Fractint for Windows (Winfract)

Note: This is a "public-beta" release of Winfract. Read the New Features in this Version help topic for details, including how and where to send bug reports.

Help Topics

New Features in this Version

WinFract Version **17.2x** is a 'Public-Beta' release and a major upgrade to the previous general release, Winfract Version 17.1. Whats new to the program is a great deal of core technology from the DOS version of Fractint, meaning better performance, an improved user interface, and many new options that were not in the older versions of Winfract. Winfract will be distributed on diskette as part of the book "Fractals for Windows", ISBN 1-878739-25-5, to be published in December 1992 by Waite Group Press. Because it takes several months to print, bind, and distribute the book, we are releasing this interim version of Winfract now (and perhaps another interim release later - this is very much a "work-in-progress") so folks can find and we can fix any bugs before the disk goes in the book. We've discovered over the years that users are far better at uncovering programming bugs than programmers are, so **please help us** by reporting any bugs you run into. Send your bug reports to Bert Tyler (CompuServe ID 73477,433 and available via the Internet as 73477.433@compuserve.com).

New Features Added in this 17.30 "Public Beta" Release:

The Dialog box logic has been reworked internally to act more like Fractint, in that the internal fractal engine returns to its main calling loop before the Windows dialog boxes are invoked. This is much easier on the program's stack space.

Various and sundry **bug fixes**, including elimination of the "double/phantom zoom-boxes" problem that appeared in version 17.25.

New Features Added in the 17.27 "Public Beta" Release:

Various and sundry **bug fixes**, including fixing the inability to display large bitmaps with some Windows video drivers, running out of environment space on those folks with a lot of environment variables, fixes to the Mandelbrot/Julia toggle logic, forgetting the location of the current directory when performing a File Load after the first File Save, GP faults using logarithmic palettes in combination with other options, and crashing when zooming deeply into the Mandelbrot/Julia images on PCs without an FPU. Also, I *think* I've finally managed to type in my CompuServe ID right in the previous paragraphs <embarrassed grin>.

New Features Added in the 17.26 "Public Beta" Release:

New "**Reset All Options**" menu item in the **Fractals** menu that resets all fractal-related items to their initial defaults.

If your " hotkey actions" option is set to Fractint-style prompts, pressing the **TAB** key now brings up the Fractint-style status screen.

Image updating is a bit more intelligent (and a bit more frequent, with large images).

Various and sundry **bug fixes**.

New Features Added in the 17.25 "Public Beta" Release:

Many, many **bug fixes** - loading @Parameter entries actually works now (would we lie to you?)

Zooming has been reworked - the old "non-zoom-bar" method of zooming has been replaced with new **Zoom In** and **Zoom Out** options in the "View" menulist. See the section on Zooming in on an Image for details.

The menu structure has been completely revised into five menu items:

File contains file, printer, and clipboard-related items as well as the traditional "Exit".
Fractal contains all of the fractal-related options, including new **3D Params** and **Starfield** options.
View contains all of the non-fractal-related items, including new **Zoom-Box** and **Pixel-By-Pixel Update** options.
Colors contains all color-related options, including loading and saving of colormap files and color-cycling.
Help contains all help-related items, including both the Winfract-specific and Fractint-specific help systems.

There's a new, experimental, undocumented-anywhere-else, and maybe never-to-be-seen-again "orbits" toggle. When the **Pixel-By-Pixel Update** option is active (and only then), pressing the **"*"** key either above the "8" or on the numeric keypad) toggles on and off the display of orbits, "Fractint-style". We're still debating the merits of this option, as it really slows down the Windows display.

New Features Added in the 17.2 "Public Beta Release:

Many of the new features are discussed in more detail in the next section ("Fractint-Style HelpScreens and Prompts"), but in summary:

- Fractint's hypertext-like online help (all 350+ screens' worth!) is now available at the push of a button (specifically, the Control-F1 button - the F1 button brings up this Winfract-specific help system).

- Many Fractint-style hot-keys have been added. Depending on the status of the "Hot-key Actions" option (found in the "Options" menuitem), these hot-keys will either take you directly to the appropriate Winfract menu or take you right to the next feature...

- Many of Fractint's prompting screens have been incorporated into Winfract. Note that some of these screens list options that aren't connected to anything in the Winfract environment (the sound options on the 'X' screen come to mind). However, those of you who are used to Fractint's prompting screens may find this option more familiar than the Windows menus.

- Finally, there are a number of new menu-items on the Windows menus, and many of the selection screens have new options to choose from. The old "Options and Doodads" selection screen has been expanded to three screens corresponding to their Equivalents in Fractint: "Basic Options", "Extended Options", and "Fractal Parameters". The 3D and 3D Overlay screens now let you specify raytracing output. Finally, there are brand-new screens in the Files options for the loading and saving of Parameter ("Batch") files.

Fractint-Style Help and Prompts

Fractint's hypertext-like online help (all 350+ screens' worth!) is available at the push of a button (specifically, the Control-F1 button - the F1 button brings up this Winfract-specific help system). Note that Fractint's online help is just that - Fractint-specific online help. While much of that help is common to both Fractint and Winfract, some of it (the sections on printing, the palette editor, and the 'V(iew)' options come to mind right away) describe items that have nothing to do with the Winfract environment. Also, Fractint's online help is not context-sensitive at the moment - you always start at Fractint's initial help index screen when you press Control-F1.

