WILBUR

1. Introduction

Every program needs some suggestions for getting started. The more complicated the program, the more examples are needed. Wilbur has gotten to be a rather complicated program in many ways, so a number of different tutorials are provided, from using Wilbur to make image bits like the program logo to dealing with real-world data sets to making completely synthetic world maps. Enjoy!

The tutorials used to be part of the main documentation but it turned out that they were primarily determining the size of the documentation file. To ease the burden, the tutorials were split out into a separate file.

2. Wilbur Logo

The Wilbur logo (the first image in this document) was created using PhotoShop and Wilbur. Wilbur was responsible for the initial field generation (the background) and the final coloring, Photoshop was responsible for the text processing and merging the text onto the background.

An image was created using Wilbur (a ridged multifractal, I forget the exact parameters) at about the final desired resolution.

The *Texture->Gray Maps->Height* operation was performed to convert the data to a 256-color height field. It was exported to an 8-bit BMP surface and read into PhotoShop. In Photoshop, the image was converted to grayscale to get rid of the palette information.

Then, a grayscale image the same size as the Wilbur terrain was created. The word "Wilbur" was painted into the middle of this image, and then a combination of blurring, duplication, and subtraction was used to get the "worn" look on the letters.

The surface and text images were added together (which clips the height nicely to 256 levels to give the flat tops on the letters).

The image was saved as a BMP file and read back into Wilbur as a surface.

The final coloring was set up and a blur operation was added to get rid of some of the 256-color steps. A little painting was done to get rid of some of the uglier discontinuities.

The result was output as a 24-bit BMP image.

The output was converted to JPEG using Photoshop. The final product is the image on the title page.

3. GTOPO30 and TerrainBase Usage

The GTOPO30 data set covers the whole earth at roughly 1 km resolution. Unfortunately, it doesn't include seafloor information. The TerrainBase data set covers the whole earth at roughly 10 km resolution and includes a pretty rough approximation of the seafloor topography.

From version 1.13 onward, Wilbur has the ability to merge the two.

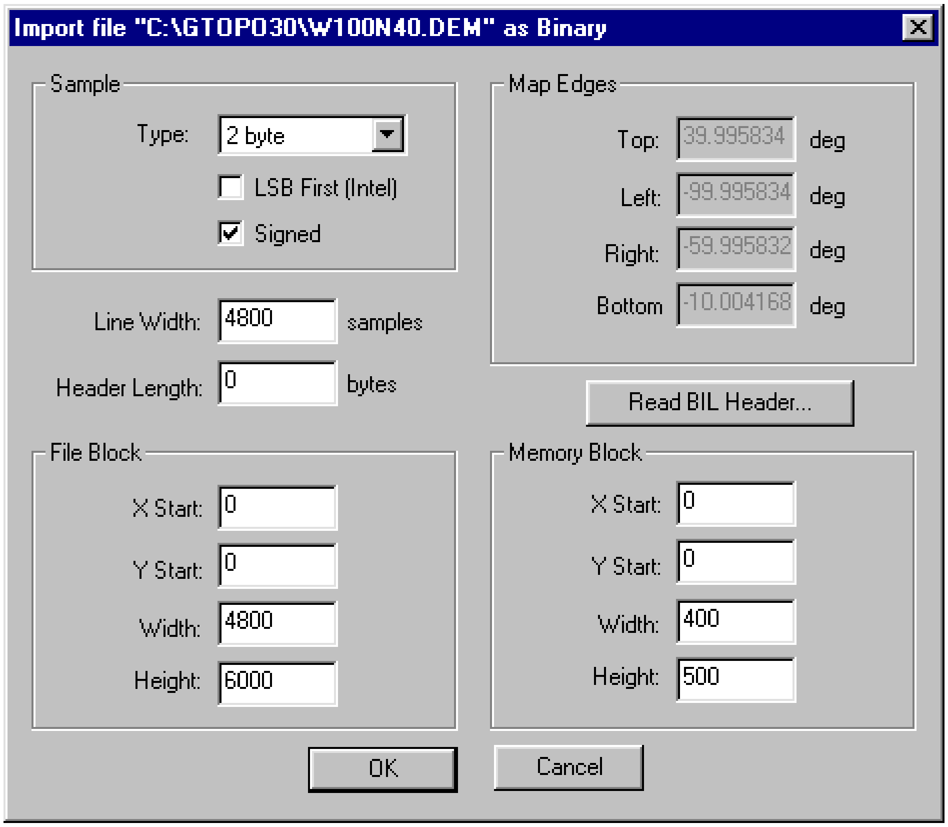
For this example, we'll use the W100N40 data set from the USGS for the GTOPO30 data and the full TerrainBase data for the TerrainBase data. This data set has the southeastern United States down through Mexico and northern South America.

