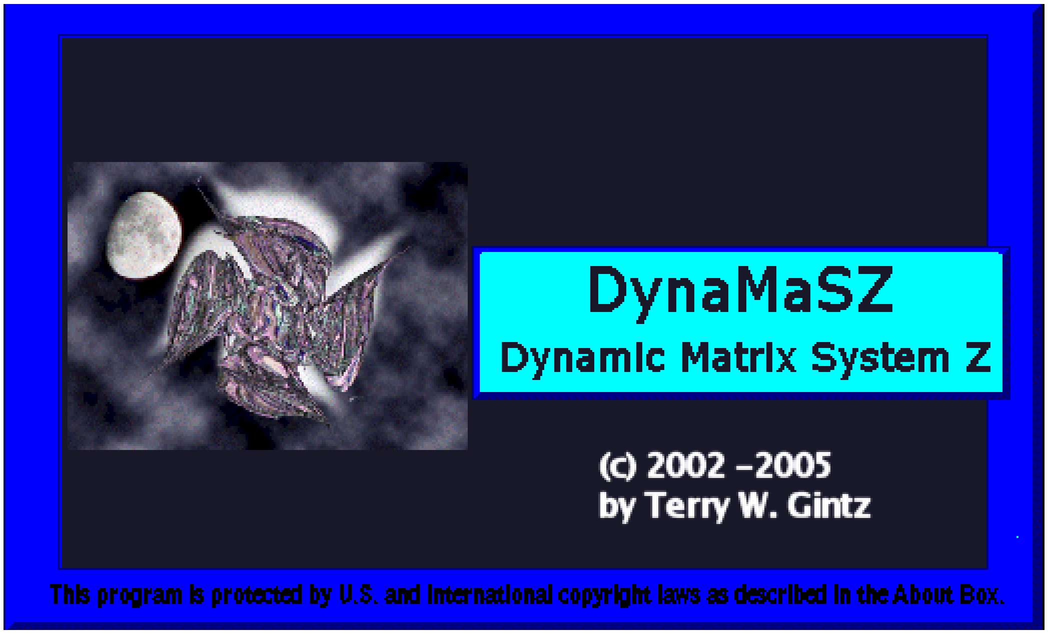


DynaMaSZ Application Help

Table of Contents

1 Main Index

DynaMaSZ Help Index

A DynaMaSZ Tutorial by Godwin Vickers

Matrix Algebra, An Introduction by Godwin Vickers

DynaMaSZ Remote

Commands

File menu
Edit menu
Image menu
Type menu
Render menu
Pixel menu
View menu
Window menu
Video menu
Demo menu
Help menu

1.1 Title Bar

Title Bar

The title bar is located along the top of a window. It contains the name of the application and drawing.

To move the window, drag the title bar. Note: You can also move dialog boxes by dragging their title bars.

A title bar may contain the following elements:

- Application Control-menu button
- Drawing Control-menu button
- Maximize button
- Minimize button
- Name of the application
- Name of the drawing
- Restore button

1.2 Scroll bars

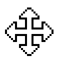
Scroll bars

Displayed at the right and bottom edges of the drawing window. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the drawing. You can use the mouse to scroll to other parts of the drawing.

1.3 Size

Size command (System menu)

Use this command to display a four-headed arrow so you can size the active window with the arrow keys.



After the pointer changes to the four-headed arrow:
1. Press one of the DIRECTION keys (left, right, up, or down arrow key) to move the pointer to the border you want to move.
2. Press a DIRECTION key to move the border.
3. Press ENTER when the window is the size you want.

Note: This command is unavailable if you maximize the window.

Shortcut
Mouse: Drag the size bars at the corners or edges of the window.

1.4 Move

Move command (Control menu)

Use this command to display a four-headed arrow so you can move the active window or dialog box with the arrow keys.

Note: This command is unavailable if you maximize the window.

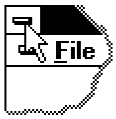
Shortcut
Keys: CTRL+F7

1.5 Close

Close command (Control menus)

Use this command to close the active window or dialog box.

Double-clicking a Control-menu box is the same as choosing the Close command.



Shortcuts
Keys: CTRL+F4 closes a drawing window
ALT+F4 closes the application

1.6 Restore

Restore command (Control menu)

Use this command to return the active window to its size and position before you chose the Maximize or Minimize command.

1.7 Minimize Command

Minimize command (application Control menu)

Use this command to reduce the DynaMaSZ window to an icon.

Shortcut

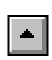
Mouse: Click the minimize icon  on the title bar.
Keys: ALT+F9

1.8 Maximize Command

Maximize command (System menu)

Use this command to enlarge the active window to fill the available space.

Shortcut

Mouse: Click the maximize icon  on the title bar; or double-click the title bar.
Keys: CTRL+F10 enlarges a drawing window.

1.9 Next Window

Next Window command (drawing Control menu)

Use this command to switch to the next open drawing window. DynaMaSZ determines which window is next according to the order in which you opened the windows.

Shortcut
Keys: CTRL+F6

1.10 Previous Window

Previous Window command (drawing Control menu)

Use this command to switch to the previous open drawing window. DynaMaSZ determines which window is previous according to the order in which you opened the windows.

Shortcut
Keys: SHIFT+CTRL+F6

1.11 Switch to

Switch to command (application Control menu)

Use this command to display a list of all open applications. Use this "Task List" to switch to or close an application on the list.

Shortcut
Keys: CTRL+ESC

Dialog Box Options
When you choose the Switch To command, you will be presented with a dialog box with the following options:

Task List
Select the application you want to switch to or close.

Switch To
Makes the selected application active.

End Task
Closes the selected application.

Cancel
Closes the Task List box.

Cascade
Arranges open applications so they overlap and you can see each title bar. This option does not affect applications reduced to icons.

Tile
Arranges open applications into windows that do not overlap. This option does not affect applications reduced to icons.

Arrange Icons
Arranges the icons of all minimized applications across the bottom of the screen.

2 DynaMaSZ Remote

DynaMaSZ Remote

The remote provides access to many of the most-used commands in DynaMaSZ. Info about each button can be obtained by using the '?' located near the close box in the top right-hand corner.

2.1 New

New button

Use this button to open a new drawing window in DynaMaSZ. This is useful to view minor changes to a drawing. Use the Copy Data and Paste Data commands from the Edit menu to transfer current drawing parameters to the new window.

2.2 Undo

Undo button

Use this command to undo the last action. An image can be continued after an undo, if continue was enabled before the last action, but not after an image is resized. Color-cycling is disabled after using Undo.

2.3 Size

Size button

This allows you to set the drawing area for a picture, independent of the Windows screen size. It also shows which size is currently in use. The aspect for the drawing is based on the ratio of X (horizontal width) to Y (vertical height.) The custom setting allows for any size/aspect that system memory will permit. Videos are limited to the standard 4/3 vga aspect or 1/1. The minimum size for an image is 40X30.

2.4 Color

Color button

Use the palette editor to modify the color palette in use.

2.5 Batch

Batch button

Here you set parameters for randomizing, batching and saving random-generated images to disk.

2.6 View

View button

Displays the entire plot, expanding or shrinking the image to fit in a maximized window without title bar, scroll bars or menu bar. At all other times, part of the picture is hidden by the inclusion of the title bar, toolbar, scroll bars and menu bar. To exit full-screen mode, press any key or click the left-mouse button.

2.7 Draw button

Draw button

Use this button to draw or redraw the image for the current fractal variables. Clicking inside the draw window with the left-mouse button stops all plotting. Use the Cont button to restart plotting from the current column.

2.8 Abort button

Abort button

Use this command to stop drawing. Clicking inside a window's drawing area or close box (or the program close box) will also stop the drawing. Note: once a plot has started DynaMaSZ continues to draw the image for that window regardless of which drawing window has the input focus, until done or aborted. You can open and close other drawing windows without affecting the current drawing, but only one drawing is active at any time.

2.9 Cont

Continue button

Continues a plot that was aborted early. The plot is restarted at the beginning of the last row drawn.

2.10 Scan

Scan button

This generates a random Julia set from a formula's Mandelbrot 'P' space. Random points in a formula's current Mandelbrot space are scanned for an interesting Julia set. Rendering options are unchanged.

2.11 Rend

Rend button

The current plot is ray-traced using whatever rendering values are set. If the plot has been drawn in the current session (without using undo or reloading) then the ray-tracing will take place in real time, else the plot will be redrawn and then ray-traced.

2.12 Light

Light button

Edit lightpoint and viewpoint variables

2.13 Pilot

Pilot

Opens the Pilot window to adjust key parameters, rotate, zoom and redraw the figure interactively. The current image is reduced to one quarter normal for faster redraw. Each click on a Pilot button increments or decrements a parameter. The Speed slider controls the rate at which the buttons operate (default is 10.)

Press the space bar or Click on Ok to open a new window and draw the altered image full-size. Press Esc or click on Cancel to exit this mode without opening a new window. Note: when using this option while an AVI stream is open, a new window isn't opened, but the altered figure is drawn in the current draw window, the changed parameters replacing the previous ones.

2.14 Text

Text

Allows you to edit text and font and apply it to a drawing. Select the font button to set the font style, size and color. In the text window click on Okay to add a line of text to the current image. (You can add multiple lines of text too, up to 80 characters.) The cursor will change to a crosshair. Position the cursor where you want the text to start and left-click the mouse. Note: font and title text are saved in the file "prefs.txt" in DynaMaSZ's startup directory. Title text can also be edited (as a single line only) in the Edit/Formula window.

2.15 Help

Help button

Use this button to access the DynaMaSZ Help Index.

2.16 Formula

Formula button

Use this button to change formulas or type.

2.17 Params

Params button

Use this button to edit 3-D parameters.

2.18 Mat

Mat button

This generates a random Julia set according to the options set in the Batch window.

2.19 Comp

Comp button

This generates a random composite Julia set according to the options set in the Batch window.

2.20 Rand Rend

Rand Rend button

The current plot is ray-traced using random rendering values (randomizes coloring filter or generalized coloring parameters.) If the plot has been drawn in the current session (without using undo or reloading) then the ray-tracing will take place in real time, else the plot will be redrawn and then ray-traced.

2.21 Save

Save button

Use this button to save and name the active drawing. DynaMaSZ displays the Save As dialog box so you can name your drawing.

To save a drawing with its existing name and directory, use the Save command in the File menu.

2.22 Load

Load button

Use this button to open an existing data/image file in a new window. You can open multiple image files at once. Use the Window menu to switch among the multiple open images.

2.23 Bmp

BMP radio button

Use this button to select the BMP format when loading and saving fractals. This is the default Windows bitmap format, readable by most Windows programs that use image files. This is also the fastest method of loading and saving fractals, but requires more disk space, since no compression is used. Windows keeps track of the last six BMP files saved or loaded (displayed in the Files menu.)

2.24 Png

PNG radio button

Use this button to select the PNG format when loading and saving fractals. This format uses medium compression without loss of image quality.

2.25 Jpg

JPG radio button

Use this button to select the JPEG format when loading and saving fractals. This format uses moderate compression but with some loss of image quality. Preferable for posting to the net, since most browsers can display jpeg files.

2.26 |||||

||||| button

Through a series of windows, this allows you to name and open an avi animation stream and choose a compression method. After choosing the frame rate (1-60) and using the file requester to name the file, you are given a choice of compression methods. You can also choose no compression for optimum view quality. (All compression methods degrade the original images, some more than others.) The first key frame in the stream is then drawn and written to the file.

Note: after the stream is opened, the size of the fractal that can be drawn is fixed at the size of the frame. No changes can be made to the size until the stream is closed.

2.27 >

> button

With this option, frames are written to a stream based on the difference between the current key frame and the previous key frame. The first key frame is written when you open a stream. The next key frame is created each time you use this option. In between you can zoom or change Fvr variables as much as necessary. The stream is only written to when this option is used. The last key frame is automatically saved after the 'tween' series is written. The number of frames may range from 1-1500 frames between keys. With a frame number of 1 only the key frames are written. This allows animation to be created that incorporate all scalable variables in DynaMaSZ.

Use the Cancel button to exit this dialog without initializing a new series of frames.

Check the Log Scaling box if you want the frames to be written with logarithmic space between frames, else linear space is used. Useful when zooming, where frames would otherwise be packed together at the end of the frame series.

2.28 []

[ ] button

Closes any open avi stream file. You need to do this before viewing the file or creating a new avi file. The stream is also closed when you exit DynaMaSZ.

2.29 V

V button

Opens an avi file for viewing. You can preview any multimedia file by clicking on its file name. A multimedia box will appear to the right of the file list. Click on okay to open the main view window.

There are buttons to Play a file forwards or Backwards, or forward automatically with Auto rewind/repeat. Click on Slow to slow down a video. Each click on Slow halves the viewing speed. A click on Stop freezes viewing and restores the view speed to normal playback.

Use the Open button to view a different avi file. Use the Save button to save the file in a different compression format. You must use a different name to save the file than the name that was used to open it. Click on the left-mouse button or any key to abort a save operation.

Note: the view avi requester can be used to preview any multimedia file, including midi files.

3 File menu

File menu commands

The File menu offers the following commands:

New	Creates a new drawing.
Open	Opens an existing drawing.
Close	Closes an opened drawing.
Save	Saves an opened drawing using the same file name.
Save As	Saves an opened drawing to a specified file name.
Load Parameters	Load parameters from an existing drawing.
Load Palette [PQZ]	Load palette file.
Load Palette [MAP]	Load a Fractint map file.
Load Texture	Load QuaSZ texture file [QTX]
Open [JPG]	Load jpeg.
Open [PNG]	Load png.
Open [JPG+DZB]	Load JPEG + Zbuffer file.
Save Parameters	Save parameters for an opened drawing to a specified file name.
Save Palette [PQZ]	Save palette to file.
Save Texture	Save texture file [QTX].
Save AS [JPG]	Save in jpeg format.
Save As [PNG]	Save in png format.
Save AS [JPG+DZB]	Save in JPEG format with Zbuffer.
Import Parameters [FPR]	Load parameter file created with Fractal Projector

Export Options

Save as OBJ	Save polygonized matrix as Wavefront object.
Simplify Mesh	Simplify mesh.
Save as POV	Save polygonized matrix as a pov triangle object.
Smooth Triangles	Convert triangle mesh to smooth_triangle mesh.
Set Max Faces	Set target face size for simplify and smooth options.
Save as WRL	Save polygonized matrix as virtual reality file.
Save as DXF	Save polygonized matrix as AutoCad dxf file.
Save as LFM	Save polygonized matrix as LightFlow mesh file.
Set Max Vertices	Set maximum number of vertices allocated for M polygon.
Exit	Exits DynaMaSZ.

3.1 File New command

New command (File menu)

Use this command to create a new drawing window in DynaMaSZ. The image and data for the opening picture are used to create the new window.

You can open an existing data/image file with the Open command.

Shortcuts

Keys: CTRL+N

3.2 File Open command

Open command (File menu)

Use this command to open an existing data/image file in a new window. You can open multiple image files at once. Use the Window menu to switch among the multiple open images. See Window 1, 2, ... command.

You can create new images with the New command.

Shortcuts

Toolbar: 

Keys: CTRL+O

3.2.1 File Open dialog box

File Open dialog box

The following options allow you to specify which file to open:

File Name

Type or select the filename you want to open. This box lists files with the extension you select in the List Files of Type box.

List Files of Type
Select the type of file you want to open:
<< List your application's file types here. >>

Drives
Select the drive in which DynaMaSZ stores the file that you want to open.

Directories
Select the directory in which DynaMaSZ stores the file that you want to open.

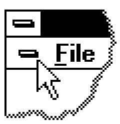
Network...
Choose this button to connect to a network location, assigning it a new drive letter.

3.3 File Close command

Close command (File menu)

Use this command to close the window containing the active image. If you close a window without saving, you lose all changes made since the last time you saved it.

You can also close a drawing by using the Close icon on the drawing window, as shown below:



3.4 File Save command

Save command (File menu)

Use this command to save the active drawing to its current name and directory. When you save a drawing for the first time, DynaMaSZ displays the Save As dialog box so you can name your drawing. If you want to change the name and directory of an existing drawing before you save it, choose the Save As command.

Shortcuts

Toolbar: 
Keys: CTRL+S

3.5 File Save As command

Save As command (File menu)

Use this command to save and name the active drawing. DynaMaSZ displays the Save As dialog box so you can name your drawing.

To save a drawing with its existing name and directory, use the Save command.

3.5.1 File Save As dialog box

File Save As dialog box

The following options allow you to specify the name and location of the file you're about to save:

File Name

Type a new filename to save a drawing with a different name. A filename can contain up to eight characters and an extension of up to three characters. DynaMaSZ adds the extension you specify in the Save File As Type box.

Drives

Select the drive in which you want to store the drawing.

Directories

Select the directory in which you want to store the drawing.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

3.6 File Load Parameters command

Load Parameters command (File menu)

Use this command to load a data file [.dmz]. The data file contains all variables to recreate an image created previously with DynaMaSZ.