Many Fractint-style hot-keys are also available. Depending on the status of the "Hot-key Actions" option (found in the "Options" menuitem), these hot-keys will either take you directly to the appropriate Winfract menu or to a text-based prompting screen directly from Fractint. Note that some functions (printing and selecting video modes are prime examples) are so different between the DOS and Windows environments that you will be taken to a Windows-style menu even if you've selected Fractint-style prompts.

The Fractint-style "Hot-Keys" and their actions are listed below. Items marked with a "*" are not connected to a Fractint-style prompting screen and will always take you directly to a Windows-style equivalent.

T	Select Fractal Type
S	Save Fractal Image *
P	Print Fractal Image *
R	Restore Fractal Image
3	3D Restore
O	3D Overlay
X	Basic Options
Y	Extended Doodads
Z	Fractal Parameters
I	3D Parameters
A	Starfields
@	Select Parameter file and item
B	Save Parameter item
L	Load Color-Map
W	Write (save) Color-Map *
DEL	Select Image Size *
INS	Reset All Options
TAB	Status display

In addition to the above, Winfract also responds to many of the Fractint-style color-cycling hot-keys: 'C', '+', '-', '<', '>', the four cursor keys, SPACE, and ENTER. Unlike Fractint, you don't have to enter "color-cycling mode" to use these keys - Color-Cycling mode and the main image mode are one and the same.

Many of Fractint's prompting screens have been also incorporated into Winfract. Note that some of these screens list options that aren't connected to anything in the Winfract environment (the sound options on the 'X' screen come to mind). However, those of you who are used to Fractint's prompting screens may find this option more comfortable than the Windows menus. Currently, these Fractint-style prompting screens are only selectable using the Hot-Keys (using the Windows menu-bars always gives you Windows-style entry screens), and then only if you have chosen "Fractint-style Prompts" as your "Hot-Key Options" default.

Limitations in Winfract

Fractint for Windows is (and will continue to be) a "port" of Fractint for DOS to the Windows environment, retaining the Fractint for DOS fractal and GIF engines, but replacing its front-end and graphics interface with a Windows engine. As such, its fractal engine will always "trail" that of Fractint for DOS (although hopefully not by much - the two 17.1 versions were uploaded within hours of each other), and the user interface will tend to do the same except where the Windows version adds functionality simply by virtue of the Windows interface.

There are three main causes behind the various bugs and limitations in this program. For future brevity, these causes are tagged FFD (problems related to the fact that much of the code is and will continue to be from Fractint For Dos), WLIM (limitations caused by Windows, or some Windows video/printer drivers that aren't limitations under MS-DOS), and NYI (Not Yet Implemented - hey, we're working on it!). We used to hide behind a fourth excuse - calling ourselves novice Windows programmers and using the "what can you expect?" claim - but that wore a little thin. FFD and WLIM limitations are probably permanent - hopefully NYI problems are less so.

At any rate, the major limitations of Winfract continue to be:

- You can run only one "instance" of Fractint for Windows. FFD: Fractint for DOS is riddled with initialized FAR data ("char far myvalue = 0;"), and the Windows SDK silently but firmly tags any program containing code like that as a single-instance program.

- Palette support is limited. WLIM: The "stock" VGA driver supplied with Win 3.0 doesn't support palette-modification by applications at all (most third-party 256-color SuperVGA drivers do, though). "Plasma Clouds" look Godawful using anything less than a 256-color Windows driver. Color-cycling is limited to video drivers which support palette-manipulation (and, for 16-color drivers, color-cycling affects the background windows as well). Note that 16-bit and 24-bit "true color" Windows video drivers don't *have* a palette to manipulate, so we can't color-cycle them.

- Fractint for Windows is not as "background process" friendly as it should be when it updates the screen image - if it is updating a large image in its entirety, it can grab the machine for seconds at a time. FFD and WLIM - Windows only gives other programs a shot at the CPU when the currently active program volunteers to give up control, and the fractal engine that Winfract is based on only gives us the opportunity to do that when it calls the routine that checks the keyboard - an event that may happen only once a second or so.

File Menu

File **Open** loads either Fractint-for-DOS-style or "generic" GIF files. Note that the correct palette isn't completely displayed on the screen. Windows "reserves" the first twenty palettes for its own use, and "adjusts" Fractint's images accordingly - and for 16-color drivers like the VGA driver supplied with Windows that don't support palette-manipulation by applications programs, cheerfully ignores our attempts at palette-manipulation. If the restored image is resumable , the image will resume calculating as soon as it is loaded.

File **Save** saves your current image (by default, as a GIF (version 89a) file, compatible with Fractint-for-DOS, your favorite DOS GIF viewers, and (of course) the "File Open" option, but optionally as a GIF87a or a BMP file). Again, the palette you see on the screen may not be the one that the program is "using" (and gets saved to the disk file). If the saved image is resumable , the image will resume calculating as soon as it is finished saving. Note that Winfract can only open GIF89a-format images as fractal images, and cannot open BMP-format images at all.

File **3D Open** and File **3D Overlay** load in your image using "3D" transformations. The 3D Open option clears your image first and then loads the new one, while the Overlay option leaves your original image intact and adds the new image over it.

File **Print** does the best job it can sending the image to your printer. The program does not "dither" for black-and-white printers - it just "candy stripes" adjacent colors the same way that Fractint-for-DOS does.

Read Batchfile and **Write Batchfile** load and save "batch-file" parameters of your (and others) favorite images. As soon as you Read a a batchfile image (saved as an entry in a PAR file such as FRACTINT.PAR), Winfract will begin generating that image at whatever image size you are currently using. The disadvantage to saving your favorite images as PARameter files rather than GIF images is that Winfract has to recreate each image every time you select it. The advantage is that you can fit a lot more formulas on your hard disk this way, and they're a tad easier to E-mail to your friends!

