil Wilson), "test" is "mandel" again
  <I>nversion command for most fractal types
  <Q>uaternary decomposition toggle and "DECOMP=" argument
  <E>ditor for Barnsley IFS parameters
  Command-line options for 3D parameters
  Spherical 3D calculations 5x faster
  3D now clips properly to screen edges and works at extreme perspective
  "RSEED=" argument for reproducible plasma clouds
  Faster plasma clouds (by 40% on a 386)
  Sensitivity to "continuous potential" algorithm for all types except
    plasma and IFS
  Palette-map <S>ave and Restore (<M>) commands
  <L>ogarithmic and <N>ormal palette-mapping commands and arguments
  Maxiter increased to 32,000 to support log palette maps
  .MAP and .IFS files can now reside anywhere along the DOS path
  Direct-video support for Hercules adapters (Dean Souleles)
  Tandy 1000 160x200x16 mode (Tom Price)
  320x400x256 register-compatible-VGA "tweaked" mode
  ATI VGA Wonder 1024x768x16 direct-video mode (Mark Peterson)
  1024x768x16 direct-video mode for all supported chipsets
  Tseng 640x400x256 mode
  "Roll-your-own" video mode 19
  New video-table "hot-keys" eliminate need for enhanced keyboard to access
    later entries


Version 9.3, 8/89

  <P>rint command and "PRINTER=" argument (Matt Saucier)
  8514/A video modes (Kyle Powell)
  SSTOOLS.INI sensitivity and '@THISFILE' argument
  Continuous-potential algorithm for Mandelbrot/Julia sets
  Light source 3D option for all fractal types
  "Distance estimator" M/J method (Phil Wilson) implemented as "test" type
  LambdaCosine and LambdaExponent types
  Color cycling mode for 640x350x16 EGA adapters
  Plasma clouds for 16-color and 4-color video modes
  Improved TARGA support (Joe McLain)
  CGA modes now use direct-video read/writes
  Tandy 1000 320x200x16 and 640x200x4 modes (Tom Price)
  TRIDENT chip-set super-VGA video modes (Lew Ramsey)
  Direct-access video modes for TRIDENT, Chips & Technologies, and ATI VGA
    WONDER adapters (John Bridges). and, unlike version 9.1, they WORK in
    version 9.3!
  "zoom-out" (<Ctrl><Enter>) command
  <D>os command for shelling out
  2/4/16-color Disk/RAM video mode capability and 2-color video modes
    supporting full-page printer graphics
  "INSIDE=-1" option (treated dynamically as "INSIDE=maxiter")
  Improved <H>elp and sound routines (even a "SOUND=off" argument)
  Turbo-C and TASM compatibility (really!  Would we lie to you?)


Version 8.1, 6/89

  <3>D restore-from-disk and 3D <O>verlay commands, "3D=" argument
```

    Fast Newton algorithm including inversion option (Lee Crocker)
    16-bit Mandelbrot/Julia logic for 386-class speed with non-386 PCs on
      "large" images (Mark Peterson)
    Restore now loads .GIF files (as plasma clouds)
    TARGA video modes and color-map file options (Joe McLain)
    30 new color-cycling palette options (<Shft><F1> to <Alt><F10>)
    "Disk-video, RAM-video, EMS-video" modes
    Lambda sets now use integer math (with 80386 speedups)
    "WARN=yes" argument to prevent over-writing old .GIF files

Version 7.0, 4/89

    Restore from disk (from prior save-to-disk using v. 7.0 or later)
    New types: Newton, Lambda, Mandelfp, Juliafp, Plasma, Lambdasine
    Many new color-cycling options (for VGA adapters only)
    New periodicity logic (Mark Peterson)
    Initial displays recognize (and use) symmetry
    Solid-guessing option (now the default)
    Context-sensitive <H>elp
    Customizable video mode configuration file (FRACTINT.CFG)
    "Batch mode" option
    Improved super-VGA support (with direct video read/writes)
    Non-standard 360 x 480 x 256 color mode on a STANDARD IBM VGA!

Version 6.0, 2/89

    32-bit integer math emulated for non-386 processors; FRACT386 renamed
      FRACTINT
    More video modes

Version 5.1, 1/89

    Save to disk
    New! Improved! (and Incompatible!) optional arguments format
    "Correct" initial image aspect ratio
    More video modes

Version 4.0, 12/88

    Mouse support (Mike Kaufman)
    Dynamic iteration limits
    Color cycling
    Dual-pass mode
    More video modes, including "tweaked" modes for IBM VGA and register-
      compatible adapters

Version 3.1, 11/88

    Julia sets

Version 2.1, 10/23/88 (the "debut" on CIS)