3.7 File Load Palettes command

Load Palette command (File menu)

Use this command to load a palette file [.pqz]. The palette file contains a palette created previously with DynaMaSZ. You also have the option in the file descriptor box to select palette and coloring filter, to reload both palette and the coloring filter that was saved along with it.

3.8 File Load Palette command

Load Palette [MAP] command (File menu)

Use this command to load a Fractint-type map file. The palette in the map file replaces the currently selected palette.

3.9 File Load Texture command

Load Texture command (File menu)

Use this command to load variables that make up the texture and noise parameters. This also loads the palette, coloring filter, orbit trap and coloring options in a texture file [qtx].

3.10 File Open [JPG] command

Open [JPG] command (File menu)

Use this command to load parameters and a bitmap file that were saved in jpeg format. There is an option in the file-type box to load only the bitmap too. This replaces the Open command for those who need a smaller sized bitmap file. Note: the last files list doesn't keep track of images loaded in JPEG format.

3.11 File Open [PNG] command

Open [PNG] command (File menu)

Use this command to load parameters and a bitmap file that was saved in png format. There is an option in the file-type box to load only the bitmap too. This replaces the Open command for those who need a smaller sized bitmap file. Note: the last files list doesn't keep track of images loaded in PNG format.

3.12 File Open [JPG+DZB] command

Open [JPG+DZB] command (File menu)

Use this command to load parameters, zbuffer file and a bitmap file that was saved in jpeg format. This option is provided so that you can continue a plot that was saved in another draw session. Note: the last files list doesn't keep track of images loaded in JPEG format.

3.13 File Save Parameters command

Save Parameters command (File menu)

Use this command to save all data elements for the current image in a data file [.dmz].

3.14 File Save Palette command

Save Palette command (File menu)

Use this command to save a palette for the current image in a palette file [.pqz]. Also saves the Coloring Filter used for surface mapping.

3.15 File Save Texture command

Save Texture command (File menu)

Use this command to save the variables that make up the texture and noise parameters for the current figure. This also saves the palette, coloring filter, orbit trap and coloring options in the texture file [qtx].

3.16 File Save As [JPG] command

Save As [JPG] command (File menu)

Use this command to save the parameters and active bitmap in jpeg format. There is an option in the file-type box to save only the bitmap too. This replaces the Save and Save As command for those who need a smaller sized bitmap file. Note: the last files list doesn't keep track of images saved in JPEG format.

3.17 File Save As [PNG] command

Save As [PNG] command (File menu)

Use this command to save the parameters and active bitmap in png format. There is an option in the file-type box to save only the bitmap too. This replaces the Save and Save As command for those who need a smaller sized bitmap file. Note: the last files list doesn't keep track of images saved in PNG format.

3.18 File Save As [JPG+DZB] command

Save As [JPG+DZB] command (File menu)

Use this command to save the parameters, zbuffer and active bitmap in jpeg format. This option is provided so that you can continue a plot in another draw session. Not available for anti-aliased picture. To enable this option the zbuffer must be created first by initializing a plot, using the Draw command or Auto-Draw option. Note: the last files list doesn't keep track of images saved in JPEG format.

3.19 File Import Parameters [FPR] command

Import Parameters [FPR] command (File menu)

Use this command to load a Fractal Projector-type parameter file.

3.20 File Save Q Polygon [OBJ] command

Export -> Save as OBJ command (File menu)

Use this command to save a matrix image as a true 3-D object. This uses John C. Hart's Implicit code (Quaternion Julia Set server) to polygonize a matrix formula, and then writes the triangles to a Wavefront object file. The memory requirements for this routine are high, 20MB or more for a typical Julia set rendered at 320X240. The output file can be very large too, 4MB or more, depending on the precision required. The higher the precision, the smoother the finished object. Precision is set with the Steps variable in the Initial Values window, where precision=10/Steps.

Note: some formulas produce asymmetrical object files with this routine, where one side of the matrix polygon isn't resolved completely. Usually one side is markedly smoother in this case.

3.21 File Simplify mesh command

Export -> Simplify Mesh command (File menu)

When this flag is set (default on) the object meshes are simplified using Garland's mesh-simplification algorithm before outputting to a Wavefront obj or POV mesh file, resulting in a much smaller export file. Set the number of facets in the target mesh file with the Set Max Faces command. You can set the resolution of the object as high as necessary (with the Params/Steps variable) to produce a finely detailed quaternion, but the output file remains about the same. Use the smoothing feature in Bryce to smooth the resulting object mesh.

3.22 File Save Q Polygon[POV] command

Export -> Save as POV command (File menu)

Use this command to save a quaternion as a true 3-D object. This uses John C. Hart's Implicit code (Quaternion Julia Set server) to polygonize a matrix formula, and then outputs the triangles to a pov file. The pov file is written as a simple scene, the triangles part of a "union" object, with camera and lighting elements compatible with POV 3.5. This can be used as a starting point for more complex compositions. The memory requirements for this routine are 20MB or more for a typical Julia set matrix rendered at 320X240. The output file can be very large too, up to 40MB or more, at the highest precision. The higher the precision, the smoother the finished object becomes. Precision is set with the Steps variable in the Initial Values window, where precision=10/Steps.

Note: some formulas produce asymmetrical object files with this routine, where one side of the matrix polygon isn't resolved completely. Usually one side is markedly smoother in this case.

3.23 File Smooth command

Export -> Smooth command (File menu)

When this flag is set (default on) the object facets are converted to smooth_triangles before outputting to a POV mesh file. Surface normals are calculated for all triangles that share common vertices.

3.24 File Set max faces command

Export -> Set Max Faces command (File menu)

Set the number of facets in the target mesh file (obj or POV format.) Use the minimum number of faces to produce the quality of mesh desired. It is better to reduce faces to a minimum and do the smoothing in Bryce than to export an object at maximum resolution without mesh simplification. Object files load and smooth much faster in Bryce, and smoothing is usually necessary anyway to reduce jagged edges or blocky facets. (POV uses smooth_triangles to accomplish the same thing.) Up to 30X mesh reduction or more is possible with Garland's mesh-simplification algorithm.

3.25 File Save Q Polygon [WRL] command

Export -> Save as WRL command (File menu)

Use this command to save a matrix image as a true 3-D object. This uses John C. Hart's Implicit code (Quaternion Julia Set server) to polygonize a matrix formula, and then writes the triangles to a virtual reality file. The memory requirements for this routine are high, 20MB or more for a typical Julia set matrix rendered at 320X240. The output file can be very large too, 4MB or more, depending on the precision required. The higher the precision, the smoother the finished object. Precision is set with the Steps variable in the Initial Values window, where precision=10/Steps.

Note: some formulas produce asymmetrical object files with this routine, where one side of the matrix polygon isn't resolved completely. Usually one side is markedly smoother in this case.

3.26 File Save Q Polygon[DXF] command

Export -> Save as DXF command (File menu)

Use this command to save a quaternion as a true 3-D object. This uses John C. Hart's Implicit code (Quaternion Julia Set server) to polygonize a matrix formula, and then writes the triangles to an AutoCad dxf file. The memory requirements for this routine are 20MB or more for a typical Julia set matrix rendered at 320X240. The output file can be very large too, 4MB or more, depending on the precision required. The higher the precision, the smoother the finished object becomes. Precision is set with the Steps variable in the Initial Values window, where precision=10/Steps.

Note: some formulas produce asymmetrical object files with this routine, where one side of the matrix polygon isn't resolved completely. Usually one side is markedly smoother in this case.

3.27 File Save Q Polygon [LFM] command

Export -> Save as LFM command (File menu)

Use this command to save a matrix image as a true 3-D object. This uses John C. Hart's Implicit code (Quaternion Julia Set server) to polygonize a matrix formula, then writes the triangles to a LightFlow mesh file. The memory requirements for this routine are high, 20MB or more for a typical Julia set matrix rendered at 320X240. The output file can be very large too, 4MB or more, depending on the precision required. The higher the precision, the smoother the finished object. Precision is set with the Steps variable in the Initial Values window, where precision=10/Steps.

Note: some formulas produce asymmetrical object files with this routine, where one side of the matrix polygon isn't resolved completely. Usually one side is markedly smoother in this case.

3.28 File Set Max Vertices command

Set Max Indices (File menu)

Use this command to set the maximum number of indices that are allocated by the polygonizing routine. Default is 5,000,000 indices. Use less to limit the amount of memory used while polygonizing. Use more if necessary for higher resolution. Note: unless you have an application that can use very large object files, there's a limit to how much resolution is obtainable with the polygonizing routine. Bryce4 has problems with object files produced by DynaMaSZ that are much larger than 2.5MB.

3.29 File 1, 2, 3, 4, 5, 6 command

1, 2, 3, 4, 5, 6 command (File menu)

Use the numbers and filenames listed at the bottom of the File menu to open the last six drawings you closed. Choose the number that corresponds with the drawing you want to open.

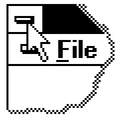
3.30 File Exit command

Exit command (File menu)

Use this command to end your DynaMaSZ session. You can also use the Close command on the application Control menu. Note: if you choose to exit while plotting, the program does not terminate, but stops the plotting so the program can be safely exited.

Shortcuts

Mouse: Double-click the application's Control menu button.



Keys: ALT+F4

4 Edit menu

Edit menu commands

The Edit menu offers the following commands:

Undo	Undo last edit, action or zoom.
Copy	Copy the active view and put it on the Clipboard.
Clip	Define area of view and copy to clipboard.
Paste	Insert Clipboard contents.
Copy Data	Copy fractal data to buffer.
Paste Data	Copy data from copy buffer.
Formulas/Type	Edit formula/type data.
Drawing Parameters	Edit drawing window parameters.
Size	Sets the image size.
Initial Values	Edit initial image parameters.
Ray-Tracing Variables	Edit lighting and viewpoint variables.
Palette Editor	Edit palette.
Text	Edit and add text to drawing.
Preferences	Startup preferences and defaults.

4.1 Edit Undo command

Undo command (Edit menu)

Use this command to undo the last action. An image can be continued after an undo, if continue was enabled before the last action. Color-cycling is disabled after using Undo, though.

Shortcut
Keys: CTRL+Z

4.2 Edit Copy command

Copy command (Edit menu)

Use this command to copy the active view to the clipboard. The entire view is copied to the clipboard.

Shortcut
Keys: CTRL+C

4.3 Edit Clip command

Clip command (Edit menu)

Use this command to copy a part of the active view to the clipboard. A zoom box is used to select the part to be copied. Click outside the view frame or press escape to exit this option.

Shortcut
Keys: CTRL+L

4.4 Edit Paste command

Paste command (Edit menu)

Use this command to paste from the clipboard. The clipboard must contain a bitmap. If the bitmap is larger than the view, it is clipped. The zoom cursor is used to set the left/top corner in the view where the bitmap will be pasted. Click outside the view frame or press escape to exit this option.

Shortcut
Keys: CTRL+V

4.5 Edit Copy Data command

Copy Data command (Edit menu)

Use this command to copy the fractal data for the active view to the file "c:\zcopy.dmz". The current palette for the view is also copied.

Shortcut
Keys: CTRL+F

4.6 Edit Paste Data command

Paste Data command (Edit menu)

Use this command to paste the data in the file "c:\zcopy.dmz" to the active view. The palette stored in the file is copied to palette 10(F11).

Shortcut
Keys: CTRL+R

4.7 Formula Window

Formula/Type Window

Fun #1 and Fun #2 are combo controls selecting up to two built-in formulas. There is additionally a Type control, Latin control, a Column control and a Sign control that determine how the above formulas are processed. The 24 built-in Latin squares and eight sign matrices determine the matrix algebra used in processing the formulas. (For more details on the basis of matrix algebra and the use of the Column control, see Godwin's tutorial on Matrix Algebra.) The Custom Latin and Custom Sign buttons open editors for the Latin squares and sign matrices, so you can customize the matrix algebra. You can enter a value of 0-23 into the Latin control to use one of the built-in Latin squares, or click on the Custom Latin button and make up your own square. The Latin control will be set to 24, for custom. Similarly you can enter a value of 0-7 to use a built-in sign matrix, or click on Custom Sign, make your own, and the Sign control will be set to 8. Note: if you set the Latin or Sign controls to anything outside of their built-in ranges without using the custom editors, the default custom Latin square and sign matrix are 0 and 0 in their built-in lists. The Column control accepts values of 0-3 and determines which column is used in squaring the combined matrix.

The Type control accepts a value of 0 to 9. For a value of 0, the first formula is always used and the second formula is ignored. For a value of 1, the second formula is processed and the first formula is ignored. Type 1 is of use only if you are switching between two functions and don't want to reenter them each time you plot the other one. For a value of 2, the first formula is processed if the iterative result is greater than or equal to .5. For values of 3, the first formula is processed and its output becomes the input of the second formula, which is then processed.

Type 4 takes the average of Fun#1 and Fun#2.

Type 5 alternates between the two functions while iterating.

Type 6 takes the quotient of both functions.

Type 7 uses the lowest iterative results of both Fun#1 and Fun#2

Type 8 uses the highest iterative results of both Fun#1 and Fun#2

Type 9 uses the Formula box to enter up to 1000 characters per formula.
Text can be pasted from the clipboard to the formula box by using the keystrokes shift-insert.
Text may be moved from box to box by using shift-delete to move it first to the clipboard.

The three buttons named Rand fun#1, Rand fun#2 and Random Formula are used to pick formulas/functions at random. Clicking on Rand fun#1, a formula is chosen (from the first 100 built-in formulas) for fun #1. Clicking on Rand fun#2, a formula is chosen for fun #2. Clicking on Random Formula, a random formula is generated in the Formula box and the Type is set to 9. The Level box determines the complexity of the formula generated.

About formula syntax: This applies if you elect to enter your own formula into one of the function boxes and use the parser to generate the plot. Up to 500 user-named-complex variables and constants may be included in a formula. A variable must begin with a letter and may contain numbers and letters only. A variable may be up to 9 characters long. A constant may be up to 20 digits long, including the decimal point. DynaMaSZ uses syntax similar to Fractint's formula style with an initialization section, followed by the main formula, and an optional bailout routine. Bailout is otherwise handled by the 'Bailout' variable in the Initial Values window. Comments may be entered on the same line with a preceding ';'. A ':' terminates the initialization section. Multiple phrases may be entered in the main formula or initialization sections on the same line by using the terminator ',' between phrases. For a complete list of variables, operators and functions recognized by the parser, see Parser Information.

The Title text box is used with the hot key 'T' to annotate a picture with text. Use the Edit/Text command to change font, text color or format text into multiple lines. Text in this box is not saved in a picture's data file, but once entered the same text can be used over and over for different pictures. Useful for adding copyright/author info to batches of pictures. Since the same title text may be used many times, it is shared among views and saved in the file "prefs.txt" in DynaMaSZ's startup directory.

Click on the Okay button to use the formulas currently displayed in the window, or Cancel to exit the window without making any changes. Click on Apply to apply a new formula, etc. without closing the Formula window.

The Reset button returns all boxes and slider values to their original values when the window was opened.

4.7.1 Custom Sign Matrix Editor

Custom Sign Matrix Editor

Here you enter a value (-1 or 1) into each of the squares that make up the sign matrix. Each square represents an index to the "signs" of the composite matrix. You can also use the row or column check boxes to swap one column in the matrix with another column, or one row with another row. Select on each side of the columns or rows which column or row is to be swapped with the other, then click on Swap Cols or Swap Rows. Click on the Random Signs button to generate a random sign matrix.

4.7.2 Custom Latin Square Editor

Custom Latin Square Editor

Here you enter a value for each of the squares that make up the Latin square. You can also use the row or column check boxes to swap one column in the square with another column, or one row with another row. Select on each side of the columns or rows which column or row is to be swapped with the other, then click on Swap Cols or Swap Rows. Normally there is only one of each element (a, b, c...) in any column or row, but you can vary this to create even more algebraic possibilities. Click on the Random Latin button to generate a random Latin square.

4.8 Parameters Window

Drawing Parameters Window

The Size slider controls the overall size of the picture. The Size slider sets the horizontal resolution, while the vertical resolution is then scaled according to the full-screen VGA ratio, 4 to 3(1:1 if that aspect is selected through the Auto menu.) The Sector slider controls which of 4 sectors the picture will be drawn in, if the Size is less than or equal to (the full-screen horizontal resolution)/2. Otherwise the picture is centered according to the full-screen dimensions. This allows you to show zooms of a particular function by using different sectors, or show the affect of different plotting options. Each sector is erased individually. Note: if you try to continue a plot in a different sector than you started with, the plot will continue in the original sector. The Thumbnail button next to the Size slider is used to set a thumbnail size quickly. The thumbnail size toggles between 1/4 and 1/8 of the horizontal screen resolution, e.g. 200X150 or 100X75 for an 800X600 screen.

Select the Okay button to start a new plot from column 1. Select the Continue button to continue a plot at the row it left off, if it is not a complete drawing.

The Reset button returns all boxes and slider values to their original values when the window was opened.