File **Copy to Clipboard** copies the currently-displayed image to the Windows clipboard (in "Device-Independent Bitmap" format) where your other Windows programs can collect it. Note that, for 256-color video drivers and 256-color images, the colors in the clipboard won't visibly match the colors in Winfract's window until you change your "focus" to the clipboard's window. That's because Winfract has warned the Clipboard that it is reserving the right to color-cycle its colors, so the Clipboard is just using Windows' default 20 colors to display its colors while Winfract has control of the screen.

Fractals Options

Fractal Formula brings up a dialog box letting you select from any of approximately 80 fractal types. See the Fractal Formula Selection for formula details.

Basic Options brings up a "Basic Options" menu that lets you select any of what we consider the "basic" (mostly because we added them first) fractal options - various generation algorithms (single/dual pass, solid-guessing, boundary tracing, tesseral), inside and outside coloring schemes, maximum iterations, Biomorph and Decomposition options, etc .

Extended Options brings up an "Extended Options" menu that lets you select any of what we consider to be "extended" (because we didn't think of them until later) fractal options, including Inversion and Continuous Potential.

Fractal Params brings up a "Fractal Parameters" menu that lets you select any modifiable parameters in your fractal type as well as the specific area you may want to zoom in on.

3D Params brings up a "3D Parameters" menu that lets you select any modifiable 3D parameters in your fractal type.

Reset all Options resets all fractal-related options to the default state they were in when Winfract was first started.

Once you have selected any of these options, Winfract will clear the image and regenerate it using the new options you have given it.

View Options

Image Settings The **Image Settings** option lets you select either a pre-defined image size, or make up your own (up to an internal limit of 2048x2048) and select a two-color, 16-color, or 256-color image. Those of you with 24-bit color video cards will have to hold yourself back to 256-color images for the moment - the 256-color limit is one of the penalties of using the fractal engine from Fractint.

If your image size is larger than your window, you can use the scroll bars in the window to scroll around your image.

If you have selected the "Window Sizing" option (see below), your image window will automatically re-size to the image size you have just selected.

Zoom In, **Zoom Out**, and **Zoom Bar** toggles select which of three zooming options you want to use. For details, see the help section on Zooming in on an Image.

Coordinate Box toggles the display of an optional "Coordinate Box" displaying information about the pixel directly underneath your mouse pointer. See the help section under Coordinate Box for details.

Window Sizing toggles the "Window Sizing" option, in which the scroll-bars are eliminated and the image window is automatically sized to the size of your fractal image.

Pixel-By-Pixel Update toggles the "Pixel-By-Pixel Update" option, in which Winfract's normal method of updating its image in periodic chunks is changed to a "display every pixel as it is changed" strategy. This approach is *extremely* slow (Windows' overhead takes a heavy toll on programs that update their images this way), but it lets you see how Winfract is actually generating its images - try this using the Boundary Trace and Tesseral image-generation methods.

HotKey Actions toggles whether the various "hot-keys" Winfract is sensitive to take you to the same Windows-style menus accessed through the Windows menubar or "Fractint-style" prompting screens. See the help section on Fractint-Style Help and Prompts for details.

Status Brings up a Status messagebox displaying the status of your current image (which formula you are using, its parameters and screen corners, and whether or not it's still being generated or has finished). If the image is still being generated, Winfract continues with it when you press the "OK" button.

Color Options

Load Color-Map and **Write Color-Map** load and save external color-maps in the same manner as Fractint-for-DOS (within Windows' limitations).

Color-Cycling brings up a Color_Cycling Dialog box. See the Help section on Color_Cycling for details.

Help Options

Index brings up this help system. So does pressing the F1 key.

Using Help brings up the standard Windows "help on Help" system.

Fractint Help brings up Fractint-style help screens (all 350+ of them!) So does pressing the Control-F1 key. See the Help section on Fractint-Style Help and Prompts for details.

About Winfract brings up a standard "About Box".

Fractal Formula Selection

Using the Options **Formula** option, you can select any of over 80 fractal types (every fractal type that is available in version 17.0 of Fractint-for-DOS). After selecting a fractal type, a dialogue box pops up and prompts you for any formula parameters and the screen corners (all with reasonable default values).

A brief list of Fractal types (and their formulas) included in this release:

barnsleyj1

z(0) = pixel;
z(n+1) = (z-1)*c if real(z) >= 0, else
z(n+1) = (z+1)*modulus(c)/c
Two parameters: real and imaginary parts of c

barnsleyj2

z(0) = pixel;
if real(z(n)) * imag(c) + real(c) * imag(z((n)) >= 0
z(n+1) = (z(n)-1)*c
else
z(n+1) = (z(n)+1)*c
Two parameters: real and imaginary parts of c

barnsleyj3

z(0) = pixel;
if real(z(n) > 0 then z(n+1) = (real(z(n))^2 - imag(z(n))^2 - 1)
+ i * (2*real(z((n)) * imag(z((n))) else
z(n+1) = (real(z(n))^2 - imag(z(n))^2 - 1 + real(c) * real(z(n))
+ i * (2*real(z((n)) * imag(z((n)) + imag(c) * real(z(n))
Two parameters: real and imaginary parts of c.