```
  Video table
  CPU type detector


Version 2.0, 10/10/88

  Zoom and pan


Version 1.0, 9/88

  The original, blindingly fast, 386-specific 32-bit integer algorithm
;
;
;
```

1.31   Version13 to 14 Conversion

A number of types in Fractint version 13 and earlier were generalized in
version 14. We added a "backward compatibility" hook that (hopefully)
automatically translates these to the new form when the old files are
read. Files may be converted via:

```
   FRACTINT OLDFILE.FRA SAVENAME=NEWFILE.GIF BATCH=YES
```

In a few cases the biomorph flag was incorrectly set in older files. In
that case, add "biomorph=no" to the command line.

This procedure can also be used to convert any *.fra file to the new
GIF89a spec, which now allows storage of fractal information.

TYPES CHANGED FROM VERSION 13 –


```
V13 NAME               V14 NAME + PARAMETERS
--------               ------------------------------------


LOGMAP=YES             LOGMAP=OLD   for identical Logmap type

DEMJ                   JULIA DISTEST=nnn

DEMM                   MANDEL DISTEST=nnn

                       Note: DISTEST also available on many other types

MANSINEXP              MANFN+EXP FUNCTION=SIN

                       Note: New functions for this type are
                             cos sinh cosh exp log sqr

JULSINEXP              JULFN+EXP FUNCTION=SIN

                       Note: New functions for this type are
```

```
                                cos sinh cosh exp log sqr

MANSINZSQRD             MANFN+ZSQRD FUNCTION=SQR/SIN

                       Note: New functions for this type are
                             cos sinh cosh exp log sqr

JULSINZSQRD            JULFN+ZSQRD FUNCTION=SQR/SIN

                       Note: New functions for this type are
                             cos sinh cosh exp log sqr

LAMBDACOS             LAMBDAFN FUNCTION=COS

LAMBDACOSH            LAMBDAFN FUNCTION=COSH

LAMBDAEXP             LAMBDAFN FUNCTION=EXP

LAMBDASINE           LAMBDAFN FUNCTION=SIN

LAMBDASINH           LAMBDAFN FUNCTION=SINH

                       Note: New functions for this type are
                             log sqr

MANDELCOS             MANDELFN FUNCTION=COS

MANDELCOSH            MANDELFN FUNCTION=COSH

MANDELEXP             MANDELFN FUNCTION=EXP

MANDELSINE           MANDELFN FUNCTION=SIN

MANDELSINH           MANDELFN FUNCTION=SINH

                       Note: New functions for this type are
                             log sqr

MANDELLAMBDA          MANDELLAMBDA INITORBIT=PIXEL

POPCORN SYMMETRY=NONE   POPCORNJUL

-------------------------------------------------------------

Formulas from FRACTINT.FRM in version 13

MANDELGLASS           MANDELLAMBDA INITORBIT=.5/0

INVMANDEL            V13 divide bug may cause some image differences.

NEWTON4             V13 divide bug may cause some image differences.

SPIDER              V13 divide bug may cause some image differences.