4.9 Size

Size (Edit menu)

This allows you to set the drawing area for a picture, independent of the Windows screen size. It also shows which size is currently in use. The aspect for the drawing is based on the ratio of X (horizontal width) to Y (vertical height.) The custom setting allows for any size/aspect that system memory will permit. Videos are limited to the standard 4/3 vga aspect or 1/1. The minimum size for an image is 40X30.

4.10 Initial Values Window

Initial Values Window

This is the data-collection window for the Matrix Systems generator. Many of the variables entered here can be entered under a different name in the Parameter's window. Minx, maxx, miny and maxy are the spatial variables for framing the matrix object. These are usually updated automatically when you use the zoom box. Min Z and Max Z define the three-dimensional space that is used to map the matrix image. Normally Min Z is the negative of Max Z, but Min Z can be adjusted in the positive direction to shear off the front of the matrix object. This has the effect of exposing the insides of a matrix. Constants 1-4 are cx, cy, cz and cw respectively. Three rotate variables determine the 3-D angle of rotation. Steps and

Fine Tune are pitch adjustments that bear on the quality of the plot at an expense of lengthier calculations.

When you click on Okay or Apply, the matrix systems generator looks at the Fun#1 gadget in the Edit/Formula window. If this contains a preset variable (P0-K9) that function is iterated for its escape time, then the results are ray-traced in any 3-D object that may be created. If Fun#1 contains no preset variable, and the Type is set to 9, any custom formula (Fractint-style: as in 'z=z^3+c#') in the Formula box is used; else the original matrix Julia formula (q^2+c) is iterated and ray-traced. Not all functions may produce a usable 3-D object with this method, but it's interesting to experiment with.

The Plane variables B (Back), F (Front), and P (Position) allow you to flatten part of the matrix figure. For normal plotting, these variables default to 0. To have any effect on the image, the Back variable must be greater in value than the Front variable. The difference between back and front variables determines where the image is flattened. These variables are limited to +/-9.99, with normal values being in the range –z to +z. The position variable sets the color of the flattened plane. Note: if you set the Back variable less than -z, and the front variable greater than or equal to –z, you can get a gradient in the background, depending on the lighting and coloring settings. This sometimes has the effect of placing part of the matrix in a fog-like dimension.

Use the Random Rotate button to set random values (0-360) for the Rotation variables.

Select the "Use C as Column Vector" check box if you want the complex quad constant to be treated as a column vector in the calculations. In Julia sets, the constant doesn't change thoughout the iteration process, so this has the affect of changing the signs of the constant according to the Latin square, sign matrix and column selected in the Formula/Type window. For Julia sets you can also vary the constant's signs manually (and rearrange the C components) to obtain the same effect. Another value of this option is you can create different Mandelbrot sets by varying the Latin square, sign matrix and column. Mandelbrot images can be used as maps to find other Julia sets using the "G" hotkey command. See the hot keys section also for a description of the 'G' command.

4.11 Ray-Tracing Window

Ray-Tracing Window

The Light Point variables (lightx thru lightz) determine the direction of the light source used in the ray-tracing algorithm. The ViewPoint represents the angle at which the object is ray-traced, which can affect Phong highlights greatly. This has no effect on the camera view.

The Lighting variables shininess, highlight, gamma and ambient are used to adjust ambient light and highlights. The ranges for these variables appear beside their label. Decreasing the shininess value increases light reflected by the matrix image and the apparent sheen on the matrix image's surface. The ambient value controls the amount of ambient light that illuminates the matrix. The highlight value increases or decreases the specular (Phong) highlighting, while the gamma value increases or decreases the intensity of the light source's illumination. Once a plot is started, the lighting variables and light point can be changed without redrawing the matrix.

Click the Apply button to redisplay a plot after changing the lighting variables or light point. Click the Okay button to close the Ray-Tracing Window, applying new settings, if the variables were modified. Click on Cancel to revert to the state that existed when the ray-tracing window was opened. Click on Defaults to set the lighting and viewpoint variables to the built-in defaults for these variables.

4.12 Palette

Edit Palettes

Use the palette editor to modify the palette(s) in use.

It is important to realize that palettes are software-simulated in DynaMaSZ (since 24-bit color supports no hardware palettes), so color-cycling and palette switching are not fast operations as with a 256-color system that supports palettes.

There are copy and spread options to smooth or customize the existing palettes in DynaMaSZ. You can then save all the palettes in a .pl file, or by saving the entire function and bitmap (v1.08+ saves all the palettes in the data file.)

Colors are shown in 8 groups of 29 colors, with four colors on the last row. This makes it easy to create divide-by-8, divide-by-4 and divide-by-2 palettes with 232 colors (Use a start color of 0 and an end color of 231.) With DynaMaSZ, a palette is actually 60160 colors, with each succeeding color (except the last) followed by 255 colors that are evenly spread from one color to the next.

Use the RGB-slider controls to edit any color in the palette. Select Copy to copy any color to another spot in the palette. Select Spread to define a smooth spread of colors from the current spot to another spot in the palette. Copy and Spread take effect immediately when you select another spot with the mouse button. You can cancel the operation with the Cancel button. In DynaMaSZ, colors do not cycle smoothly when you adjust the RGB/HSV sliders. This would be too slow with true color. The Map button is used to map color changes to an image after you are done adjusting the sliders. In the HSV mode, color spreads are based on HSV values instead of RGB values, which in some cases results in brighter color spreads.

Right-click on any point on the main window and the palette color for that pixel will be displayed in the palette editor. You can use any of the color-cycling keys (after clicking on the main window) to see the effects of the cycling in the palette editor window. Note: color cycling and color-selection-from-pixel only works when the image has been drawn in the current session. If you load a pre-existing image file, you must redraw it to cycle colors, etc. Anti-aliasing, the composite figure option and hsv filtering also disable color cycling.

Use Reset to reset the colors of the palette in use, to where it was before it was cycled or modified. Note: if you change palettes with one of the function keys, any modifications to a previous palette are unaffected by the Reset button.

Use Reverse to reverse the order of the colors in the palette. This affects only those colors in the start-color to end-color range.

Use Neg to create a palette that is the complement of the current palette.

Use SRG to switch the red and green components of all palette colors.
Use SRB to switch the red and blue components of all palette colors. SRB and SRG are disabled in HSV mode. You can use these buttons to form eight different palettes by repeatedly switching red, green and blue components.

Use the Random palette button to randomize the current palette. The Randomize variables, rmin, rmax, bmin, bmax, gmin, and gmax act as limits that are applied after the palette after initial randomizing, to make the palette conform to the desired spectrum of colors.

Note: unless you click on Reset before exiting the editor, changes are permanent to the palette edited, no matter which way you close the editor (Okay button or close box.)

4.12.1 Neg Button

Neg button
Use Neg to create a palette that is the complement of the current palette.

4.12.2 Map Button

Map button
In DynaMaSZ, colors do not cycle smoothly when you adjust the RGB/HSV sliders. This would be too slow with true color. The Map button is used to map color changes to an image after you are done adjusting the sliders.

4.12.3 SRG Button

SRG button
Use SRG to switch the red and green components of all palette colors. RGB mode only.

4.12.4 SRB Button

SRB button
Use SRG to switch the red and blue components of all palette colors. RGB mode only.

4.12.5 Reset Button

Reset button
Use Reset to reset the colors of the palette in use, to where it was before it was cycled or modified. Note: if you change palettes with one of the function keys, any modifications to a previous palette are unaffected by the Reset button.

4.12.6 Red Slider

Red slider
Use the RGB/HSV-slider controls to edit any color in the palette.

4.12.7 Green Slider

Green slider
Use the RGB/HSV-slider controls to edit any color in the palette.

4.12.8 Blue Slider

Blue slider
Use the RGB/HSV-slider controls to edit any color in the palette.

4.12.9 Cancel Button

Cancel button
You can cancel a copy or spread operation with the Cancel button.

4.12.10 Okay Button

Okay button
Click on Okay to exit the palette editor, applying unmapped color changes to picture (if color-cycling is enabled.)

4.12.11 Spread Button

Spread button
Select Spread to define a smooth spread of colors from the current spot to another spot in the palette.

4.12.12 Copy Button

Copy button
Select Copy to copy any color to another spot in the palette.

4.12.13 H/R Button

H/R button
Change from HSV to RGB mode or back. In the HSV mode, color spreads are based on HSV values instead of RGB values, which in some cases results in brighter color spreads.

4.12.14 Rand Button

Random palette button
Use to create a random palette. Fast way to define palettes.

4.12.15 Red edit box

Red edit box
Shows red/hue value of selected color index.

4.12.16 Green edit box

Green edit box
Shows green/saturation value of selected color index.

4.12.17 Blue edit box

Blue edit box
Shows blue/value magnitude of selected color index.

4.12.18 Reverse button

Reverse button
Use Reverse to reverse the order of the colors in the palette. This affects only those colors in the start-color to end-color range. Useful for reversing divide-by-eight palettes, etc., for orbit-trap pictures that require a reversed palette.

4.13 Edit Text command

Text (Edit menu)

Allows you to edit text and font and apply it to a drawing. Select the font button to set the font style, size and color. In the text window click on Okay to add a line of text to the current image. (You can add multiple lines of text too, up to 80 characters.) The cursor will change to a crosshair. Position the cursor where you want the text to start and left-click the mouse. Note: font and title text are saved in the file "prefs.txt" in DynaMaSZ's startup directory. Title text can also be edited (as a single line only) in the Edit/Formula window.

4.14 Preferences

Preferences (Edit menu)
Each time you use the Reset command, DynaMaSZ restores data variables to built-in defaults. The Set Defaults button allows you to change some of the data variable defaults to whatever the current settings are. Some of the customizable variables include step, fine, formula, viewpoint, lighting, rotational angles, Phong and x/y space. (Iterations, Type, Constants, and a few other variables are excluded to maintain compatibility with the 'G' command.) The new Reset defaults are saved in the file "prefs.txt" when you close the program (if the Defaults check box is selected.) The check boxes in the group "Save on Program Close" allow you to change the default startup mode of a few Auto options, such as Auto Redraw, and the Random Setup variables. By keeping the boxes selected, DynaMaSZ saves the last changes you make to these options. If you want to go back to the initial settings (the way DynaMaSZ was packaged originally) you can click on the Reset Defaults button. This restores the data, Auto variables and random setup defaults.

5 Image menu

Image menu commands

The Image menu offers the following commands:

Draw	Draw the picture.
Draw Composite	Draw composite from figures 1-4.
Plot To File	Plot large bitmap images directly to png file.
Plot Files In Directory	Disk render .dmz files in working directory.
Auto Redraw	Redraw image on command.
Auto Clear	Clear drawing area before new plot.
Auto Dust	Remove isolated pixels and single-pixel threads.
Auto Sound Alert	Enable or turn off sound alerts.
Auto Remote	Open remote automatically at startup.
Auto Time	Show time used to plot image.
Merge Sum	Merge current pixel color with previous color summing colors.
Merge And	Merge current pixel color with previous color anding colors.
Merge Or	Merge current pixel color with previous color oring colors.
Merge High	Merge current pixel color with previous color by choosing highest rgb.
Merge Low	Merge current pixel color with previous color by choosing lowest rgb.
Merge Back	Merge current pixel color with previous color by excluding background color.
Merge Diff	Merge current pixel color with previous color by using difference of colors.
Hide Dialogs	Hide dialogs for active drawing.
Show Dialog	Show drawings for active drawing.
Abort	Abort drawing.
Continue	Continue drawing.
Zoom	Zoom into rectangle.
New View on Zoom	New view on zoom.
Clone	Clone current view.
Full Screen	View image full-screen.
Pilot	Use Pilot to rotate, pan and zoom.
Scan	Scan Mandelbrot border for matrix Julia set.
Dive	Peel off outer layer of matrix.
Ray Trace	Ray trace 3D plot.
Reset->	Reset coordinates, current figure or all figures
Figure 1	Switch to figure one.
Figure 2	Switch to figure two.
Figure 3	Switch to figure three.
Figure 4	Switch to figure four.
Composite	Select figures to merge.

5.1 Image Draw command

Draw command (Image menu)

Use this command to draw or redraw the image for the current fractal variables. Clicking inside the draw window with the left-mouse button stops all plotting. Use the Continue command to restart plotting from the current column.

5.2 Image Draw Composite command

Draw Composite command (Image menu)

Use this command to draw or redraw an image defined in the Composite command as a merging of figures 1-4. Clicking inside the draw window with the left-mouse button stops all plotting. Continue is disabled for this command.

5.3 Plot to file

Plot to File

Allows you to plot a large bitmap directly to a .png file without the added system requirements of keeping the whole bitmap in memory. The Target group sets the bitmap resolution (width 800 to 14400. Drawing aspect is that of the current image.) Click on Okay to set the target file name and start a new plot to file. Note: the 3200X2400 bitmap size is suitable for 8 1/2X11 printouts at 320-720 dpi. The larger bitmap sizes are suitable for poster-size printouts. This option is not available with the merging options, or with anti-aliasing. Also, solid-guessing is disabled when using this option.

5.4 Plot Files in Directory

Plot Files in Directory

Allows you to plot a set of large bitmaps directly to a .png files without the added system requirements of keeping any of the images in memory. The Target group sets the bitmap resolution (width 800 to 14400. Drawing aspect is that of the current image.) All data files (.dmz) in the working directory are enlarged to this resolution. Click on Okay to start. Note: the 3200X2400 bitmap size is suitable for 8 1/2X11 printouts at 320-720 dpi. The larger bitmap sizes are suitable for poster-size printouts. Merging, anti-aliasing and solid-guessing are disabled when using this option.

5.5 Image Redraw command

Auto Redraw command (Image menu)

With this command disabled (on by default), redraw does not occur except when the Draw command is executed, or Continue. Most of the time you want to see the results of changing a parameter or mapping option, so redraw occurs automatically with parameter or mapping changes. Sometimes you want to change more than one parameter before redrawing the image. So you need to turn this option off then.

5.6 Image Auto Clear command

Auto Clear command (Image menu)

With this command enabled (on by default), the drawing area is cleared before starting a new plot. You can turn off this option when you want to see the effect of minor changes to parameters, as they affect the plot pixel by pixel, or when setting up a multiple-layered fractal. Note: when you disable auto clear, no pre-image is drawn while the image is being calculated, and any background image (loaded through Open [Jpeg] or Open [Png]) is retained. You can use the shift-c command (hot keys) to clear the drawing area at any time.

5.7 Image Auto Dust command

Auto Dust command (Image menu)

With this command enabled (on by default), single pixels and matrix threads one pixel wide are removed after ray-tracing. Sort of like the de-speckle filter in some image-processing programs.

5.8 Image Auto Alert command

Auto Sound Alert command (Image menu)

With this command enabled (on by default), the user is notified by a sound clip when a drawing is completed or user-canceled. By disabling this command the completion exclamation is suppressed and also any alert that contains a message box. Note: some sound clips are automatically generated by Windows, or there is no text alert for a given error condition. In these cases the sound alert is unaffected by the Auto Alert command.

5.9 Image Auto Remote command

Auto Remote command (Image menu)

With this command enabled (on by default), the remote is opened immediately at program startup. Handy if you find the remote useful and don't want to click on the toolbar button each time the program starts up.

5.10 Image Auto Time command

Auto Time command (Image menu)

With this command enabled (on by default), the time that an image takes to plot is displayed when the plot is complete. DynaMaSZ saves the condition of this option at session's end, so if you disable it and close the program, the option will be disabled when you restart DynaMaSZ.

5.11 Image Merge Sum command

Merge Sum command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using a summing algorithm. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.12 Image Merge And command

Merge And command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using an anding algorithm. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.13 Image Merge Or command

Merge Or command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using an oring algorithm. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.14 Image Merge High command

Merge High command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using the highest rgb values of both images. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.15 Image Merge Low command

Merge Low command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using the lowest rgb values of both images. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.16 Image Merge Back command

Merge Back command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using the rgb components of the new image if the new color index is not zero; else the old rgb values are retained. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.17 Image Merge Diff command

Merge Diff command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using the difference of the rgb values of both images. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.18 Image Hide dialogs command

Hide dialogs command (Image menu)

Use this command to hide all open non-modal dialogs in the active window. This helps to de-clutter the screen and avoid confusion if two or more draw windows are open. Each draw window has its own dialogs. Disabled if no dialogs are open.

5.19 Image Show dialogs command

Show dialogs command (Image menu)

Use this command to show all open non-modal dialogs in the active window. This command restores any dialogs that may have been hidden by the Hide dialogs command. Each draw window has its own dialogs. Disabled if no dialogs are open or hidden. Note: you can restore individual dialogs by selecting the command that opened them originally.

5.20 Image Abort command

Abort command (Image menu)

Use this command to stop drawing. Clicking inside a window's drawing area or close box (or the program close box) will also stop the drawing. Note: once a plot has started DynaMaSZ continues to draw the image for that window regardless of which drawing window has the input focus, until done or aborted. You can open and close other drawing windows without affecting the current drawing, but only one drawing is active at any time.

5.21 Continue Draw

Continue Draw (Image menu)

Continues a plot that was aborted early. The plot is restarted at the beginning of the last row drawn. Continue is disabled when an Image/Merge option is selected. Continue isn't available for 3-D images that have been reloaded from a previous drawing session.