barnsleym1

z(0) = c = pixel;
if real(z) >= 0 then
z(n+1) = (z-1)*c
else
z(n+1) = (z+1)*modulus(c)/c.
Parameters are perturbations of z(0)

barnsleym2

z(0) = c = pixel;
if real(z)*imag(c) + real(c)*imag(z) >= 0
z(n+1) = (z-1)*c
else
z(n+1) = (z+1)*c
Parameters are perturbations of z(0)

barnsleym3

z(0) = c = pixel;
if real(z(n) > 0 then z(n+1) = (real(z(n))^2 - imag(z(n))^2 - 1)
+ i * (2*real(z((n)) * imag(z((n))) else
z(n+1) = (real(z(n))^2 - imag(z(n))^2 - 1 + real(c) * real(z(n))
+ i * (2*real(z((n)) * imag(z((n)) + imag(c) * real(z(n))
Parameters are pertubations of z(0)

bifurcation

Pictoral representation of a population growth model.
Let P = new population, p = oldpopulation, r = growth rate
The model is: P = p + r*p*(1-p).
Two parameters: Filter Cycles and Seed Population.

bif+sinpi
Bifurcation variation: model is: P = p + r*sin(PI*p).
Two parameters: Filter Cycles and Seed Population.

bif=sinpi
Bifurcation variation: model is: P = r*sin(PI*p).
Two parameters: Filter Cycles and Seed Population.

biflambda
Bifurcation variation: model is: P = r*p*(1-p)P.
Two parameters: Filter Cycles and Seed Population.

bifstewart
Bifurcation variation: model is: P = (r*p*p) - 1.
Two parameters: Filter Cycles and Seed Population.

Circle
Circle pattern by John Connett
x + iy = pixel
z = a*(x^2 + y^2)
c = integer part of z
color = c modulo(number of colors)

cmplxmarksjul
A generalization of the marksjulia fractal.
z(0) = pixel;
z(n+1) = (c^exp)*z(n) + c.
Four parameters: real and imaginary parts of c and exp.

cmplxmarksmand
A generalization of the marksmandel fractal.
z(0) = c = pixel;
z(n+1) = (c^exp)*z(n) + c.
Four parameters: real and imaginary parts of
perturbation of z(0) and exp.

complexnewton, complexbasin
Newton fractal types extended to complex degrees. Complexnewton colors pixels according to the number of iterations required to escape to a root. Complexbasin colors pixels according to which root captures the orbit. The equation is based on the newton formula for solving the equation z^p = r
z(0) = pixel;
z(n+1) = ((p - 1) * z(n)^p + r)/(p * z(n)^(p - 1)).
Four parameters: real & imaginary parts of degree p and root r

diffusion
Diffusion Limited Aggregation. Randomly moving points accumulate. One parameter: border width (default 10)

fn+fn(pix)

c = z(0) = pixel;
z(n+1) = fn1(z) + p*fn2(c)
Six parameters: real and imaginary parts of the perturbation of z(0) and factor p, and the functions fn1, and fn2.

fn(z*z)
z(0) = pixel;
z(n+1) = fn(z(n)*z(n))
One parameter: the function fn.

fn*fn
z(0) = pixel; z(n+1) = fn1(n)*fn2(n)
Two parameters: the functions fn1 and fn2.

fn*z+z
z(0) = pixel; z(n+1) = p1*fn(z(n))*z(n) + p2*z(n)
Six parameters: the real and imaginary components of p1 and p2, and the functions fn1 and fn2.

fn+fn
z(0) = pixel;
z(n+1) = p1*fn1(z(n))+p2*fn2(z(n))
Six parameters: The real and imaginary components of p1 and p2, and the functions fn1 and fn2.

formula
Formula interpreter - write your own formulas as text files!

gingerbread
Orbit in two dimensions defined by:
x(n+1) = 1 - y(n) + |x(n)|
y(n+1) = x(n)
Two parameters: initial values of x(0) and y(0).

henon
Orbit in two dimensions defined by:
x(n+1) = 1 + y(n) - a*x(n)*x(n)
y(n+1) = b*x(n)
Two parameters: a and b

Hopalong
Hopalong attractor by Barry Martin - orbit in two dimensions.
z(0) = y(0) = 0;
x(n+1) = y(n) - sign(x(n))*sqrt(abs(b*x(n)-c))
y(n+1) = a - x(n)
Parameters are a, b, and c.

IFS
Barnsley IFS fractals.

julfn+exp
A generalized Clifford Pickover fractal.
z(0) = pixel;
z(n+1) = fn(z(n)) + e^z(n) + c.
Three parameters: real & imaginary parts of c, and fn

julfn+zsqrd
z(0) = pixel;
z(n+1) = fn(z(n)) + z(n)^2 + c
Three parameters: real & imaginary parts of c, and fn

julia
Classic Julia set fractal.
z(0) = pixel; z(n+1) = z(n)^2 + c.
Two parameters: real and imaginary parts of c.

julia4
Fourth-power Julia set fractals, a special case
of julzpower kept for speed.
z(0) = pixel;
z(n+1) = z(n)^4 + c.
Two parameters: real and imaginary parts of c.

julibrot
'Julibrot' 4-dimensional fractals.

julzpower
z(0) = pixel;
z(n+1) = z(n)^m + c.
Three parameters: real & imaginary parts of c, exponent m

julzzpwr
z(0) = pixel;
z(n+1) = z(n)^z(n) + z(n)^m + c.
Three parameters: real & imaginary parts of c, exponent m

kamtorus, kamtorus3d
Series of orbits superimposed.
3d version has 'orbit' the z dimension.
x(0) = y(0) = orbit/3;
x(n+1) = x(n)*cos(a) + (x(n)*x(n)-y(n))*sin(a)
y(n+1) = x(n)*sin(a) - (x(n)*x(n)-y(n))*cos(a)
After each orbit, 'orbit' is incremented by a step size.
Parameters: a, step size, stop value for 'orbit', and
points per orbit.