MANDELSINE           MANDELFN FUNCTION=SIN BAILOUT=50

MANDELCOSINE         MANDELFN FUNCTION=COS BAILOUT=50
```

```
MANDELHYPSINE              MANDELFN FUNCTION=SINH BAILOUT=50

MANDELHYPCOSINE            MANDELFN FUNCTION=COSH BAILOUT=50

SCOTTSIN PARAMS=nnn        FN+FN FUNCTION=SIN/SQR BAILOUT=nnn+3

SCOTTSINH PARAMS=nnn       FN+FN FUNCTION=SINH/SQR BAILOUT=nnn+3

SCOTTCOS PARAMS=nnn        FN+FN FUNCTION=COS/SQR BAILOUT=nnn+3

SCOTTCOSH PARAMS=nnn       FN+FN FUNCTION=COSH/SQR BAILOUT=nnn+3

SCOTTLPC PARAMS=nnn        FN+FN FUNCTION=LOG/COS BAILOUT=nnn+3

SCOTTLPS PARAMS=nnn        FN+FN FUNCTION=LOG/SIN BAILOUT=nnn+3
                           Note: New functions for this type are
                           sin/sin sin/cos sin/sinh sin/cosh sin/exp
                           cos/cos cos/sinh cos/cosh cos/exp
                           sinh/sinh sinh/cosh sinh/exp sinh/log
                           cosh/cosh cosh/exp cosh/log
                           exp/exp exp/log exp/sqr log/log log/sqr sqr/sqr

SCOTTSZSA PARAMS=nnn       FN(Z*Z) FUNCTION=SIN BAILOUT=nnn+3

SCOTTCZSA PARAMS=nnn       FN(Z*Z) FUNCTION=COS BAILOUT=nnn+3

                           Note: New functions for this type are
                           sinh cosh exp log sqr

SCOTTZSZZ PARAMS=nnn       FN*Z+Z FUNCTION=SIN BAILOUT=nnn+3

SCOTTZCZZ PARAMS=nnn       FN*Z+Z FUNCTION=COS BAILOUT=nnn+3

                           Note: New functions for this type are
                           sinh cosh exp log sqr

SCOTTSZSB PARAMS=nnn       FN*FN FUNCTION=SIN/SIN BAILOUT=nnn+3

SCOTTCZSB PARAMS=nnn       FN*FN FUNCTION=COS/COS BAILOUT=nnn+3

SCOTTLTS PARAMS=nnn        FN*FN FUNCTION=LOG/SIN BAILOUT=nnn+3

SCOTTLTC PARAMS=nnn        FN*FN FUNCTION=LOG/COS BAILOUT=nnn+3

                           Note: New functions for this type are
                           sin/cos sin/sinh sin/cosh sin/exp sin/sqr
                           cos/sinh cos/cosh cos/exp cos/sqr
                           sinh/sinh sinh/cosh sinh/exp sinh/log sinh/sqr
                           cosh/cosh cosh/exp cosh/log cosh/sqr
                           exp/exp exp/log exp/sqr log/log log/sqr sqr/sqr

SCOTTSIC PARAMS=nnn        SQR(1/FN) FUNCTION=COS BAILOUT=nnn+3

SCOTTSIS PARAMS=nnn        SQR(1/FN) FUNCTION=SIN BAILOUT=nnn+3

TETRATE PARAMS=nnn         TETRATE BAILOUT=nnn+3
```

```
                         Note: New function type sqr(1/fn) with
                               sin cos sinh cosh exp log sqr

                         Note: New function type sqr(fn) with
                               sin cos sinh cosh exp log sqr
;
;
;
; FRACTINT intro screen primary authors
;
 Current Primary Authors
 Bert Tyler        [73477,433] (CompuServe) 73477.433@compuserve.com (Internet)
 Timothy Wegner    [71320,675]              twegner@phoenix.net
 Jonathan Osuch    [73277,1432]             73277.1432@compuserve.com
 Wesley Loewer                              loewer@tenet.edu
 Contributing Authors