5.22 Zoom

Zoom (Image menu)

Turns on zoom mode, so that detail of the current plot may be magnified. Alternatively, just click inside any drawing window, move the mouse, and the zoom box will appear. Using the mouse, move the zoom box over the portion of the plot you wish to magnify. Hold the left mouse button to shrink the box or the right button to enlarge it. Use the up and down arrow keys to squash or expand the box, changing the aspect of the image. Use the Shift key to enlarge the zoom box X4 for quickly zooming outward. Use the Ctrl key to shrink the zoom box by 4. You start a zoom by pressing the space bar. You abort a zoom by clicking outside the main window or in the title bar, or by pressing the escape key. The program will begin a new plot at the new coordinates. You may zoom in by defining a box inside the current drawing area. You zoom out by drawing a box outside the current drawing area. The outer zoom limits are between -1000 and 1000. The precision is that of double precision (64 bits)

5.23 Image New View on Zoom command

New view on zoom (Image menu)

With this option enabled, a new window is opened with each zoom, instead of the zoom box area replacing the original image. Ignored in avi mode.

5.24 Image Clone

Clone (Image menu)

A new draw window is opened that contains the same fractal data as the window it was opened from. This is useful for comparing minor changes in texturing options, etc. Similar to using the copy/paste data commands except that all figures are copied to the new view.

5.25 Full Screen

Full Screen (Image menu)

Displays the entire plot, expanding or shrinking the image to fit in a maximized window without title bar, scroll bars or menu bar. At all other times, part of the picture is hidden by the inclusion of the title bar, toolbar, scroll bars and menu bar. To exit full-screen mode, press any key or click the left-mouse button.

5.26 Pilot

Pilot (Image menu)

Opens the Pilot window to adjust key parameters, rotate, zoom and redraw the figure interactively. The current image is reduced to one quarter normal for faster redraw. Each click on a Pilot button increments or decrements a parameter. The Speed slider controls the rate at which the buttons operate (default is 10.)

Press the space bar or Click on Ok to open a new window and draw the altered image full-size. Press Esc or click on Cancel to exit this mode without opening a new window. Note: when using this option while an AVI stream is open, a new window isn't opened, but the altered figure is drawn in the current draw window, the changed parameters replacing the previous ones.

5.27 Scan

Scan (Image menu)

Equivalent to the Shift+G hot-key. Enabled when the Type is Mandelbrot. A matrix Julia set is generated in sector 2, using an iteration count of 10 and other parameters are changed temporarily to suit matrix plots. (Z is set to 2.0, the rotational variables are reset and the light source is set to the default, if random lighting is enabled.). Once you find an interesting matrix set using "G", like the J command another window is opened that sets the fractal parameters to those in the exploratory qjulia window. The parameters in the exploratory window revert to their original Mandelbrot settings.

5.28 Dive

Dive (Image menu)

Select Dive to go beneath the surface of a matrix image. Some matrix images have a smooth border that doesn't show the turbulence below the surface. Using the Dive option strips off the border layer to reveal what's underneath.

5.29 Ray Trace

Ray Trace

Uses a ray-tracing algorithm to add a light source to 3-D plots. Color palettes should be continuous (dark to light to dark) to take best advantage of this option. The light source parameters may be altered in the Edit Ray-Tracing Variables window. This option is the default. Note: if you prefer to generate matrix images without ray-tracing, deselect this flag before drawing the plot. The image draw will be the pre-image, using the existing palette, as is, without ray-tracing (Phong and shading.)

5.30 Reset

Reset

Reset the current figure or all figures to an empty Mandelbrot. All functions in the New Formula data are blanked. All options on the Flags menu are reset to their default settings. The Print Function Data ignores any reset figures.

The Ranges Only command resets only the real Z and imaginary Z ranges in the Parameters window (to +/-2.0 and +/-1.5.) No other menus or variables are affected. This is useful in conjunction with the "P" command to generate and view Julia sets. After setting the complex-C variable via shift-P (Caps Lock off), you need to reset the Z ranges to see the entire Julia set after zooming into a Mandelbrot set. The Reset All option resets all figures.

5.31 Figure #1

Figure #1

Switch to Function #1. Current settings are saved for the previous image.

5.32 Figure #2

Figure #2

Switch to Function #2. Current settings are saved for the previous image.

5.33 Figure #3

Figure #3

Switch to Function #3. Current settings are saved for the previous image.

5.34 Figure #4

Figure #4

Switch to Function #4. Current settings are saved for the previous image.

5.35 Image Composite command

Composite command (Image menu)

Opens the Composite Figure window, where you can define a set of figures to merge into one image. All the merging options in the Merge Color menu are supported, plus "ALL" which is usually used for the first figure to be drawn. The "ALL" option transfers all rgb information for a figure to the drawing area, without checking the rgb state of the pixel. You can define up to four figures (layers), as part of the composite, but each figure should contain an image (if used in the composite.)

6 Type menu

Type menu commands

The Type menu offers the following commands:

Mandelbrot0	Mandelbrot set (orbit starts at zero.)
MandelbrotP	Mandelbrot set (orbit starts at pixel.)
Julia	Julia set.

6.1 Mandelbrot0

Mandelbrot0

Mandelbrots base their mapping on varying inputs of complex C, which corresponds to the min/max values set in the Parameters window. With MandelbrotP, the initial value of Z is set to zero.

6.2 MandelbrotP

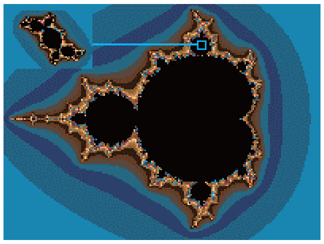
MandelbrotP

Mandelbrots base their mapping on varying inputs of complex C, which corresponds to the min/max values set in the Parameters window. With MandelbrotP, the initial value of Z is set to the value of the pixel being iterated. Usually used with cubic Mandelbrot formulas.

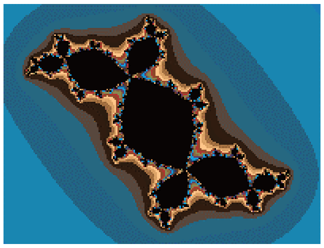
6.3 Julia

Julia

Julia sets normally have a fixed complex C, with varying inputs of Z, which corresponds to the min/max values set in the Parameters window. This option, without the Bound flag set, generates the so-called 'filled-in' Julia set, which includes non-escaping points as well as the Julia set.



Julia from Mandelbrot



Julia set

7 Render menu

Render menu commands

The Render menu offers the following commands:

Coloring Filter	Define coloring filter.
Surface Filter	Define surface filter.
X Mapping Filter	Use bitmap file for matrix rendering: X mapping.
Y Mapping	Use bitmap file for matrix rendering: Y mapping.
Z Mapping	Use bitmap file for matrix rendering: Z mapping.
Remap	Remap bitmap for matrix rendering.
Anti-Alias	Use anti-aliasing, with 1X4 or 1X2 super-sampling.
Link Coloring To Pixel	Set coloring to match absolute coordinates of image.
Atan Coloring	Use Atan algorithm for coloring.
Bof60 Coloring	Use Bof60 algorithm for coloring.
Potential Coloring	Color by magnitude of z.
Filter	Choose an optional tail-end filter.
Orbit Traps	Set orbit trapping method.
Add Noise	Add noise to coloring.
Factors	Edit noise factors.
Reset Noise Seed	Re-seed random noise generator.
Texture Scale	Set scaling factor for texture.
Apply (Generalized Coloring)	Apply non-palette based coloring method.
(Blend submenu)	
linear scale	linear scale color blending.
average	average color blending.
subtractive	subtractive color blending.
sum of squares 1	sum #1 color blending.
sum of squares 2	sum #2 color blending.
sin #1	sine color blending.
atan #1	atan #1 color blending.
additive	additive color blending.
log	log color blending.
atan #2	atan #2 color blending.
atan #3	atan #3 color blending.
sin #2	sine #2 color blending.
sin #3	sine #3 color blending.

atan #4	atan #4 color blending.
fractal dimension	fractal dimension color blending.
Color Parameters	Color controls.
RGB	RGB color mapping.
RBG	RBG color mapping.
GRB	GRB color mapping.
GBR	GBR color mapping.
BRG	BRG color mapping.
BGR	BGR color mapping.
Triangle Algorithm	Handle color overflow by triangle algorithm.
Sine Algorithm	Handle color overflow by sine algorithm.
Gray Scale	Gray scale mapping.
Invert	Invert colors.
Fractal Dimension	Apply fractal dimension algorithm to current blend option.
Apply ...	Apply external rendering function....
Process Bailout	Process bailout from external rendering function.
Rendering Libraries Off	Disable external rendering libraries.

7.1 Coloring Filter

Coloring Filter

Here you define a coloring filter based on a real function. A generalization of Earl Hinrichs' sine-wave coloring method, the function can be any formula, up to 80 characters, that uses the z-buffer variable and framing variables x and y. Sample function: .1*sin(z)+cos(x*x). The Magnify slider is used to control the intensity of the filter. Click on Apply to apply a new coloring formula without closing the window. Click on Okay to close the window and apply changes. Click on Cancel to close the window without applying changes. Use the Random Filter button to generate a random coloring filter. The best filters will use the z value and one of the other variables (x or y.)

Matrix images normally use palette index one (the second index, zero being reserved for the background color) for their predominant color, with pixel intensities/colors affected by the lighting variables. When the coloring filter formula is defined, up to 235 colors can be used (the full palette) to create mixed textures.

The trig and exponential functions translated include sine (sin), arc sine (asn), cosine (cos), arc cosine (acs), tangent (tan), hyperbolic tangent (th), hyperbolic sine (sh), hyperbolic cosine (ch), log (log), natural log (ln), power (pow), arc tangent (atn), absolute value (abs), exponential (exp) and square root (sqr.)

The math functions are *(multiply),-(subtract), /(divide), and +(add).

The constants are PI and E (ln (1)), plus any floating-point number up to 9 digits (including the decimal point).

The power function (x to the y power) is entered in standard notation: x^y', with optional parenthesis necessary around complex exponents or variables.

Note: Range limits exist for arguments to these functions: exp, arc sine, hyperbolic sine, arc cosine, hyperbolic cosine, arc tangent, and hyperbolic tangent (+/-100.0 for the exponential, +/-200.0 for hyperbolic functions, +/-1.0 for the arc functions), the log functions (must be >0) and the power function (x must be integral and non-zero when y<0, and 0^0 is undefined). Square root is undefined for x<0. No filtering is done when these limits are exceeded.

Syntax for an acceptable formula is AS([XY])+bs([xy])...
.up to 80 characters per formula. Algebraic notation is supported to a limited degree. E.G. you can enter a variable as 2x^2, instead of 2*x*x.
A and B are optional constants.

S is an optional trig function (1 to three letters: 1 will work for sine, cosine and tangent, but use the above abbreviations for the other functions. X and Y are the standard variables. The '+' could be any of the math functions.

The parser interprets up to 10 levels of parenthesis. Use parenthesis to separate complex expressions. Use parenthesis to embed trig functions within other trig functions, etc.

7.2 Surface Filter

Surface Filter

Here you define a surface filter based on a real function. Like a coloring filter, except that the formula is used to warp the matrix image's shape. The Magnify variable is used to control the intensity of the filter. Click on Apply to apply a new surface filter without closing the window. Click on Okay to close the window and apply changes. Click on Cancel to close the window without applying changes. Use the Random Filter button to generate a random surface filter. The best surface filters will use the z value and one or both of the other variables (x or y.)

7.3 X mapping

X mapping

This option allows you to use a separate bitmap or picture to color a 3-D image. Enabled when a bitmap has been copied to the clipboard, each pixel of the image is replaced with a corresponding color in the clipboard image, depending on the mapping option used (x, y or z mapping.) This produces textured effects that add realism to the image. Notes: the internal structure of the 3-D image is used to determine each mapping algorithm, so the overall effect is to texture the image like a wood-grain rather than a decal. When you first enable this option, whatever is in the clipboard gets copied to a buffer file for rendering. To change the picture in the buffer, you need to change the clipboard image then use Remap to reinitialize the buffer. The clipboard image isn't saved with the data file, so you need to remember which bitmap file is used for the mapping, to redo a fractal like this later (you can sometimes leave a comment in the Fun#2 edit box in the Edit/Formula window for this purpose.).

7.4 Y mapping

Y mapping

This option allows you to use a separate bitmap or picture to color a 3-D image. Enabled when a bitmap has been copied to the clipboard, each pixel of the image is replaced with a corresponding color in the clipboard image, depending on the mapping option used (x, y or z mapping.) This produces textured effects that add realism to the image. Notes: the internal structure of the 3-D image is used to determine each mapping algorithm, so the overall effect is to texture the image like a wood-grain rather than a decal. When you first enable this option, whatever is in the clipboard gets copied to a buffer file for rendering. To change the picture in the buffer, you need to change the clipboard image then use Remap to reinitialize the buffer. The clipboard image isn't saved with the data file, so you need to remember which bitmap file is used for the mapping, to redo a fractal like this later (you can sometimes leave a comment in the Fun#2 edit box in the Edit/Formula window for this purpose.).

7.5 Z mapping

Z mapping

This option allows you to use a separate bitmap or picture to color a 3-D image. Enabled when a bitmap has been copied to the clipboard, each pixel of the image is replaced with a corresponding color in the clipboard image, depending on the mapping option used (x, y or z mapping.) This produces textured effects that add realism to the image. Notes: the internal structure of the 3-D image is used to determine each mapping algorithm, so the overall effect is to texture the image like a wood-grain rather than a decal. When you first enable this option, whatever is in the clipboard gets copied to a buffer file for rendering. To change the picture in the buffer, you need to change the clipboard image then use Remap to reinitialize the buffer. The clipboard image isn't saved with the data file, so you need to remember which bitmap file is used for the mapping, to redo a fractal like this later (you can sometimes leave a comment in the Fun#2 edit box in the Edit/Formula window for this purpose.).

7.6 Remap

Remap

Allows you to change the mapping buffer, after initializing an image with X, Y or Z mapping. To change the picture in the buffer, you need to change the clipboard image then use Remap to reinitialize the buffer.

7.7 Anti-Alias

Anti-Alias

Applies a 2 to 1 or 4 to 1 averaging filter to every pixel plotted, to reduce jaggies and other high-frequency noise. This increases the processing time 4 to 8 times, so is mainly a final rendering method, not for general development use. Only single-pass mode is supported -- or without ray-tracing, solid-guessing can be used), plot-to-file. Note: because of the lengthy time required for applying the anti-aliasing filter, and because anti-aliasing calculates different smoothing colors each time the palette is changed, all palette-switching hot keys and the light bulb button are disabled with the anti-alias flag set.

7.8 Link Coloring To Pixel

Link Coloring To Pixel

Set coloring to match absolute coordinates of image. This uses extra buffers to track a figure's texture, so that when you rotate it, the texture moves with the figure. Used with the Atan Coloring, Bof60 Coloring, Potential Coloring, Filter, Orbit Traps and the Noise functions. Due the extra memory required for this command, you won't be able to open an image much larger than 1600X1200, unless you have more than 128MB of system memory.

7.9 Atan Coloring

Atan Coloring

Uses an Atan algorithm by David Makin to color an image.

7.10 Bof60 Coloring

Bof60 Coloring

A variation of the Bof60 algorithm found in the classic Pietgen/Ricter text, The Beauty of Fractals, adapted by David Makin to color a 3-D image.

7.11 Potential Coloring

Potential Coloring

The magnitude of z is used to color the image.

7.12 Filter

Filter

Based on Stephen C. Ferguson's filter algorithms in his program Iterations, this option allows you to choose one of 26 tail-end filters for surface rendering. Corresponds roughly to its effect on the basic Mandelbrot-squared set. The effect will vary with the formula and fractal type chosen.

The Magnify variable is used to intensify or de-intensify the effect of the filter. This value can range from 1-500 nominally. Click on Apply to apply a new filter without closing the window. Click on Okay to close the window and apply changes. Click on Cancel to close the window without applying changes.

7.13 Orbit traps

Orbit Traps

Orbit traps are used with the "Link Coloring to Pixel" option.

This includes methods that trap the orbit of a point if it comes in range of a pre-specified area or areas.

The Epsilon-Cross method colors points only if the absolute value of Z-real or Z-imaginary is less than or equal to Epsilon (a small value.) Other points are mapped at the time they blow up (exceed the zlimit.) This produces hair-like structures that branch wildly from the complex set boundaries.

The Globe method uses a circular area around the origin to map a point's orbits. This produces sphere-like structures.

The Ring method uses an area formed by two circles around the origin to map a point's orbits. This produces ring-like structures.

The Four-Circles method (Paul Carlson) uses four circular areas to map a point's orbit. This produces sphere-like structures.

The Square method uses an area formed by two squares around the origin to map a point's orbits. This produces ring-like structures with right angles.

The Petal method (Paul Carlson) also uses four trap areas to form flower-like patterns.

7.14 Orbit trap values

Orbit Trap Values

Enter a value for Epsilon and Epsilon2, which are used to define the size of the orbit trap areas (.001-2.0 and 0.0-epsilon.) The exclude box is used to exclude the first # iterations (0-99) from orbit trapping.