lambda
Classic Lambda fractal. 'Julia' variant of Mandellambda.
z(0) = pixel;
z(n+1) = lambda*z(n)*(1 - z(n)^2).
Two parameters: real and imaginary parts of lambda.

lambdafn
z(0) = pixel;
z(n+1) = lambda * fn(z(n)).
Three parameters: real, imag portions of lambda, and fn

lorenz, lorenz3d
Lorenz two lobe attractor - orbit in three dimensions.
In 2d the x and y components are projected to form the image.
z(0) = y(0) = z(0) = 1;
x(n+1) = x(n) + (-a*x(n)*dt) + (a*y(n)*dt)

y(n+1) = y(n) + (b*x(n)*dt) - (y(n)*dt) - (z(n)*x(n)*dt)
z(n+1) = z(n) + (-c*z(n)*dt) + (x(n)*y(n)*dt)
Parameters are dt, a, b, and c.

lorenz3d1

Lorenz one lobe attractor - orbit in three dimensions.
The original formulas were developed by Rick Miranda and Emily Stone.
z(0) = y(0) = z(0) = 1; norm = sqrt(x(n)^2 + y(n)^2)
x(n+1) = x(n) + (-a*dt-dt)*x(n) + (a*dt-b*dt)*y(n)
+ (dt-a*dt)*norm + y(n)*dt*z(n)
y(n+1) = y(n) + (b*dt-a*dt)*x(n) - (a*dt+dt)*y(n)
+ (b*dt+a*dt)*norm - x(n)*dt*z(n) - norm*z(n)*dt
z(n+1) = z(n) +(y(n)*dt/2) - c*dt*z(n)
Parameters are dt, a, b, and c.

lorenz3d3

Lorenz three lobe attractor - orbit in three dimensions.
The original formulas were developed by Rick Miranda and Emily Stone.
z(0) = y(0) = z(0) = 1; norm = sqrt(x(n)^2 + y(n)^2)
x(n+1) = x(n) +(-(a*dt+dt)*x(n) + (a*dt-b*dt+z(n)*dt)*y(n))/3
+ ((dt-a*dt)*(x(n)^2-y(n)^2)
+ 2*(b*dt+a*dt-z(n)*dt)*x(n)*y(n))/(3*norm)
y(n+1) = y(n) +((b*dt-a*dt-z(n)*dt)*x(n) - (a*dt+dt)*y(n))/3
+ (2*(a*dt-dt)*x(n)*y(n)
+ (b*dt+a*dt-z(n)*dt)*(x(n)^2-y(n)^2))/(3*norm)
z(n+1) = z(n) +(3*x(n)*dt*x(n)*y(n)-y(n)*dt*y(n)^2)/2 - c*dt*z(n)
Parameters are dt, a, b, and c.

lorenz3d4

Lorenz four lobe attractor - orbit in three dimensions.
The original formulas were developed by Rick Miranda and Emily Stone.
z(0) = y(0) = z(0) = 1;
x(n+1) = x(n) +(-a*dt*x(n)^3
+ (2*a*dt+b*dt-z(n)*dt)*x(n)^2*y(n) + (a*dt-2*dt)*x(n)*y(n)^2
+ (z(n)*dt-b*dt)*y(n)^3) / (2 * (x(n)^2+y(n)^2))
y(n+1) = y(n) +((b*dt-z(n)*dt)*x(n)^3 + (a*dt-2*dt)*x(n)^2*y(n)
+ (-2*a*dt-b*dt+z(n)*dt)*x(n)*y(n)^2
- a*dt*y(n)^3) / (2 * (x(n)^2+y(n)^2))
z(n+1) = z(n) +(2*x(n)*dt*x(n)^2*y(n) - 2*x(n)*dt*y(n)^3 - c*dt*z(n))
Parameters are dt, a, b, and c.

lsystem

Using a turtle-graphics control language and starting with
an initial axiom string, carries out string substitutions the
specified number of times (the order), and plots the resulting.

lyapunov

Derived from the Bifurcation fractal, the Lyapunov plots the Lyapunov
Exponent for a population model where the Growth parameter varies between
two values in a periodic manner.

magnet1j

z(0) = pixel;

$$z(n+1) = \left| \frac{z(n)^2 + (c-1)}{2*z(n) + (c-2)} \right|^2$$

Parameters: the real and imaginary parts of c

magnet1m

z(0) = 0; c = pixel;

$$z(n+1) = \left| \frac{z(n)^2 + (c-1)}{2z(n) + (c-2)} \right|^2$$

Parameters: the real & imaginary parts of perturbation of z(0)

magnet2j

z(0) = pixel;

$$z(n+1) = \left| \frac{z(n)^3 + 3(C-1)z(n) + (C-1)(C-2)}{3(z(n)^2) + 3(C-2)z(n) + (C-1)(C-2) - 1} \right|^2$$

Parameters: the real and imaginary parts of c

magnet2m

z(0) = 0; c = pixel;

$$z(n+1) = \left| \frac{z(n)^3 + 3(C-1)z(n) + (C-1)(C-2)}{3(z(n)^2) + 3(C-2)z(n) + (C-1)(C-2) - 1} \right|^2$$

Parameters: the real and imaginary parts of perturbation of z(0)

mandel

Classic Mandelbrot set fractal.

z(0) = c = pixel;

z(n+1) = z(n)^2 + c.