First, get the data sets (see http://www.ridgenet.net/~jslayton/terrain.html for details). Decompress the .DEM file and the .HDR file. The .SRC files can be ignored for our purposes as can the rest of the files in the distribution. Note to WinZip users: turn off the smart CR/LF conversion facility for TAR files or your files may become corrupted.

Next, load the GTOPO30 data into Wilbur. The full data set is 4800x6000 samples, or 57.6 MB on disk (2 bytes/sample). In memory, the system needs 5 bytes per sample for a 256-color image of 7 bytes/sample for a 24-bit image (the default). That’s 138 MB for the 256-color map or 193 MB for the full-color map. Windows 95 automatically eats about 1/4 of total system memory for disk cache and other purposes on large-memory systems, so about 256 MB is a good amount of memory to process the whole data set at full resolution for 256-color image. To reduce the memory requirements, we'll downsample and load the data set at the same time using *File->Import->Image Subsection*. For this example, we'll use a 400x500 image size so that the surface will fit comfortably into any reasonable machine (it’s only about 1.5 MB in memory).

Set the surface size to 400x500 using *Surface->Size*.

Select *File->Import->Image Subsection*. A dialog will appear asking for the name of the file to load. Select the W100N40.DEM file extracted from the USGS file. Another dialog (shown below) will appear. Click the "Read BIL Header" button and then select the W100N40.HDR file that was also extracted from the file. Reading the header file will import the data settings for the .DEM file. In the memory block section of the dialog (lower right), enter 400 for width and 500 for height. Click OK and the system will import the data, resampling from 4800x6000 pixels to 400x500 pixels..



Import file "C:\GTOPO30\W100N40.DEM" as Binary

Sample

Type: 2 byte

□ LSB First (Intel)