Click on Apply to apply changes without closing the window. Click on Okay to close the window and apply changes, if any. Click on Cancel to exit the window without changing parameters.

Epsilon2 is used to create windows into the stalks. The default value is 0.0, which produces solid stalks. Epsilon2 has no effect on the Petal method.

7.15 Add Noise

Add Noise

Add noise to image texture. A variation of Perlin's noise algorithm is used to add natural randomness to an image's coloring.

7.16 Factors

Factors

Edit noise factors. The Blend variable determines how much noise is added to an image. The higher the blend, the more pronounced the noise appears. This also tends to darken an image, which can be compensated for by decreasing Gamma. The Grain variable determines the frequency of the noise. The higher the grain, the noisier the image appears. You can adjust how the noise maps to an image by changing the scale factors. Higher scale factors make the image noisier on the respective axis (x, y and z.) Additional variables affect the type and shaping of the noise data: Gaussian is an alternate form of noise, while Planet, Check, Tooth, Barber and Wood apply a specific envelope to the noise. The Marble variable is used to introduce a low frequency or high frequency modulation on top of the noise. You can achieve marble-like textures by combining a high frequency marble value with a low frequency Blend value. The marble variable also adds a high-frequency bump map to the wood envelope.

The Surface Warp variable allows you to apply the same noise to a matrix figure's shape also, like a surface filter. Small values are best for creating realistic surface variations, like stone and wood grain.

7.17 Reset Noise Seed

Reset Noise Seed

The random noise generator is re-seeded. Use this to create variations on the noise texture.

7.18 Generalized coloring

7.18.1 Apply command

Apply command (Render menu)

Use this command to switch to Steve Ferguson's generalized coloring mode. Images are colored via the selection in the Blend submenu and color-controls dialog, instead of the palette-based coloring filters. To switch back to palette mode, apply a coloring filter or one of the palette-based textures, such as Atan, Potential or one of the orbit traps.

7.18.2 Color Parameters command

Color Parameters command (Render menu)

Use this command to adjust the color controls when in the generalized coloring mode.

7.18.3 Blend

7.18.3.1 Blend 00 command

linear scale command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.2 Blend 01 command

average command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.3 Blend 02 command

subtractive command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.4 Blend 03 command

sum of squares 1 command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.5 Blend 04 command

sum of squares 2 command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.6 Blend 05 command

sin #1 command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.7 Blend 06 command

atan #1 command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.8 Blend 07 command

additive command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.9 Blend 08 command

log command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.10 Blend 09 command

atan #2 (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.11 Blend 10 command

atan #3 (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.12 Blend 11 command

sin #2 command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.13 Blend 12 command

sin #3 command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.14 Blend 13 command

atan #4 command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels.

7.18.3.15 fractal dimension command

fractal dimension command (Blend submenu)

Color blending for generalized coloring mode. The selected formula is applied while mapping colors to pixels. Uses an early version of Steve Ferguson's fractal dimension algorithm.

7.18.4 Color RGB command

RGB command (Render menu)

Use this command to use red/green/blue mapping, if in the generalized coloring mode.

7.18.5 Color RBG command

RBG command (Render menu)

Use this command to use red/blue/green mapping, if in the generalized coloring mode.

7.18.6 Color GRB command

GRB command (Render menu)

Use this command to use green/red/blue mapping, if in the generalized coloring mode.

7.18.7 Color GBR command

GBR command (Render menu)

Use this command to use green/blue/red mapping, if in the generalized coloring mode.

7.18.8 Color BRG command

BRG command (Render menu)

Use this command to use blue/red/green mapping, if in the generalized coloring mode.

7.18.9 Color BGR command

BGR command (Render menu)

Use this command to use blue/green/red mapping, if in the generalized coloring mode.

7.18.10 Color Triangle algorithm command

Triangle Algorithm command (Render menu)

When color values exceed the range of rgb components or palette indexes, the values are scaled with a triangle algorithm, or linear ramp (non-palette mode only.)

7.18.11 Color Sine algorithm command

Sine Algorithm command (Render menu)

When color values exceed the range of rgb components, the values are scaled with Steven C. Ferguson's sine algorithm (non-palette mode only.)

7.18.12 Color Gray Scale command

Gray Scale command (Render menu)

Use this command to color the active image with gray tones, if in the generalized coloring mode.

7.18.13 Color Invert command

Invert command (Render menu)

Use this command to invert image colors, if in the generalized coloring mode.

7.18.14 Fractal Dimension command

Fractal Dimension command (Render menu)

Generalized fractal dimension algorithm (S. Ferguson), for use with any blend option (Blend 14 is retained for compatibility with previous QuaSZ versions.)

7.18.15 Rendering Library functions (rll)

7.18.15.1 Apply command

Apply (external rendering function) command (Render menu)

Use one of the TieraZon2-compatible-plug-in rendering methods. To use, this option automatically enables generalized coloring mode. There is a large potpourri of rendering methods (280+) in the TieraZon/Dofo-Zon Elite libraries that you can experiment with. Some have their own bailout routines that can distort the basic matrix shape in interesting ways. For version 1.03 plug-ins, each method can have its own dedicated configuration box, containing as many variables as is needed to customize the function. Pre-1.03 plug-ins use a fixed configuration box, with only a scaling and orbit-trap variable. Note: selecting one of the palette-based rendering methods in the Render menu, such as Atan or Potential Coloring, or selecting a 2-D fractal Type, will disable and deallocate any external rendering function in use.

7.18.15.2 Process Bailout command

Process Bailout command (Render menu)

Deselect this option if you don't want to use an external rendering function's bailout routine. The bailout routine can sometimes distort a matrix shape in undesirable ways, or you may like the coloring effect of the rendering method by itself.

7.18.15.3 Rendering libraries off command

Rendering Libraries off command (Render menu)

Disable and deallocate the external rendering function selected.

7.19 Texture Scale

Texture Scale

Opens a window to edit texture scale factors. The higher the scale factors, the more repetitive the texture becomes. You can adjust the factors to make the texture asymmetrical on the x, y or z-axis. Scale A is used to adjust the texture scale for the atan and Bof60 coloring options. Click on Apply to apply changes without closing the window. Click on Okay to close the window and apply changes. Click on Cancel to close the window without applying changes.

8 Pixel menu

Pixel menu commands

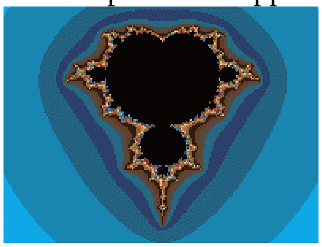
The Pixel menu offers the following commands:

Phoenix	Phoenix orientation.
Invert	Invert image around circle.
Invert Off	Reset inversion flag.
Symmetry->	Horizontal, vertical or XY symmetry.
Switch Z For C	Switch z for c.
Solid-Guessing	Solid-guessing plotting mode.
Cliff's Slice	Use C. Pickover's 'CPCB' slice for hypercomplex planes.

8.1 Phoenix

Phoenix orientation

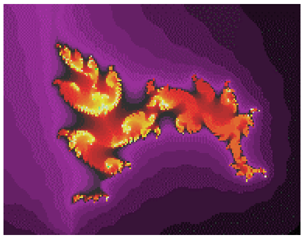
The Phoenix flag rotates the planes, so that the imaginary plane is mapped horizontally and the real plane is mapped vertically.



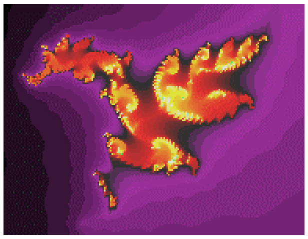
8.2 Invert

Invert

The Invert flag inverts the plane around a circle. A window is opened that allows the user to specify the circle's radius and center coordinates. Select Auto Coords to let DMZ calculate the center coordinates and circle radius. Using Auto Coords, the new radius and center coordinates are calculated when the picture is next drawn. You can zoom on an inverted picture as long as radius and center coordinates remain the same. Use the Perspective box to alter the X/Y symmetry of the inversion. A smaller Perspective value (less than 1.0) stretches the inversion in the vertical direction.



Original picture



Inverted

8.3 Invert Off

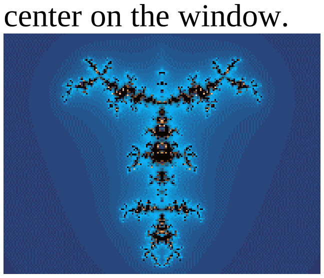
Invert Off

Turns off the inversion flag. Alternatively you can set the inversion radius to 0.0 to turn off inversion.

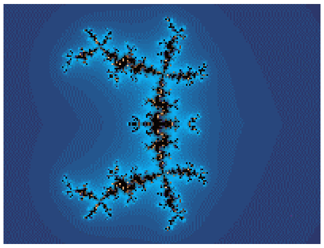
8.4 Symmetry

Symmetry

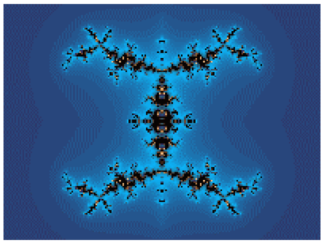
This produces a mirror image from left to right (vertical) or top to bottom (horizontal) or both (xy). You can zoom with symmetry, but the results will be uncertain if the zoom box is off-center on the window.



Vertical symmetry



Horizontal symmetry



XY symmetry

8.5 Switch

Switch Z For C

When a Switch flag is set, you have switch Z for C. When Z is switched for C, normally you get Mandelbrots from Julia sets and vice versa.

8.6 Solid Guessing

Solid Guessing

In the solid-guessing plotting mode, the program guesses at colors that lie inside rectangular areas of the plot. It first computes all the perimeter pixels of a rectangle, and checks if all the pixels have the same color. If so, all the pixels inside the rectangle are colored the same and no further calculations are done on that rectangle. Otherwise the rectangle is broken into four parts and the above procedure is repeated for each part. If any of the perimeter pixels are different at this point, all the remaining pixels in the smaller rectangle are computed. The screen is updated in groups of 16 lines.

8.7 Cliff's Slice

Cliff's Slice

With this option set, the 4-D z-planes are rotated to match C. Pickover's quaternion examples in "Computers, Pattern, Chaos and Beauty.

9 View menu

View menu commands

The View menu offers the following commands:

Toolbar	Shows or hides the toolbar.
Status Bar	Shows or hides the status bar.

9.1 View Toolbar command

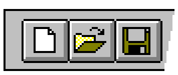
Toolbar command (View menu)

Use this command to display and hide the Toolbar, which includes buttons for some of the most common commands in DynaMaSZ, such as File Open. A check mark appears next to the menu item when the Toolbar is displayed.

See Toolbar for help on using the toolbar.

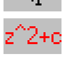
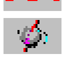
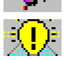
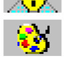
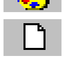
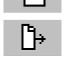
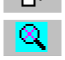
9.1.1 toolbar

Toolbar



The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in DynaMaSZ,

To hide or display the Toolbar, choose Toolbar from the View menu (ALT, V, T).

Click	To
	Open a new drawing, based on the default template.
	Open an existing drawing. DynaMaSZ displays the Open dialog box, in which you can locate and open the desired file.
	Save the active drawing or template with a new name. DynaMaSZ displays the Save As dialog box.
	Set image size.
	Edit formula/type data.
	Edit fractal parameters.
	Edit ray-tracing variables.
	Edit palette.
	Draw image from current parameters.
	Continue drawing.
	Zoom into rectangle.

	Show picture full-screen.
	Reset coordinates.
	Draw Mandelbrot set
	Draw Julia set
	Display info about DynaMaSZ.
	Display DynaMaSZ's help index.

9.2 View Status Bar Command

Status Bar command (View menu)

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed.

See Status Bar for help on using the status bar.

9.2.1 status bar

Status Bar

The status bar is displayed at the bottom of the DynaMaSZ window. To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them. If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right areas of the status bar indicate which of the following keys are latched down:

Indicator	Description
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.

10 Window menu

Window menu commands

The Window menu offers the following commands, which enable you to arrange multiple images in the application window:

Cascade	Arranges windows in an overlapped fashion.

Tile	Arranges windows in non-overlapped tiles.
Arrange Icons	Arranges icons of closed windows.
Size Desktop	Size drawing area to window frame.
Window 1, 2, ...	Goes to specified window.

10.1 Cascade

Cascade command (Window menu)

Use this command to arrange multiple opened windows in an overlapped fashion.

10.2 Tile

Tile command (Window menu)

Use this command to arrange multiple opened windows in a non-overlapped fashion.

10.3 Arrange Icons

Window Arrange Icons Command

Use this command to arrange the icons for minimized windows at the bottom of the main window. If there is an open drawing window at the bottom of the main window, then some or all of the icons may not be visible because they will be underneath this drawing window.

10.4 Size DeskTop

Window Size DeskTop Command

Use this command to size the active drawing window to its frame size. Use after Tile command to reduce white space around a drawing that is smaller than screen size.

10.5 1, 2, ...

1, 2, ... command (Window menu)

DynaMaSZ displays a list of currently open drawing windows at the bottom of the Window menu. A check mark appears in front of the drawing name of the active window. Choose a drawing from this list to make its window active.

11 Video menu

Video menu commands

The Audio/Video menu offers the following commands:

Open AVI Stream	Open AVI file for writing and draw initial frame.
Write Frames	Write frames to AVI file.
Close AVI Stream	Close an existing AVI stream.
View AVI	View an AVI animation file.
AVI Composite	Generate composite video.
AVI Object	Output video frames as obj file.
AVI Wrl	Output video frames as wrl files.
AVI Lfm	Output video frames as lightflow mesh files.

11.1 Open Avi Stream

Open Avi Stream…

Through a series of windows, this allows you to name and open an avi animation stream , select the frames per second, and choose a compression method. After using the file requester to name the file, you are given a choice of compression methods. The compression methods include Intel Indeo Video®, Microsoft Video 1 and Cinepak Codec by Radius. You can also choose no compression for optimum view quality. (All compression methods degrade the original images, some more than others.) The first key frame in the stream is then drawn and written to the file.

Notes: after the stream is opened, the size of the fractal that can be drawn is fixed at the size of the frame. No changes can be made to the size until the stream is closed. If you open a video stream after setting up a batch mode (Demo menu), then the frames will be written as a series of bmp, obj, wrl or lfm files, depending on whether AVI Object , AVI WRL or AVI Lfm is also checked.

11.2 Write Frames

Write Frames…

With this option, frames are written to a stream based on the difference between the current key frame and the previous key frame. The first key frame is written when you open a stream. The next key frame is created each time you use this option. In between you can zoom or change Avi variables as much as necessary. The stream is only written to when this option is used. The last key frame is automatically saved after the 'tween' series is written. The number of frames may range from 1-1500 frames between keys. With a frame number of 1 only the key frames are written. This allows animation to be created that incorporates all scalable variables in DynaMaSZ.

Use the Cancel button to exit this dialog without initializing a new series of frames.

Check the Log Scaling box if you want the frames to be written with logarithmic space between frames, else linear space is used. Useful when zooming, where frames would otherwise be packed together at the end of the frame series.

Notes: key frames are saved in parameter files (dmz), with filenames of "bvf_image#_title.dmz", where '#' is the number of the keyframe and 'title' is the name of the working fractal file. If you open a video stream after setting up a batch mode (Demo menu), then the frames will be written as a series of bmp, obj, wrl or lfm files, depending on whether AVI Object, AVI WRL or AVI Lfm is also checked.

11.3 Close Avi Stream

Close Avi Stream
Closes any open avi stream file. You need to do this before viewing the file or creating a new avi file. The stream is also closed when you exit DynaMaSZ.

11.4 View Avi

View Avi…
Opens an avi file for viewing. You can preview any multimedia file by clicking on its file name. A multimedia box will appear to the right of the file list. Click on okay to open the main view window.

There are buttons to Play a file forwards or Backwards, or forward automatically with Auto rewind/repeat. Click on Slow to slow down a video. Each click on Slow halves the viewing speed. A click on Stop freezes viewing and restores the view speed to normal playback.

Use the Open button to view a different avi file. Use the Save button to save the file in a different compression format. You must use a different name to save the file than the name that was used to open it. Click on the left-mouse button or any key to abort a save operation.

Note: the view avi requester can be used to preview any multimedia file, including midi files.

11.5 Avi Composite

AVI Composite
When this flag is set, DynaMaSZ generates composite frames for a video according to the settings in the Image/Composite window. Each frame may then consist of a merging of up to 4 figures (1-4). You must set this flag and the composite options before beginning a video. After an avi stream has been opened, you can then use variations of any figure in the composite to produce tweens while using the Write Frames option. As usual, you vary data in the figure(s) before writing frames.

11.6 AVI Object

AVI Object
When this flag is set, DynaMaSZ generates single frames in obj format instead of opening a video stream. The 3-D object files can be exported into a program such as Bryce, for post-processing into videos. This works in conjunction with the Demo/Batch mode command, to set the target disk directory and file name, so is only enabled when Batch mode is set.