Two parameters: real & imaginary perturbations of z(0)

mandel4

Special case of mandelzpower kept for speed.

z(0) = c = pixel;

z(n+1) = z(n)^4 + c.

Parameters: real & imaginary perturbations of z(0)

mandelfn

z(0) = c = pixel;

z(n+1) = c*fn(z(n)).

Parameters: real & imaginary perturbations of z(0), and fn

Martin

Attractor fractal by Barry Martin - orbit in two dimensions.

z(0) = y(0) = 0;

x(n+1) = y(n) - sin(x(n))

y(n+1) = a - x(n)

Parameter is a (try a value near pi)

mandellambda

z(0) = .5; lambda = pixel;

z(n+1) = lambda*z(n)*(1 - z(n)^2).

Parameters: real & imaginary perturbations of z(0)

manfn+exp

'Mandelbrot-Equivalent' for the julfn+exp fractal.

z(0) = c = pixel;

z(n+1) = fn(z(n)) + e^z(n) + C.

Parameters: real & imaginary perturbations of z(0), and fn

manfn+zsqrd
'Mandelbrot-Equivalent' for the Julfn+zsqrd fractal.
z(0) = c = pixel;
z(n+1) = fn(z(n)) + z(n)^2 + c.
Parameters: real & imaginary perturbations of z(0), and fn

manowar
c = z1(0) = z(0) = pixel;
z(n+1) = z(n)^2 + z1(n) + c;
z1(n+1) = z(n);
Parameters: real & imaginary perturbations of z(0)

manowarj
z1(0) = z(0) = pixel;
z(n+1) = z(n)^2 + z1(n) + c;
z1(n+1) = z(n);
Parameters: real & imaginary perturbations of z(0)

manzpower
'Mandelbrot-Equivalent' for julzpower.
z(0) = c = pixel;
z(n+1) = z(n)^exp + c; try exp = e = 2.71828...
Parameters: real & imaginary perturbations of z(0), real & imaginary parts of exponent exp.

manzzpwr
'Mandelbrot-Equivalent' for the julzzpwr fractal.
z(0) = c = pixel
z(n+1) = z(n)^z(n) + z(n)^exp + C.
Parameters: real & imaginary perturbations of z(0), and exponent

marksjulia
A variant of the julia-lambda fractal.
z(0) = pixel;
z(n+1) = (c^exp)*z(n) + c.
Parameters: real & imaginary parts of c, and exponent

marksmandel
A variant of the mandel-lambda fractal.
z(0) = c = pixel;
z(n+1) = (c^exp)*z(n) + c.
Parameters: real & imaginary perturbations of z(0), and exponent

marksmandelpwr
The marksmandelpwr formula type generalized (it previously had fn=sqr hard coded).
z(0) = pixel, c = z(0) ^ (z(0) - 1):
z(n+1) = c * fn(z(n)) + pixel,
Parameters: real and imaginary pertubations of z(0), and fn

newtbasin
Based on the Newton formula for finding the roots of z^p - 1.
Pixels are colored according to which root captures the orbit.
z(0) = pixel;

z(n+1) = ((p-1)*z(n)^p + 1)/(p*z(n)^(p - 1)).
Two parameters: the polynomial degree p, and a flag to turn on color stripes to show alternate iterations.

newton
Based on the Newton formula for finding the roots of z^p - 1. Pixels are colored according to the iteration when the orbit is captured by a root.
z(0) = pixel;
z(n+1) = ((p-1)*z(n)^p + 1)/(p*z(n)^(p - 1)).
One parameter: the polynomial degree p.

pickover
Orbit in three dimensions defined by:
x(n+1) = sin(a*y(n)) - z(n)*cos(b*x(n))
y(n+1) = z(n)*sin(c*x(n)) - cos(d*y(n))
z(n+1) = sin(x(n))
Parameters: a, b, c, and d.

plasma
Random, cloud-like formations. Requires 4 or more colors. A recursive algorithm repeatedly subdivides the screen and colors pixels according to an average of surrounding pixels and a random color, less random as the grid size decreases. Three parameters: 'graininess' (.5 to 50, default = 2), old/new algorithm, seed value used.

popcorn
The orbits in two dimensions defined by:
x(0) = xpixel, y(0) = ypixel;
x(n+1) = x(n) - h*sin(y(n) + tan(3*y(n))
y(n+1) = y(n) - h*sin(x(n) + tan(3*x(n))
are plotted for each screen pixel and superimposed.
One parameter: step size h.

popcornjul
Conventional Julia using the popcorn formula:
x(0) = xpixel, y(0) = ypixel;
x(n+1) = x(n) - h*sin(y(n) + tan(3*y(n))
y(n+1) = y(n) - h*sin(x(n) + tan(3*x(n))
One parameter: step size h.

rossler3D
Orbit in three dimensions defined by:
x(0) = y(0) = z(0) = 1;
x(n+1) = x(n) - y(n)*dt - z(n)*dt
y(n+1) = y(n) + x(n)*dt + a*y(n)*dt
z(n+1) = z(n) + b*dt + x(n)*z(n)*dt - c*z(n)*dt
Parameters are dt, a, b, and c.

sierpinski
Sierpinski gasket - Julia set producing a 'Swiss cheese triangle'
z(n+1) = (2*x,2*y-1) if y > .5;
else (2*x-1,2*y) if x > .5;
else (2*x,2*y)
No parameters.