; room for 14 authors at a time here
```

```
  SPACEBAR toggles scrolling off/on
    Copyright (C) 1990-95 The Stone Soup Group.  Fractint may be freely copied
    and distributed but may not be sold.  See help for more information.
;
;
;
;
; FRACTINT intro screen contributing authors.
;
                    ...
 Michael Abrash    360x480x256, 320x400x256 VGA video modes
 Joseph Albrecht   Tandy video, CGA video speedup
 Kevin Allen       kevina@microsoft.com Finite attractor, bifurcation engine
 Steve Bennett     restore-from-disk logic
 Rob Beyer         [71021,2074] Barnsley IFS, Lorenz fractals
 Francois Blais    [70700,446] Lyapunov Fractals, LYAPUNOV.MAP
 Dennis Bragg      [75300,2456] DXF Raytracing output option
 Pieter Branderhorst Past main author, solid guessing, menus
 Juan J. Buhler    jbuhler@gidef.edu.ar Diffusion options, inverse Julia type
 Mike Burkey       376x564x256, 400x564x256, and 832x612x256 VGA video modes
 Robin Bussell     Palette-editor "freestyle" option, "browser" feature
 John Bridges      [75300,2137] superVGA support, 360x480x256 mode
 Brian Corbino     [71611,702] Tandy 1000 640x200x16 video mode
 Lee Crocker       lcrocker@netcom.com Fast Newton, Inversion, Decomposition..
 Monte Davis       [71450,3542] Documentation
```

```
Paul De Leeuw      RDS (Random Dot Stereogram) Feature
Chuck Ebbert       [76306,1226] cmprsd & sqrt logmap, fpu speedups, fast parser
Dan Farmer         [74431,1075] orbits enhancements
Richard Finegold   [76701,153] 8/16/../256-Way Decomposition option
Frank Fussenegger  Mandelbrot speedups
Mike Gelvin        [73337,520] Mandelbrot speedups
Lawrence Gozum     [73437,2372] Tseng 640x400x256 Video Mode
David Guenther     [70531,3525] Boundary Tracing algorithm
Norman Hills       [71621,1352] Ranges option
Richard Hughes     [70461,3272] "inside=", "outside=" coloring options
Mike Kaufman       [kaufman@eecs.nwu.edu] mouse support, other features
Adrian Mariano     [adrian@u.washington.edu] Diffusion & L-Systems
Charles Marslett   [75300,1636] VESA video and IIT math chip support
Joe McLain         [75066,1257] TARGA Support, color-map files
Bob Montgomery     [73357,3140] (Author of VPIC) Fast text I/O routines
Bret Mulvey        plasma clouds
Roy Murphy         [76376,721] Lyapunov Fractals
Ethan Nagel        [70022,2552] Palette editor, integrated help/doc system
Yavuz Onder        yavuz@bnr.ca Postscript printer driver
Mark Peterson      [73642,1775] Past main author, parser, julibrot
Kyle Powell        [76704,12] 8514/A Support
Marc Reinig        [72410,77] Lots of 3D options
Matt Saucier       [72371,3101] Printer Support
Herb Savage        [75260,217] 'inside=bof60', 'inside=bof61' options
Ken Shirriff       shirriff@sprite.berkeley.edu Quaternions, CA, Xfract port
Lee Skinner        [75450,3631] Tetrate fractal types and more
Michael Snyder     [75300,642] julia inverse and Julia-In-A-Window using MIIM
Dean Souleles      [75115,1671] Hercules Support
Kurt Sowa          [73467,2013] Color Printer Support
Hugh Steele        cyclerange feature
John Swenson       [75300,2136] Postscript printer features
Chris Taylor       Floating&Fixed-point algorithm speedups, Tesseral Option
Scott Taylor       [72401,410] PostScript, Kam Torus, many fn types.
Bill Townsend      Mandelbrot Speedups
Paul Varner        [73237,441] Extended Memory support for Disk Video
Dave Warker        Integer Mandelbrot Fractals concept
Aaron Williams     Register-compatible 8514/A code
Phil Wilson        [76247,3145] Distance Estimator, Bifurcation fractals
Nicholas Wilt      Lsystem speedups
Richard Wilton     Tweaked VGA Video modes
; Byte Magazine    Tweaked VGA Modes
; MS-Kermit        Keyboard Routines
; PC Magazine      Sound Routines
; PC Tech Journal  CPU, FPU Detectors
```