11.7 AVI WRL

AVI WRL
When this flag is set, DynaMaSZ generates single frames in wrl format instead of opening a

video stream. The 3-D object files can be exported into a program such as Bryce, for post-processing into videos. This works in conjunction with the Demo/Batch mode command, to set the target disk directory and file name, so is only enabled when Batch mode is set.

11.8 AVI LFM

AVI LFM

When this flag is set, DynaMaSZ generates single frames in lfm format instead of opening a video stream. The 3-D object files can be exported into a program such as Python, for post-processing into videos. This works in conjunction with the Demo/Batch mode command, to set the target disk directory and file name, so is only enabled when Batch mode is set.

12 Demo menu

Demo menu commands

The Demo menu offers the following commands, which illustrate various features of DynaMaSZ:

Random Matrix	Generate random matrix image.
Random Composite	Generate random composite matrix image.
Random Render	Apply random coloring filter.
Batch Mode	Repeat random fractal and save to file.

12.1 Random Matrix

Random Matrix (Demo menu)

A random matrix image is generated. Matrices are randomized.

See the hot keys section also for a description of the 'G' command.

12.2 Random Composite

Random Composite (Demo menu)

A random 3-D Julia fractal is generated using one of the composite Types. Two formulas are selected at random and mixed using one of Types 2-8. This option can be applied to all built-in formulas, including cubic Mandelbrot and octonion formulas. Not available when the custom formula batch option is checked.

12.3 Random Render

Random Render (Demo menu)

A random coloring filter is applied. This changes the surface textures of the 3-D figure.

12.4 Batch Mode

Batch mode/Random Setup (Demo menu)

Here you set parameters for batching and saving random-generated images to disk. You can also customize random variables to direct how the random scanning process works. When the Repetitions value is non-zero, up to 1000 random images can be generated and saved to disk. Use a unique Filename to prevent batch files from overwriting existing image files. The Scan Limit directs the program on how many scans it makes through each formula before it skips to a new formula (if an interesting 3-D fractal hasn't been found.)

If you select this option before opening a video stream, then instead of an AVI stream, the program initializes a set of bmp image files. Each 'frame' is written to the directory specified in the Demo/Batch mode window. If AVI Object or AVI WRL is checked before opening the stream, then the frames are written as a series of 3-D object files. Each frame is numbered with a postfix to the Batch-mode name from 0000 to 9999, e.g. 'quat0001.bmp'.

There are radio boxes that allow you to customize how random variables are processed to create new 3-D fractals:

Formula -- (default on) check to randomize built-in formula used
Lighting -- (default off) check to set default lighting
Symmetry -- (default off) check to randomize symmetry used
Rotation -- (default on) check to randomize camera angles
Coloring -- (default off) check to reset coloring parameters
Iteration -- (default off) check to randomize iterations
Z-Space -- (default on) check to set default z-space
Constants -- (default on) check to randomize the complex constants cj-cK
Latin Square -- (default off) check to randomize Latin square
Sign Matrix -- (default off) check to randomize sign matrix
Column -- (default off) check to randomize column
Custom Formula -- (default off) check to generate a random formula

The Bounds variable (default 0) acts to delimit the boundary scan after finding a random Julia set. Since the scanning process is closely connected with the Mandelbrot set boundaries, most matrix images found this way are very connected/closed figures. The bounds variable adds a random distance from the Mandelbrot boundary to produce more open fractals. A good value to start with is 25 if you want to experiment with this option.

13 Help menu

Help menu commands

The Help menu offers the following commands, which provide you assistance with this application:

Getting Started	Tutorial for new users of DynaMaSZ.

Index	Offers you an index to topics on which you can get help.
Parser	Quick reference to DynaMaSZ's parser variables and functions.
Hot Keys	Quick reference to DynaMaSZ's hot keys.
Built-in Formulas	Quick reference to DynaMaSZ's built-in formulas.
Bibliography	Sources for fractal information and complex numbers.
About DynaMaSZ	Displays the version number and author info for this application.

13.1 Tip of the day

Tip of the day

Supplies extra information (in addition to this help file) that may be useful in some cases. Users are welcome to add to the file "tips.txt", if they have discovered an undocumented feature or method of obtaining certain results. These "tips" may become part of the next release of DynaMaSZ (drop them in an email to MysticFractal@comcast.net)

13.2 Getting Started

DynaMaSZ - Dynamic Matrix System Z

A Tutorial by Godwin Vickers
Version 1.2
March 2002

Overview:

DynaMaSZ is scientific visualization software, which allows you to render 4D Julia-sets and 4D Mandelbrot-sets of matrix algebras. These algebras are all 4x4 matrices and can be defined from two separate sets of rules. 24 Latin squares can be combined to 8 sign matrices. The result is a set of 596 different 4x4 matrix algebras to which any kind of iterative functions can be applied.

This tutorial will explain most of the features of DynaMaSZ. Some options won't be described here either because they are self-explanatory, because they are already fully explained in DynaMaSZ's documentation or because they require a very deep understanding of the math involved here.

1/ The Demo menu:

The Demo menu is very useful to scan randomly all possible parameter combinations according to several search rules. The Batch mode/Random setup window is the central command panel, which allows the user to select that parameters will be randomized and which parameters will remain fixed during the scanning process.

a/ The Batch-mode/Random Setup (or scanner/Random Generator) :

"Filename" and "Repetitions" are batch-mode parameters so we'll discuss these later on in this tutorial.

- Scan Limit : it tells DynaMaSZ how many times each formula will be scanned to find interesting shapes. A low value (below 50) will scan many formulas and eventually stop once an appropriate image is found. A high value (up to 99999) will go through fewer formulas because the scanning process is more likely to find an appropriate image within the first formulas scanned. This is how it works : you have a built-in database of formulas and DynaMaSZ will randomly pick up one and scan it several times, if it finds a good picture it will render it and the scanning is over. If no appropriate picture is found using that formula then DynaMaSZ will randomly choose another formula and scan again. This will go on until a final picture is found.

- Bounds : acts to delimit the boundary scan after finding a random Julia set. Since the scanning process is closely connected with the Mandelbrot set boundaries, most matrix images found this way are very connected/closed figures. The bounds variable adds a random distance from the Mandelbrot boundary to produce more open fractals. A good value to start with is 25 if you want to experiment with this option.

- Formula : selecting this option will allow the formula scanner to randomly scan any formula in the formula database. If you don't select this option the scanner will only scan one formula, the one that is already been used. Basically this option lets you choose between scanning the whole built-in formula database and only one formula. Usually the right thing to do is to select this option, run the scanner several times until you find a picture that you like, unselect this option and run the scanner again a few times so that in the end the scanner will only scan your selected formula.

- Default Lighting : selecting this option will change the lighting parameters for every new picture that the scanner renders. If you like the current lighting settings you should not select this option. If you are not satisfied with the current lighting settings or if you want to see some alternate settings you should select this option.

- Symmetry : this option allows the user to render either the left part of the picture and mirror it on the right, render the top part of the picture and mirror it below or render only one fourth of the picture and mirror it along the X and Y axis. The default is 'off'.

- Rotation : this option allows the scanner to randomly rotate each new object. Note that in DynaMaSZ the camera is always fixed, only the 3D objects can be rotated.

- Latin square : checking this option will randomly select a Latin square in the built-in database of 24 4x4 Latin squares.

- Column : selecting this option will change the way matrix multiplication is made to allow 'hidden' Julia-sets to be rendered. Instead of multiplying the first column-vector of the matrix to the whole matrix to square it this feature chooses randomly one of the four matrix column-vectors and multiplies it to the whole matrix.

- Coloring : if this option is selected the palette and the way the palette is mapped to the 3D objects will be randomized every time the formula scanner is launched. If you like the current palette you should not select this option. You should only select this option if you don't like the current palette or if you want to see other palettes.

- Default Iterations : selecting this option will randomly change the iteration value for each new scan. The higher the iteration value the more precise the 3D objects will be and vice-versa. Note that high iteration values take more time to render and that low iterations values are faster and look smoother.

- Default Z-Space : selecting this option will randomize x and y for every new scan. z and w are not affected by this option.

- Constants : selecting this option randomizes all 4 dimensions x, y, z and w for each new scan. This will create unpredictable shapes and is a very important option.

- Sign matrix : checking this option will randomly select a sign matrix in the built-in database of 8 4x4 sign matrices.

- Custom signs : checking this option will randomly generate a sign matrix without following any rules (may not generate fractals, only recommended for experimental purposes).

- Custom formula : checking this option will randomly generate a formula during the scanning process. This feature doesn't pick up a formula in the list if built-in formulas.

b/ The functions:

Now that you know more about the options of the Random Generator or scanner you can choose to randomize everything or only several options depending on what you are looking for. At this stage you are ready to launch the scanning process. Click on "OK", go to the Demo menu again and select between Random Matrix or Random Composite to launch the scanner.

- Random Matrix : this command can scan all built-in formulas according to the parameters set by the user in the Batch mode/Random setup window. If you are not satisfied with the rendered picture you can keep on re-launching the scanner until you like the picture randomly generated.

- Random Composite : this command scans all built-in formulas and finds two formulas randomly which are used together to generate a Julia set. The way the two formulas interact is randomized and all dimensions can be randomized as well. This feature will give totally unpredictable shapes that are always randomly chosen and optimized so that they look interesting to the human eye.

- Random Render : this command should only be used once you are satisfied with a Julia-set shape but not with the colors. It will not randomize the colors but will randomly select a way to map the color palette to your 3D object from several built-in palette-mapping functions.

2/ The Edit Menu

The Edit menu can be used before or after the Demo menu. You can also use the DynaMaSZ without ever using the Demo menu and choosing to set up all parameters manually in the Edit menu.

"**Formula/Type**" has the following options :

- fun #1 : select one of the built-in functions (or formulas).
- fun #2 : select a second function from the database if you want to create a composite (combo) formula.
- Type : you can choose from 0 to 8 how you want the two formulas to interact. 9 lets you use the parser to create your own custom formula.
- Latin : lets you select one of the 24 built-in 4x4 Latin squares.
- Column : (between 0 and 3) Lets you choose which of the four columns will be multiplied to the whole matrix. This is an important feature that allows many more shapes to be discovered.
- Sign : lets you select one of the four built-in sign matrices.
- Custom Latin : lets you define a 4x4 matrix of 4 variables x, y, z and w (for experts only).
- Custom sign : lets you define a 4x4 matrix of numeric constants (for experts only).
- Random Formula : this command creates a random standard formula according to the level of randomization selected. There are 9 levels of randomization (from 2 to 10): 2 generates a small formula and 9 generates a very long formula.
- Random fun #1 and Random fun #2 randomly select functions 1 and 2 respectively.
- Formula : lets you define a custom formula using the DynaMaSZ interpreter. The help menu includes a full description of the interpreter and of its features.

"**Size**": here the user can select one of the 10 built-in picture sizes or a custom picture size. The larger the size of the picture the longer the rendering takes.

"**Initial values**" has the following options:

-X/Y/Z Min. and Max. are the window's parameters, you don't need to manually edit these values.

- 4th Dimension : this parameter is a dimension that you can use to change the shape of your 3d objects, usually between -1 and 1. I suggest not changing this value but it could be used for animations. Default is 0.

- Rotation : you can use X, Y and Z to rotate a Julia-set (in degrees). For fast rotation use the Shift-R hotkey instead of changing these values manually (that option is described further down).

- Constants : these are the most important parameters. The constants of your Julia-set should in general be set between -1 and 1 each to create distortions but sometimes they can be pushed to two or more.

- Max. Iter : this is the iteration value. Low values (below 5) generate very simple and smooth shapes while higher values are more complex and detailed. The higher the value the longer the rendering time takes. A value of 7 or 8 is usually appropriate but some formulas require higher or lower values to show their full potential.

- Bailout : this parameter should stay at 4.0 - its default value - but decreasing the value can be interesting in an animation if you want the Julia-set to become smaller and fade to nothing.

- Steps : this value determines the quality of the rendering. The default value of 200 should only be used when creating the Julia-set and modifying it. For the final, full-size rendering you should set the value to 2000 or more for optimal picture quality. This parameter is extremely important and should always be changed before a final rendering but it affects rendering-time tremendously so you will need a very fast computer if you want high-precision renderings.

- FineTune : this parameter also affects the quality of the rendering and should be left at 20 before the final rendering. Set a maximum value of 99 just before your final rendering.

"Ray-tracing variables" : these variables are all related to lighting.

- Lightpoint : the X,Y and Z values represent the coordinates of the source of light in 3D space. A low Z value along with and X or Y value above 1 can be used to add more shadows.

- ViewPoint : represents the angle at which the object is ray-traced, which can affect Phong highlights greatly. This has no effect on the camera view.

3/ The Type Menu

Here you can choose between Mandelbrot0, MandelbrotP and Julia. Mandelbrot0 will render a standard Mandelbrot set (orbit starts at zero). MandelbrotP will render a Mandelbrot set but orbit starts at pixel. Julia will render a Julia-set.

For any given formula you have the choice between rendering a 3D Julia set and a 3D Mandelbrot set. This is the same as using "Switch Z for C" in the Pixel menu. The two 2D options allow you to display your selected formula as regular 2D fractal.

4/ The Render Menu

Anti-aliasing can only be used if solid Guessing is disabled in the Pixel menu. This feature will create excellent quality renderings but it is an extremely slow process. 1x4 is better than 1x2 but much slower. Another way to achieve a similar effect is to set a much bigger image size and then to reduce it with an image-editing program like Photoshop or Paint Shop Pro.

You also have access to huge list of coloring filters and a noise generator to customize the look of your Julia-set. You can use one of the 280 TieraZon2-compatible-plug-in rendering methods for added flexibility.

5/ The Pixel menu

Solid guessing should only be used before the final rendering because it is very fast. For the final rendering it should be disabled for optimal quality.

6/ The Color menu

Once you have generated a Julia-set that you like you should go to the color menu and click on 'Random palette' as many times as you want until you find an appropriate palette. You can also choose to create a palette manually by using the three RGB (Red, Green and Blue) parameters and the Spread option.

7/ The Video menu

Julia-sets may look great as simple pictures but because of their multi-dimensions they are especially interesting as animations. There are two ways of creating a video with DynaMaSZ, either by saving an AVI file directly or by saving each frame of the whole animation individually. If you want to use the frame-by-frame method you have to create a 3D object that you like then go to the Batch Mode/Random Setup window and choose a name for the series of files in the Filename box. Then set a repetitions value of 1 and click Ok. After this you can use the Video menu. If you don't want each image to be saved separately but prefer one big AVI file then go directly to the Video menu.

In the Video menu click on 'Open AVI Stream'. If you've selected the batch option (that is frame by frame) you can ignore the following pop-up window by simply clicking on 'Ok'. If you want to save an AVI file then type in the name of the AVI filer to be saved and also the number of frames per second. For standard PAL it should be 25 and for NTSC it should be 30 (the human eye is only supposed to be able to see 24 pictures per seconds, anything more is more or less a waste of space but some people are more perceptive than others). Finally Click Ok and then choose the directory where you want to save your AVI file or your frames. If you haven't selected the frame-by-frame option you will then be asked which AVI compression method to use. If you want a professional quality video you should select 'Uncompressed frames' but depending on the size of the frames you can also select MPEG-4 v1, v2 or V3 (V3 is hard to find and will also need to be installed on the computer that plays the video) or DivX (install the DivX codecs first and select the slow-motion codecs).

That's it, at this stage your video is ready to be calculated and you'll be able to repeat this process easily over and over again as you create more videos. The whole concept here is to create some keyframes and DynaMaSZ will automatically interpolate between keyframes. As the very first picture of the animation is your first keyframe you now need to define a second keyframe in your animation. For example you can zoom in or zoom out or you can rotate the Julia-set, change some parameters, etc. as long as you keep the same formula, color palette and you do not switch from a Julia set to a Mandelbrot set (or vice versa) the keyframes will be interpolated. You can go to Edit/Initial values and change some values here, especially x, y, z or w and/or the rotation parameters if you want the Julia-set to rotate as well.

Once you have changed some parameters you have to tell DynaMaSZ it is a new keyframe by going to the Video window and clicking on 'Write Frames'. In the popup window select how many frames you want DynaMaSZ to render between the previous keyframe and this new one. A small value will create a short and fast video while a high value will create a long and slow video. Select 'Log Scaling' only if you want some logarithmic effect (starts fast and ends slow). Click on 'Ok'. Now DynaMaSZ is going to render all the desired frames and it may take several hours or days depending on the speed of your CPU. Once the final frame is rendered you can go on modifying parameters and defining keyframes as many times as you want until you reach the end of your animation. To finish the animation go to the Video menu and click on 'Close AVI Stream'.

If your video is a large uncompressed AVI file you will need to compress it with Premiere or VirtualDub (http://www.virtualdub.com) (a great free video software). For small videos MPEG-1 is fine and for large videos MPEG-2 is the best and MPEG-4 is also good. If your video is a series of BMP pictures you can open Premiere and import a series of 'numbered stills' and then compress it and save it as one large video file.