spider

c(0) = z(0) = pixel;
z(n+1) = z(n)^2 + c(n);
c(n+1) = c(n)/2 + z(n+1)

Parameters: real & imaginary perturbation of z(0)

sqr(1/fn)

z(0) = pixel;
z(n+1) = (1/fn(z(n))^2

One parameter: the function fn.

sqr(fn)

z(0) = pixel;
z(n+1) = fn(z(n))^2

One parameter: the function fn.

test

'test' point letting us (and you!) easily add fractal types via the c module testpt.c. Default set up is a mandelbrot fractal.
Four parameters: user hooks (not used by default testpt.c).

tetrate

z(0) = c = pixel;
z(n+1) = c^z(n)

Parameters: real & imaginary perturbation of z(0)

tim's_error

A serendipitous coding error in marksmandelpwr brings to life an ancient pterodactyl! (Try setting fn to sqr.)

z(0) = pixel, c = z(0) ^ (z(0) - 1):
tmp = fn(z(n))
real(tmp) = real(tmp) * real(c) - imag(tmp) * imag(c);
imag(tmp) = real(tmp) * imag(c) - imag(tmp) * real(c);
z(n+1) = tmp + pixel;

Parameters: real & imaginary pertubations of z(0) and function fn

unity

z(0) = pixel;
x = real(z(n)), y = imag(z(n))
One = x^2 + y^2;
y = (2 - One) * x;
x = (2 - One) * y;
z(n+1) = x + i*y

No parameters.

Zooming in on an Image

There are two different methods of generating and using "Zoom-Boxes". Which method you use depends on which of the "Zoom Box" checkboxes you have selected using the "Views" menuitem. If you've never selected any of these checkboxes, the "Zoom In" option is in effect.

If you have the "Zoom In Box" or "Zoom Out Box" options enabled, Winfract's zoom box is activated in the following fashion:

With the "Zoom In Box" option activated, move your mouse pointer to the center of the area you wish to zoom in on. Press and hold down the left mouse button as you move the mouse away from that center point. A "Zoom Box" will be displayed, changing in size as you move the mouse. When you have the zoom box the size you want, let go of the left mouse button.
If you want to adjust the location of the zoom box at this point, move your mouse pointer inside the zoom box, press and hold down the left mouse button again, and move the mouse. The zoom box follows the mouse pointer until you let go of the left mouse button.
To activate the zoom, double-click the left mouse button with the mouse pointer inside the zoom box. To disable the zoom box instead (say, you changed your mind, or the box isn't really the right size), double-click on the left mouse button with the mouse pointer positioned somewhere outside of the zoom box (pressing the ESCAPE key gets rid of the zoom box, too).

"Zoom Out" works just like "Zoom In", except that you will be zooming out - the zoom box is actually showing you where on the new image your current image will display.

If you have enabled the "Zoom-Box" option, a vertical Zoom-Box scroll-bar is displayed in addition to your fractal image. The Zoom-Box defaults to the mid-position, in which your Zoom-Box exactly covers the area of your fractal image. Moving the scroll-button up shrinks your zoom-box (and makes it visible on the fractal image). You can them move the zoom-box around by moving your cursor inside the zoom-box and then holding down your left mouse-button and moving the cursor and zoom-box as a single unit. Double-clicking the left mouse button will cause your image to be redrawn using your current zoom-box coordinates.

You can also "Zoom Out" by moving the zoom-box scroll bar below the midpoint. When you do this, the zoom-box that is displayed is actually showing you where (and how small) your currently displayed image would be if you double-clicked the left mouse button at that point. Moving the zoom-box and double-clicking to perform the zoom is done the same way as when you are "zooming in".

Mandelbrot/Julia Toggling

You can switch from a "Mandelbrot Set" image to its "Julia Set" at the location of the mouse cursor by clicking on the right mouse button. When the corresponding "Julia Set" image is on the screen, click the right mouse button again to get the "Mandelbrot Set" image back. The terms "Mandelbrot Set" and "Julia Set" are in quotes because many fractal types (mandel4 and julia4, for instance) have this "Mandelbrot Set"/"Julia Set" relationship.

Color Cycling

You start and stop Color-cycling mode using the Color-Cycling menu. This menu also lets you determine the direction of the cycling (the designations "in" and "out" were arbitrarily picked based on how the effect looked on a single Mandelbrot image), whether to just rotate the existing colors or generate new ones randomly, and (for random color-generation) whether the colors are to change with a low, medium, or high frequency.

You can also use the following "**Hot-keys**" for color-cycling
spacebar - toggles color-cycling on and off
'<', '>' - shifts the palette "in" or "out" one color
left, right arrows - turn on color-cycling and set the "direction"
up, down arrows - speed up or slow down the color-cycling speed
ENTER - initiate random color-cycling with all new colors
(uhh, note that Windows takes a second or so to "fire up" color-cycling, so be patient when you press one of these keys.)

Sorry, but at the current time color-cycling is restricted to display devices whose Windows drivers are capable of palette manipulation - and the "stock" VGA driver distributed with Win 3.x doesn't have it! In fact, at the moment the only Windows video drivers we've ever seen capable of palette manipulation have been 256-color drivers. If Winfract ever does find itself running with a palette-based 16-color video driver (it's theoretically possible), it will color-cycle only when its window has the input focus (temporarily converting the remainder of your Windows screen to black-and-white), and return to the "stock" palette values when it loses this focus.