8/ Hotkeys, exporting and advanced tips

A fast way of rotating your favourite Julia-sets is to use the Shift-R (R as Rotate) hotkey and then left-right, up-down and ins-del to rotate it. This is the most important hotkey, for all other hotkeys please read the appropriate DynaMaSZ help file. [Author's note: the shift-R command has been replaced with the Pilot.]

You can export Julia-set-meshes by using a polygonising feature :
- Lightflow (LFM)
- Maya (OBJ)
- VRML 1.0 (WRL)

This is done by selecting the File/Save other/Save Q Polygon option.

If you wish to export a Julia-set animation you have to follow the same steps as with a frame by frame video rendering but before opening the video stream you have to select the appropriate export function in the Video menu (LFM, OBJ or WRL).

Here is an important tip to generate great Julia-set videos :

You may want your Julia-set video to start with a blank screen, then a Julia-set suddenly emerges from darkness, grows and morphs into various shapes and then fades again into darkness. To do this you can save each keyframe of your video individually prior to opening an AVI stream. First find a great Julia-set by using the Random Generator and change one of its 4 dimensions so that it becomes a blank screen and save it using 'Save Parameters'. Then change the parameters again and save it (if the first keyframe is called project01.dmz, call the second keyframe project02.dmz and so on). Do the same for every keyframe and the last keyframe should also be a blank screen. Now re-open the first keyframe and open your AVI stream. Open the second keyframe using Load Parameters and Write Frames. Open the third keyframe and Write Frames. Follow the same procedure for every keyframe and finally close you AVI stream. The key to a successful Julia-set video is to learn how to play with the 4 dimensions (you can also combine it with rotations), after a while it becomes almost natural.

A useful way of discovering new interesting Julia-set matrices and formulas is to launch random batch renderings overnight. Before going to bed you can open the Batch Mode/Random Setup window, enable 'Formula', 'Latin Square', 'Sign Matrix', 'Column', 'Rotation', 'Custom Formula', 'Iterations', 'Z-Space' and 'Constants' and select how many random Julia-sets you want to render in the 'Repetitions' box. Next click 'Ok' and then select Random Matrix or Random composite in the Demo menu. The next morning you can have up to 999 random Julia-set pictures saved on your hard-drive along with their parameter files.

Godwin Vickers

(For more information on Latin squares and sign matrices, see Godwin's tutorial on Matrix Algebra.)

13.3 Index

Index command (Help menu)

Use this command to display the opening screen of Help. From the opening screen, you can jump to step-by-step instructions for using DynaMaSZ and various types of reference information.

Once you open Help, you can click the Contents button whenever you want to return to the opening screen.

13.4 Hyperalgebras

Tridimensional visualization of 4D Julia sets generated from Latin squares and sign rules

January 2002
Version 1.3
Godwin Vickers

1/ Introduction
2/ Latin squares
3/ Sign rules and sign matrices
4/ Algebras defined as matrices
5/ Iterative functions
6/ Conclusion

1/ Introduction

This document shows a new method to define 4D algebras in R4 and higher dimensions. Iterative functions can be applied to these algebras and 4D Julia sets other than quaternion Julia sets can then be rendered and explored. The purpose of this document is to describe these new algebras and how they can be created from a set of rules that can be implemented in a specific algorithm. This is not a tutorial on 4D Julia sets or how to turn 4D Julia sets to

3D pictures (via Z-buffering, ray-tracing or radiosity). Plenty of tutorials explain step-by-step how to render 4D Julia sets, how to calculate the distance between a light source or a viewpoint and a 4D Julia set, how to calculate the normal at each point on the surface of a 4D Julia set and which theorems and formulas are involved in these processes. For more information check the links at the bottom of this document.

In the late 1970's Mandelbrot applied iterative functions to complex numbers and discovered mathematical objects which offered a potentially infinite level of detail at all scales, these details following a rule of self-symmetry as scale decreases or increases. Mandelbrot called these objects fractals. 2D fractals have been extensively visualized for the two past decades but 4D fractals are still hiding some unknown territories that we can now explore thanks to the combination of appropriate algorithms and the increasing speed of microprocessors.

Quaternions are naturally the start of this journey because they are a 4D generalization of complex numbers (or the complex plane). Quaternions are the only normed division algebra of order 4. In fact these algebras are all Clifford algebras and this is the way they are categorized:

Cl(0)	1 dimension	R (real numbers)
Cl(1)	2 dimensions	C (complex numbers)
Cl(2)	4 dimensions	H (quaternions)
Cl(3)	8 dimensions	O (octonions)

Once Cl(8) is reached, we obtain a 256-dimensional algebra which is the building block of all other Clifford algebras since they can all be calculated from Cl(8) using a periodicity theorem. All Clifford algebras are normalized.

The unit quaternion can be considered as a rotation in 4 dimensions and is defined by the following matrix :

x -y -z -w
y x w -z
z -w x y
w z -y x

Once you apply an iterative function to this matrix you can generate a 3D slice of your 4D Julia set. What we will show in this document is how you can modify the original quaternion matrix algebra and define a new set of 4D matrices which follow several building rules in order to explore new 4D Julia sets.

What we need to do first is dissociate the symbols from the signs so we can create a set of two matrices which, combined together, generate the initial matrix. Here are the two separate matrices :

x y z w
y x w z
z w x y

w z y x

and

+ - - -
+ + + -
+ - + +
+ + - +

(where '+' means '+1' and '-' means '-1')

Now that we have a matrix that represents the symbolic structure of the unit quaternion and a sign matrix that defines its sign rules we can analyze the two matrices and find the laws that create them. This reverse-engineering process leads not only to the rules of quaternion algebra but also to the rules of all other similarly built 4D algebras.

2/ Latin squares

The symbolic matrix we have extracted from our unit quaternion matrix is a Latin square. So in order to explore more algebras we'll define all the possible Latin squares of order 4 and keep the essential ones.

While magic squares are made of numeric values only, Latin squares are only made of symbolic values. Latin squares are to magic squares what algebra is to arithmetic.

The rules to construct a Latin square are the following:

-Once you define the order n (n being an integer) of your Latin square, n becomes the number of dimensions of your Latin square: this means there are n rows, n columns, and n different symbols.

-Each symbol must appear once and only once on each row and column of the Latin square.

There are exactly 4!*24 Latin squares of order 4 ((1*2*3*4)*24 = 24*24 = 24^2 = 576). These 576 Latin squares are the result of the symbol permutations of the 24 Latin squares of order 4 starting with a first row in lexicographic order. This means that 24 essential Latin squares can generate 576 (24^2) Latin squares if you just permute their symbols.

Here are the 24 Latin squares of order 4 with their first row in lexicographic order:

x y z w x y z w x y z w x y z w x y z w x y z w
z w y x y w x z w x y z z w y x y w x z y x w z
w z x y w z y x z w x y y x w z z x w y z w y x
y x w z z x w y y z w x w z x y w z y x w z x y

x y z w x y z w x y z w x y z w x y z w x y z w
w z y x w z x y w z x y z w x y w x y z y x w z
z x w y y x w z z w y x y x w z y z w x w z y x

y w x z z w y x y x w z w z y x z w x y z w x y

x y z w x y z w x y z w x y z w x y z w x y z w
z w x y w z y x y x w z z w x y z w x y w z y x
w z y x z w x y w z x y y z w x w x y z y x w z
y x w z y x w z z w y x w x y z y z w x z w x y

x y z w x y z w x y z w x y z w x y z w x y z w
y z w x z x w y w z y x z x w y y z w x y x w z
w x y z y w x z y w x z w z y x z w x y z w x y
z w x y w z y x z x w y y w x z w x y z w z y x

For Latin squares of order 4, there are exactly 4 normalized Latin squares. The correct terminology for normalized Latin squares is 'reduced' or 'standardized': this means their first row and column are in lexicographic order.

3/ Sign rules and sign matrices

Let's analyze first the sign matrix we have extracted from our quaternion matrix. If we follow the rules of sign multiplication of real numbers we can find the sign multiplication rules of quaternions in R4 (+ * + = +, + * - = -, - * + = - and - * - = +). Note that '+' means '+1' and '-' means '-1'.

  + + + +

- + - - -
- + + + -
- + - + +
- + + - +

Row 1: + * - * - * - = -
Row 2: + * + * + * - = -
Row 3: + * - * + * + = -
Row 4: + * + * - * + = -

Column 1: + * + * + * + = +
Column 2: - * + * - * + = +
Column 3: - * + * + * - = +
Column 4: - * - * + * + = +

The product of each value of each row is always negative and the product of each value of each column is always positive. These are the simple rules of our quaternion sign matrix. If you replace each + with a - and each - with a + you also get a quaternion. It doesn't matter if rows are negative and if columns are positive as long as rows are always of the opposite sign of columns and that all rows are of the same sign.

All we actually need are two strings of 4 signs to define the sign rules. If 'r' stands for 'rows' and 'c' for 'columns', we can sum up quaternion sign rules like this:

r++++
c----

And since the relationship between the r sign string and the c sign string is "r is the opposite of c", we only need to define one string of signs because the second string will automatically be the opposite of the first one. So '++++' or '----' is enough to define the sign rules of quaternion algebra.

Sign rules and corresponding sign matrices of order 4:

Four pluses:

a + + + +

Three pluses:

b - + + +
c + - + +
d + + - +
e + + + -

Two pluses:

f + + - -
g - + + -
h + - + -
i - + - +
j + - - +
k - - + +

One plus:

l - - - +
m - - + -
n - + - -
o + - - -

No plus:
p - - - -

a & p are opposites : dual sign rule 1 (ap or pa) : ++++ ---- or ---- ++++
b & o are opposites : dual sign rule 2 (bo or ob) : -+++ +--- or +--- -+++
c & n are opposites : dual sign rule 3 (cn or nc) : +-++ -+-- or -+-- +-++
d & m are opposites : dual sign rule 4 (dm or md) : ++-+ --+- or --+- ++-+
e & l are opposites : dual sign rule 5 (el or le) : +++- ---+ or ---+ +++-

f & k are opposites : dual sign rule 6 (fk or kf) : ++-- --++ or --++ ++--
g & j are opposites : dual sign rule 7 (gj or jg) : -++- +--+ or +--+ -++-
h & i are opposites : dual sign rule 8 (hi or ih) : +-+- -+-+ or -+-+ +-+-

Overall 2^4 = 16 sign rules, and 16/2 = 8 dual (or opposite) sign rules.

Sign rule 1	**Sign rule 2**	**Sign rule 3**	**Sign rule 4**
+ + + +	- + + +	+ - + +	+ + - +
- - - -	+ - - -	- + - -	- - + -

Sign matrix 1 Sign matrix 2 Sign matrix 3 Sign matrix 4

```
  + + + +                  - + + +                  + - + +                 + + - +

- + - - -        + + + + +               - - + + +                - - + + +
- + + + -                - - + + +                + + + + +               - - + + +
- + - + +                - - + + +                - - + + +               + + + + +
- + + - +                - - + + +                - + - + +               - + + - +
```

Sign rule 5	**Sign rule 6**	**Sign rule 7**	**Sign rule 8**
+ + + -	+ + - -	- + + -	+ - + -
- - - +	- - + +	+ - - +	- + - +

Sign matrix 5 Sign matrix 6 Sign matrix 7 Sign matrix 8

```
  + + + -                  + + - -                  - + + -                 + - + -

- + + + -                - - + + +                + + + + +               - - + + +
- + + + -                - - + + +                - + + + -               + + + + +
- + + + -                + + + + +                - + + + -               - - + + +
+ + + + +                + + + - -                + - + + -               + + - + -
```

Overall there are 2^4 = 16 sign rules. If we discard the symmetries there are 16/2 = 8 sign rules left.

4/ Algebras defined as matrices

So far we have dissociated the original unit quaternion matrix in two separate matrices and then we found the laws to create these two matrices and all the other matrices that belong to their group.

Symbolic structure:
There are 576 = 4!*24 Latin squares of order 4.
If we discard permutations we obtain 24 Latin squares of order 4.
Note that among those 24 Latin squares some symmetries or rotations exist. I have read that there are 7 really essential Latin squares of order 4 from which you can calculate the 24 standardized Latin squares and all 576 Latin squares of order 4.

Sign rules:
There are 2^4 = 16 sign rules which are 8 pairs of symmetric sign rules.

If we now combine all the symbolic rules and all the sign rules just like we can combine the symbolic quaternion matrix and its sign matrix we obtain a group in R4 of all the matrices of order 4 which follow both sets of rules. The group we'll define here is composed of 24*8 = 192 algebras.

To combine a Latin square to a sign matrix you must not follow standard matrix multiplication. Instead you have to multiply each value of your Latin square to each corresponding value of your sign matrix.

For example:

x y z w	and	- + + +	gives	-x y z w
z w x y		+ + + +		z w x y
w x y z		- + + +		-w x y z
y z w x		+ - + -		y -z w -x

5/ Iterative functions

The iterative functions commonly used to create Julia sets are polynomials of order 2 and above.

From now on we'll be dealing with standard inter-matrix operations. Addition and subtraction are simple : add or subtract each value of the first matrix to each corresponding value of the second matrix. Squaring a matrix or multiplying it to another matrix is the key thing to understand here:

-To square a matrix (or to multiply it by itself) you have to multiply all its values by its first column.

-To multiply a matrix M1 by another matrix M2 you need to define which matrix is first and which matrix is second because M1.M2 does not necessarily equal M2.M1.
M1.M2 is the first matrix multiplied by the first column of the second matrix.
M2.M1 is the second matrix multiplied by the first column of the first matrix.
This means that depending on which matrix is placed first the product of two matrices involves a full matrix and 1/n (with n being the order of the matrix) values of another matrix.

For example, if you want to apply the classic formula f(M)=M^2+c to the matrix below you need to follow the following procedure to square your original M1 matrix:

x -y z w
y x w z
z -w x -y
w -z -y x

Multiplied with the first column of the matrix:

x
y
z
w

This gives:

x*x -y*y z*z w*w
y*x x*y w*z z*w
z*x -w*y x*z -y*w
w*x -z*y -y*z x*w

Your set of 4 discrete formulas is then (each row is a discrete function):

f(x) = x*x - y*y + z*z + w*w = x^2 - y^2 + z^2 +w^2
f(y) = y*x + x*y + w*z + z*w = 2xy + 2zw
f(z) = z*x - w*y + x*z - y*w = 2xz - 2yw
f(w) = w*x - z*y - y*z + x*w = 2xw - 2yz

If you want to multiply this matrix to the matrix M2 below,

-x y z w
 w z y x
-z w x y
 y -x w z

Then M1.M2 =

x -y z w	*	-x	=	-x*x w*(-y) -z*(z) y*(w)
y x w z		w	=	-x*y w*(x) -z*(w) y*(z)
z -w x -y		-z	=	-x*z w*(-w) -z*(x) y*(-y)
w -z -y x		y	=	-x*w w*(-z) -z*(-y) y*(x)

and M2.M1 =

-x y z w	*	x	=	x*(-x) y*(y) z*z w*w
w z y x		y	=	x*(w) y*(z) z*y w*x
-z w x y		z	=	x*(-z) y*(w) z*x w*y
y -x w z		w	=	x*(y) y*(-x) z*w w*z

To add even more flexibility you can choose not to restrict yourself to the first column and use the second, third or fourth column of your 4x4 real matrix when performing inter-matrix multiplication. This can lead to very interesting Julia sets which you would not be able to see if you only used the first column.

To divide together arbitrary matrices of this group we need to define 1/M first. For transcendental functions (functions using trigonometry) we also need additional rules.

Once we have a squared matrix f(M) we can iterate the function several times like this: f(f(f(f(f(f(f(M))))))) for 7 iterations for instance. According to a pre-defined bailout value (4 is a good choice) we can then find which points belong to the set or not depending on their convergence, divergence or cyclic behaviour.

In terms of coordinates, we are iterating 4 discrete functions (f(x), f(y), f(z) and f(w), the four rows) of our squared matrix.

6/ Conclusion

This approach allows generalization to n-dimensional matrices. For each level of dimensions the same procedure can be followed: defining various groups of Latin squares of order n (among which are isomorphic and homeomorphic groups/classes, normalized groups, etc.), defining all sign rules or matrices of order n, combining both matrices to define an algebra and then applying an iterative function to visualize them. The larger matrices become the more complex their structure can potentially be and they can therefore reveal some even more interesting Julia sets in hyperspace. As computers get faster and faster it will be possible to slowly increase the number of dimensions at the same time so that larger and larger Julia sets are defined but the number of Latin squares for each number of dimensions increases exponentially and quickly reaches billions of billions.