Distribution Policy and Procedures, Contacting the Authors

The Stone Soup Group is a loosely associated group of fanatic programmers. Fractint for Windows, like Fractint-for-DOS, is freeware, and our "contribution policy" is the same: **Don't want money. Got money. Want admiration**.

Conditions on use:

Fractint for Windows, like Fractint-for-DOS, may be freely copied and distributed but may not be sold (a nominal distribution fee may be charged for media and handling by freeware and shareware distributors.) Winfract may be used personally or in a business - if you can do your job better by using Winfract, or using images from it, that's great! It may not be given away with commercial products without explicit permission from the Stone Soup Group.

There is no warranty of Winfract's suitability for any purpose, nor any acceptance of liability, express or implied.

Virtually all of the Fractint and Winfract authors can be found on the **CompuServe** network in the **COMART** ('COMputer ART') forum in S 15 ('Fractals'). Fractint development occurs in the COMART forum - most of the authors have never met except there. New members are always welcome!

New versions of Fractint-for-DOS and Fractint-for-Windows are uploaded to the CompuServe network, and make their way to other systems from that point. The latest version of Fractint-for-Windows can be found in the "Fractals" library of COMART as **WINFRA.ZIP** (Executable/Docs) and **WINSRC.ZIP** (Complete Source). 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). In particular...

If you speak Internet and FTP, SIMTEL20 and its various mirror sites tend to carry new versions of Winfract shortly after they are released. Look in the PD:<MSDOS.GRAPHICS> and PD:<MSDOS.WINDOWS3> directories for these files .

Your favorite local BBS may carry these files as well (although perhaps not the latest versions) using naming conventions like WINFRA*.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.

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 713-524-6398). 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.

...

(What's the latest version? Well, THIS one was, way back when we uploaded it!)

A list of Fractint Authors

Fractint for Windows was originally ported from Fractint-for-DOS by Bert Tyler. This is the first Windows program that Bert ever wrote,which goes a long way towards explaining a lot of the bugs. Bert's task was made a lot easier by Pieter Branderhorst, who separated the MSDOS-specific code from Fractint-for-DOS's fractal generator modules, making a Windows port of the package possible. Mark Peterson helped track down and eliminate a lot of the bugs that were in the original program, and has since added many of the new options which take advantage of the capabilities of the Windows environment. All of the current zooming functions, for example, are from Mark.

Fractint for Windows is based heavily on (and uses the fractal generator engines straight out of) Fractint-for-DOS. A partial list of the authorsof Fractint-for-DOS, taken from Fractint version 17.1, includes:

---------- Primary Authors (this changes over time) ----------

Bert Tyler	- CompuServe (CIS) ID: [73477,433]
Timothy Wegner	- CIS ID: [71320,675] Internet: twegner@mwunix.mitre.org
Mark Peterson	- CIS ID: [70441,3353]
Pieter Branderhorst	- CIS ID: [72611,2257]

--------- Contributing Authors ----------

Michael Abrash	360x480x256, 320x400x256 VGA video modes
Joseph Albrecht	Tandy video, CGA video speedup
Kevin Allen	Finite attractor and bifurcation engine
Steve Bennett	restore-from-disk logic
Rob Beyer	[71021,2074] Barnsley IFS, Lorenz fractals
Mike Burkey	376x564x256, 400x564x256, and 832x612x256 VGA video modes
John Bridges	[75300,2137] superVGA support, 360x480x256 mode
Brian Corbino	[71611,702] Tandy 1000 640x200x16 video mode
Lee Crocker	[73407,2030] Fast Newton, Inversion, Decomposition..
Monte Davis	[71450,3542] Documentation
Chuck Ebbert	[76306,1226] cmprsd & sqrt logmap, fpu speedups
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
Wesley Loewer	fast floating-point Mandelbrot/Julia logic
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
Jonathan Osuch	[73277,1432] IIT detect
Marc Reinig	[72410,77] Lots of 3D options
Kyle Powell	[76704,12] 8514/A Support
Matt Saucier	[72371,3101] Printer Support
Herb Savage	[71640,455] 'inside=bof60', 'inside=bof61' options
Lee Skinner	Tetrate, Spider, Mandelglass fractal types and more
Dean Souleles	[75115,1671] Hercules Support

Kurt Sowa	[73467,2013] Color Printer Support
Hugh Steele	cyclerange feature
Chris Taylor	Floating&Fixed-point algorithm speedups, Tesseral Option
Scott Taylor	[72401,410] (DGWM18A) 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
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

Coordinate Box options

The **Coordinate Box** is courtesy of Mark Peterson. When you have the Coordinate Box checked, a small "coordinate window" constantly displays the current position of your mouse pointer.

From the **Options** menu you can select the coordinates to display in rectangular (default), polar, or pixel coordinates.

Rectangular coordinates correspond to Cartesian plane. The coordinates displayed are in absolute units relative to the origin. Fractint uses these coordinates to form a complex number. This complex number initializes one or more variables in the iterative calculation. The *x* coordinate is used as the *real* portion of the complex number and the *y* coordinate as the *imaginary* portion.

Pixel coordinates display the position of a point in terms of the number of pixels, or color dots, relative to the pixel in the upper left-hand corner of the image. For example, if the image size is 200 x 150, then the pixel in the lower right-hand corner of the image is coordinate (199, 149).

Polar coordinates display the position of a point in terms of its distance and angle relative to the origin. The angle can be in units of **degrees** (default), **radians**, or **grads**. Most people are familiar with degrees which divide the circle into 360 *degrees*. Grads divide the circle into 400 *grads*. Radians divide the circle into units of 2 Pi *radians*.