13.5 Parser

Parser Information

Functions (capital letters are optional, and parenthesis are necessary around complex expressions)

The following information takes the form "standard function" ---"form used by DynMaSZ to represent standard function".

sine z --- sin(z) or SIN(Z)   ; where Z can be any complex expression
hyperbolic sine z --- sinh(z) or SINH(Z)
cosine z --- cos(z) or COS(Z)
hyperbolic cosine z --- cosh(z) or COSH(Z)
tangent z --- tan(z) or TAN(Z)
hyperbolic tangent z --- tanh(z) or TANH(Z)

cotangent z --- cotan(z) or COTAN(Z)
arctangent z -- atan(z) or ATAN(Z)
arcsine z -- asin(z) or ASIN(Z)
arccosine z -- acos(z) or ACOS(Z)
hyperbolic arctangent z -- atanh(z) or ATANH(Z)
hyperbolic arcsine z -- asinh(z) or ASINH(Z)

hyperbolic arccosine z -- acosh(z) or ACOSH(Z) -- implemented as asinh(z)/atanh(z)
vers z -- vers(z) or VERS(z) -- 1- cos(z)
covers z -- covers(z) or COVERS(z) -- 1- sin(z)
exp(z) or EXP(z) -- the exponential function
log(z) or LOG(Z) -- the logarithmic function -- implemented as ln(z+1)

Bessel -- bessel(z) or BESSEL(Z) -- first order Bessel function
Laguerre -- lag(z) or LAG(Z) -- 3rd order Laguerre function
Gaussian -- gauss(z) or GAUSS(Z) -- Gaussian probability function
absolute value of z --- abs(z) or ABS(Z) -- sqrt(x*x + y*y +z*z+w*w)
z squared --- sqr(z) or SQR(Z)
z cubed -- cube(z) or CUBE(Z)
square root of z --- sqrt(z) or SQRT(Z) -- implemented as sqrt(abs(z))
real part of z --- real(z) or REAL(Z)
imaginary part of z --- imag(z) or IMAG(Z)

modulus of z --- mod(z) or MOD(Z) or |z| -- (x*x + y*y +z*z+w*w)
conjugate of z -- conj(z) or CONJ(z) -- (x-y-z-w)

reciprocal of z -- rep(z) or REP(Z) -- inverts the matrix and returns the column one vector as a result
transpose -- trans(z) or TRANS(Z) -- transposes the matrix and returns its column one vector as a result
trace -- trace(z) or TRACE(Z) -- sums the diagonal elements of the matrix
rotate columns -- rotc(z) or ROTC(Z) -- rotate matrix columns in clockwise direction
rotate rows -- rotr(z) or ROTR(Z) -- rotate matrix rows in clockwise direction
reverse rotate columns -- rotc(z) or ROTC(Z) -- rotate matrix columns in counter-clockwise direction

reverse rotate rows -- rotr(z) or ROTR(Z) -- rotate matrix rows in counter-clockwise direction
swap columns -- swapc(z) or SWAPC(Z) -- swap outer and inner matrix columns
swap rows -- swapr(z) or SWAPR(Z) -- swap outer and inner matrix rows
swap adj columns -- saltc(z) or SALTC(Z) -- swap adjacent matrix columns
swap adj rows -- saltr(z) or SALTR(Z) -- swap adjacent matrix rows

if/then/endif – if(argument), then (phrase) endif -- if argument is true then do phrase else skip phrase('then' tag is optional, but a comma should follow argument or put 'if(argument)' on separate line)

if/then/else/endif - if(argument), then (phrase) else (phrase) endif -- if argument is true then do phrase else skip phrase and do alternate phrase('then' tag is optional, but a comma should follow argument or put 'if(argument)' on separate line)

Notes:

If/then/endif and if/then/else/endif loops can be nested only when endifs follow each other at the end of the loops. For example: if(argument) if(argument) then (phrase) endif endif.

All transcendental and trig functions are implemented as power series. In the case of the log and inverse trig functions, since their domain is limited to -1<z<1 there is no guarantee that they will work correctly in all formulas. The results for all trig and transcendental functions are truncated to their column one vectors.

Math operators

+ --- addition
- --- subtraction
* --- multiplication
. -- matrix multiplication
/ --- division
% -- matrix division
^ --- power function
< --- less than
<= --- less than or equal to
> --- greater than
>= --- greater than or equal to
!= --- not equal to
== --- equal to
|| --- logical or (if arg1 is TRUE(1) or arg2 is TRUE)
&& --- logical and (if arg1 is TRUE and arg2 is TRUE)

Notes:

The multiplication operator '*' multiplies a matrix variable by the column vector of the following variable. Therefore, to match the built-in formulas, you need to enter --last-- column vectors such as c# and composite terms. For example, ''c*cos(z)+c' would be entered 'z=cos(z)*c#+c#'. z^4 is implemented in the built-in formulas as sqr(sqr(z)). Other ways of entering z^4 such as z*z*z*z will give different results.

The '.' (dot) operator multiplies two terms as matrices, then uses the column one vector for the result.

The '/' division operator uses the column vector (specified by the column variable) as the divisor. The result is returned in the column one vector. The '%' operator is used to perform matrix division. The dividend is inverted, then multiplied by the divisor, using matrix multiplication, then the column one vector is used for the result.

The power operator works with whole number exponents only. If you use a quad variable as an exponent, the absolute integral value of the variable is used.

Constants and variables

complex constant --- c# or C#, read/write.

complex conjugate --- cc# or CC#, read-only.
cx -- the constant entered in the cx box in the Parameters window(use j# for parser)
cy -- the constant entered in the cy box in the Parameters window(use k# for parser)
e --- e or E -- 1e^1 -- 2.71828, read/write.
i --- i or I -- square root of -1,read/write.
iteration --- iter# -- iteration loop counter
j --- j# or J# -- real part of the cx constant, read-only.
k --- k# or K# -- coefficient of the cy constant, read-only.

Note: j and k are the actual values of the complex constant terms as they are used in the iteration process, so will vary when the Mandelbrot option is used.

m --- m# or M# or pixel --a complex variable mapped to the pixel location as defined by the z coordinates entered in the Parameters window, read/write.
maxit -- the maximum number of iterations, as set in the Parameters window, read only
p --- p# or P# -- uses the cx constant when the Phoenix option is chosen, read-only.
p1 – the complex constant entered in the cx and cy gadgets, read-only.
pi --- pi or PI -- 3.14159, read/write.
q --- q# or Q# -- uses the cy constant when the Phoenix option is chosen, read-only
x --- x# or X# -- real part of Z, read/write.
y --- y# or Y# -- coefficient of the imaginary part of Z, read/write.
z --- z or Z -- function value at any stage of the iteration process, read/write.
zn# or ZN# -- the value of z at the previous stage of iteration, read-only.

13.6 Hot Keys

Hot keys

Shift-G -- A Julia matrix image set is generated in sector 2, using an iteration count of 10 and other parameters are changed temporarily to suit matrix plots. (Z is set to 2.0, the rotational variables are reset and the light source is set to the default, if random lighting is enabled.). Once you find an interesting matrix set using "G", another window is opened that sets the fractal parameters to those in the exploratory qjulia window. The parameters in the exploratory window revert to their original Mandelbrot settings.

Shift-T -- annotate a picture with text. Cursor changes to a crosshatch, which you position over the area where you want the text to start. Then click the left-mouse button to transfer any text (from the Edit/Text window) to the picture. Can be used with Undo. Use the Edit/Text command to change font, text color or format text into multiple lines. This is useful for adding copyright/author info to a finished picture.

Up arrow --- forward cycle colors one step, including set color -- useable during plotting.
Down arrow --- back cycle colors one step, including set color -- useable during plotting.

Shift-C -- clear the draw window to the current background color.

13.7 Built-in Formulas

Built-in Formulas (enter the following prefix into the Function #1 or Function #2 edit boxes)

1	p0; z^2+c
2	p1; (1-z)*zc
3	p2; z(z-1/z)+c
4	p3; cz^2-c
5	p4; z^2+cz^2+c
6	p5; z^3+c
7	p6; ((z^2)*(2z+c))^2+c
8	p7; z^2+j+kzn
9	p8; cz^3+c
10	p9; z^2-cz^3+c
11	r0; z^3-z+c
12	r1; z^2+z^3+c
13	r2; z^3-zn+c
14	r3; z^4-z^2+c
15	r4; (z^2-c)(z+1)
16	r5; (z+j)(z+k)(z^2+1)+c
17	r6; (z+j)(z^2+z+k)+c
18	r7; (z-1)(z^2+z+c)
19	r8; z^5+c
20	r9; z^6+c
21	e0; z(z^5+c)
22	e1; z(z^6+c)
23	e2; z^7+c
24	e3; z(z^7+c)
25	e4; z^7-z^5+z^3-z+c
26	e5; z^3+j+kzn
27	e6; cz^2+zn+c
28	e7; z^4-c^4
29	e8; (z-c)(z+1)(z-1)+c
30	e9; z^2+c^2
31	s0; z^3+c/z+c
32	s1; 1/z^2+c
33	s2; z+z^3/c+c
34	s3; z^3/c+c
35	s4; c/(z*z)+c
36	s5; 1/(z*z)-z+c
37	s6; (1+c)/(z*z)-z
38	s7; c/(z*z*z)+c
39	s8; 1/(z*z*z)-z+c
40	s9; 1/z^2-cz-c
41	"t0; 1/z^2+exp(z)-c"
42	"t1; c*cos(z)+c"

43 "t2; z^3+sin(c)"
44 "t3; 1/z^3+sinh(z)-c"
45 "t4; z^3+tanz+c"
46 "t5; z^3+coshz+c"
47 "t6; z^2-bessel(z)+c"
48 "t7; lag(z)+c --- 3rd order Laguerre function
49 "t8; z^3+Gauss(z)+c -- Gaussian probability function
50 "t9; z^2+z*arcsin(z)-c

13.8 Bibliography

Bibliography

Complex Mathematics
Churchill, Ruel.V. and Brown, James Ward: "Complex Variables and Applications", Fifth Edition, McGraw-Hill Publishing Company, New York, 1990.

Korn, Granino A. and Korn, Theresa M.: "Manual of Mathematics, McGraw-Hill Publishing Company, New York, 1967.

Fractal Theory
Barnsley, Michael: "Fractals Everywhere", Academic Press, Inc., 1988.

Devaney, Robert L.: "Chaos, Fractals, and Dynamics", Addison-Westley Publishing Company, Menlo Park, California, 1990.

Mandelbrot, Benoit B.: "The Fractal Geometry of Nature", W.H.Freeman and Company, New York, 1983.

Peitgen, H.-O. and Richter, P.H.: "The Beauty of Fractals", Springer-Verlag, Berlin Heidelberg, 1986.

Formulas and Algorithms
Burington, Richard Stevens: "Handbook of Mathematical Tables and Formulas", McGraw-Hill Publishing Company, New York, 1973.

Kellison, Stephen G.: "Fundamentals of Numerical Analysis", Richard D. Irwin,Inc. Homewood, Illinois, 1975.

Peitgen, Heinz-Otto and Saupe, Deitmar: "The Science of Fractal Images", Springer-Verlag, New York, 1988.

Pickover, Clifford A.: "Computers, Pattern, Chaos and Beauty", St. Martin's Press, New York, 1990.

Stevens, Roger T.: "Fractal Programming in C", M&T Publishing, Inc.,Redwood City, California, 1989.

Wegner, Tim, Tyler,Bert, Peterson, Mark and Branderhorst, Pieter: "Fractals for Windows", Waite Group Press, Corte Madera, CA, 1992.

Wegner, Tim and Tyler, Bert: "Fractal Creations", Second Edition, Waite Group Press, Corte Madera, CA, 1993.

Whipkey, Kenneth L. and Whipkey, Mary Nell: "The Power of Calculus", John Wiley & Sons, New York, 1986.

13.9 About DynaMaSZ

About DynaMaSZ

>>>>> DynaMaSZ™ v3.048 ©2002 -- 2005 by Terry W. Gintz

DynaMaSZ requires a true-color video adapter for best results. It may work in 16-bit (high color), but this hasn't been tested thoroughly.

Memory requirements for DynaMaSZ vary with the size of the drawing area DynaMaSZ opens on, ranging from approximately 3 megabytes memory for a 640X480 area to 48 megabytes for a 2048X1536 area. Special routines have been added to reduce memory requirements for large bitmaps (up to 14400X10800) by writing these directly to a file instead of using a memory bitmap.

Acknowledgements: Special thanks to Frode Gill for his quaternion and ray-tracing algorithms, to Dirk Meyer for his Phong-shading algorithm, and to David Makin for sharing his ideas on quaternion colorings and 3-D insights. Thanks also to Francois Guibert for his Perlin noise example, and to Kerry Mitchell and Mark Townsend for allowing me to incorporate their coloring methods from Ultra Fractal. Kudos to Godwin Vickers for his remarkable work on full-screen quaternion videos, his gracious support and algorithmic input to my programming, and his diligent efforts to expand the realms of 3-D fractal visualization. The multi-windowing interface in DynaMaSZ is courtesy of that extraordinary and prolific fractal programmer, Steven C. Ferguson. Steve's contributions to the look and feel of DynaMaSZ and previous versions of my programs have had a deep impact on my fractal imaging experiments.

For a short history of this program, see Chronology.

13.9.1 Chronology

Chronology

History of this program:

In September 1989, I first had the idea for a fractal program that allowed plotting all complex functions and formulas while attending a course on College Algebra at Lane College in Eugene, Oregon. In November 1989, ZPlot 1.0 was done. This Amiga program supported up to 32 colors, 640X400 resolution, and included about 30 built-in formulas and a simple formula parser.

May 1990 -- ZPlot 1.3d -- added 3-D projections for all formulas in the form of height fields.

May 1991 -- ZPlot 2.0 -- first 236-color version of ZPlot for Windows 3.0.

May 1995 -- ZPlot 3.1 -- ZPlot for Windows 3.1 -- 60 built-in formulas. Added hypercomplex support for most built-in formulas.

May 1997 -- ZPlot 24.02 -- first true color version of ZPlot -- 91 built-in formulas. Included support for 3-D quaternion plots, Fractint par/frm files, Steve Ferguson's filters, anti-aliasing and Paul Carlson's orbit-trap routines.

June 1997 -- ZPlot 24.03 -- added Earl Hinrichs' Torus method.

July 1997 -- ZPlot 24.08 -- added HSV filtering.

December 1997 -- Fractal Elite 1.14 -- 100 built-in formulas; added avi and midi support.

March 1998 --Split Fractal Elite into two programs, Dreamer and Medusa (multimedia.)

April 1998 -- Dofo 1.0 -- supports new Ferguson/Gintz plug-in spec.

June 1998 -- Dofo-Zon -- redesigned multi-window interface by Steve Ferguson, and includes Steve's 2-D coloring methods.

August 1998 --Dofo-Zon Elite -- combination of Fractal Elite and Dofo-Zon

October 1998 -- Dofo-Zon Elite v1.07 -- added orbital fractals and IFS slide show.

November 1998 -- Dofo-Zon Elite v1.08 -- added lsystems.

April 1999 -- Split Dofo-Zon Elite into two programs: Fractal Zplot using built-in formulas and rendering methods, and Dofo-Zon to support only plug-in formulas and rendering methods.

May 1999 -- Fractal Zplot 1.18 -- added Phong highlights, color-formula mapping and random fractal methods.

June 1999 -- completed Fractal ViZion -- first version with automatic selection of variables/options for all fractal types.

July 1999 -- Fractal Zplot 1.19 -- added cubic Mandelbrot support to quaternion option; first pc fractal program to render true 3-D Mandelbrots.

September 2000 -- Fractal Zplot 1.22 -- added support for full-screen AVI video, and extended quaternion design options

October 2000 -- QuaSZ (Quaternion System Z) 1.00 -- stand alone quaternion/hypernion/cubic Mandelbrot generator

November 2000 -- Added octonion fractals to QuaSZ 1.01.

March 2001 -- Cubics 1.0 -- my first totally-3-D fractal generator.

May 2001 -- QuaSZ 1.03 -- added Perlin noise and improved texture mapping so texture tracks with animations.

July 2001 -- QuaSZ 1.05 -- improved performance by converting many often-used dialogs to non-modal type. Added generalized coloring and external rendering library support.

October 2001 -- FraSZle 1.0, QuaSZ formula and algebra compatible version of Fractal Zplot

November 2001 -- DynaMaSZ 1.0, the world's first Dynamic Matrix Systems fractal generator

January 2002 -- MiSZle 1.1 -- generalized fractal generator with matrix algebra extensions

May 2002 -- DynaMaSZ SE 1.04 (unreleased version)-- scientific edition of DMZ, includes support for user-variable matrix dimensions (3X3 to 12X12)

January 2003 -- Pod ME 1.0 -- first stand-alone 3-D loxodromic generator, Hydra 1.0 -- first 3-D generator with user-defined quad types and Fractal Projector a Fractal ViZion-like version of DMZ SE limited to 3X3 matrices

May 2003 -- QuaSZ 3.052 -- added genetic-style function type and increased built-in formulas to 180. Other additions since July 2001: generalized coloring, support for external coloring and formula libraries, and Thomas Kroner's loxodromic functions.

May 2003 -- FraSZle and Fractal Zplot 3.052 -- added random 3D orbital fractals, new 3D export methods, upgraded most frequently-used dialogs to non-modal type and added genetic-style function type. FZ now based on FraSZle except for built-in formula list and Newton support.

Index

- 3 -

- B -

- C -

- D -

- E -

- F -

- H -

- I -

- P -

- R -

- S -

- T -

- V -