☑ Signed

Line Width: 4800 samples

Header Length: 0 bytes

Map Edges

Top: 39.995834 deg

Left: -99.995834 deg

Right: -59.995832 deg

Bottom -10.004168 deg

File Block

X Start: 0

Y Start: 0

Width: 4800

Height: 6000

Memory Block

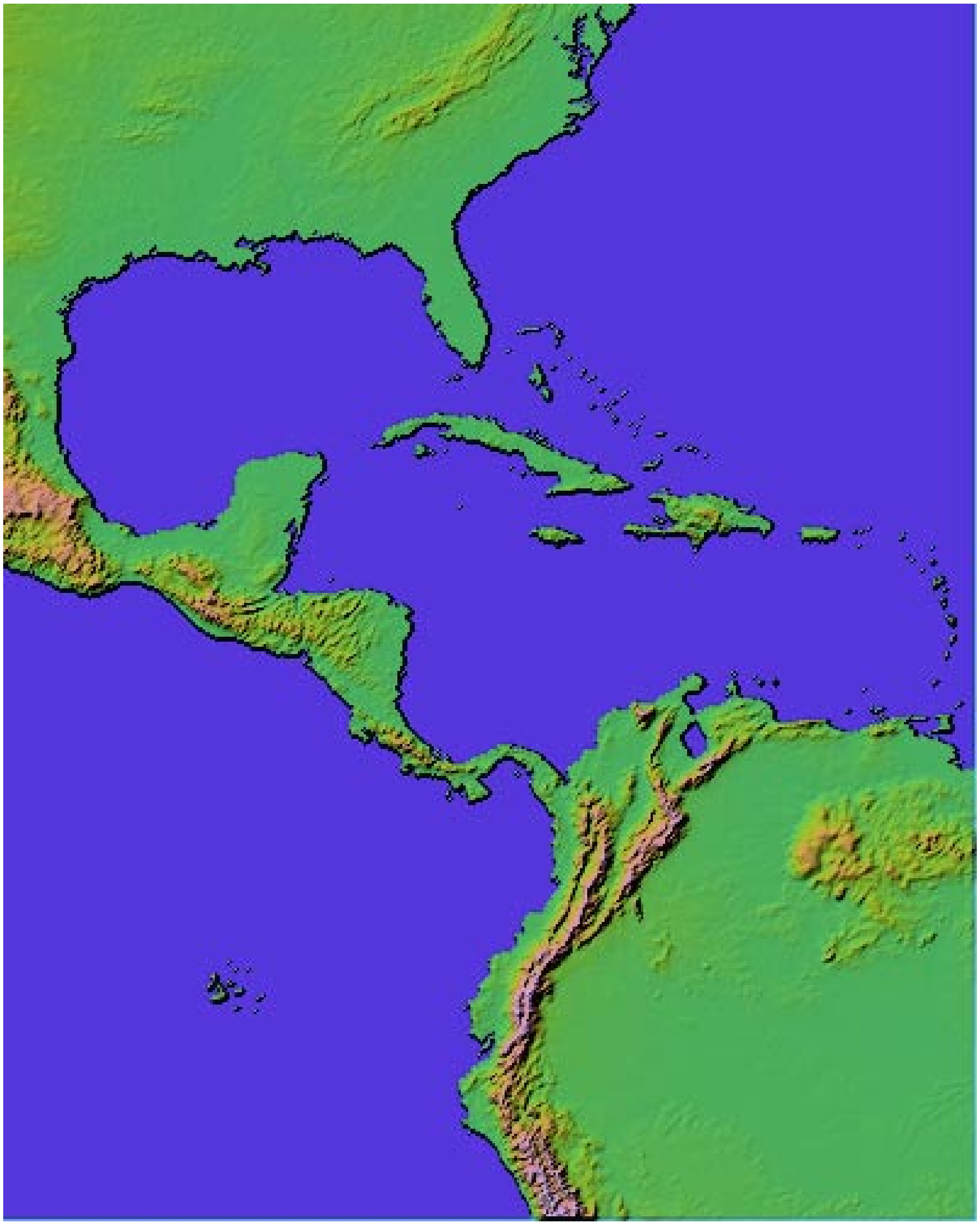
X Start: 0

Y Start: 0

Width: 400

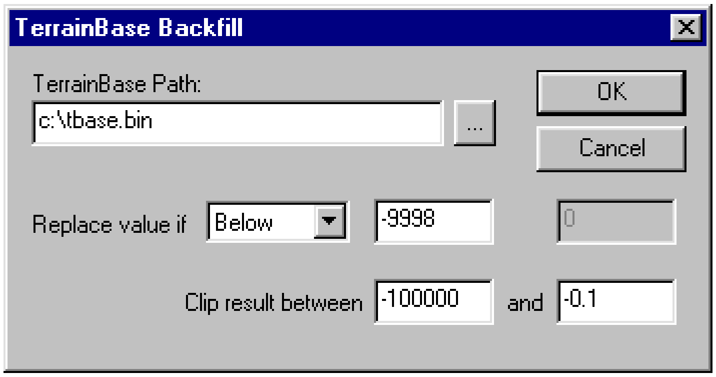
Height: 500

After the system recalculates the lighting map, it should appear somewhat like the following (assuming 24-bit calculations on a true-color display; appearances will vary based on display type):



Note the lack of sea floor and the heavy black lines that outline the continents/islands. The line is caused by the data suddenly going from roughly sea level to -9999 units. This sharp discontinuity registers as a very steep cliff, which gets painted black. There are two ways easy ways to remove the black lines: fill in with the TerrainBase data set or replace the value of -9999 (+/- 1) with a value of 0 using *Surface->Point Process->Replace*.

Now we'll put in some sea floor. While the above map is still in memory, set up for the TerrainBase import. Select *Surface->Area Process->TerrainBase Backfill*. Enter the data as shown in the dialog below, but substitute the path to the TerrainBase location on your hard drive. Click OK. All points less than -9998 (the sea data in the GTOPO30 data set) will be replaced with the TerrainBase data.



TerrainBase Backfill

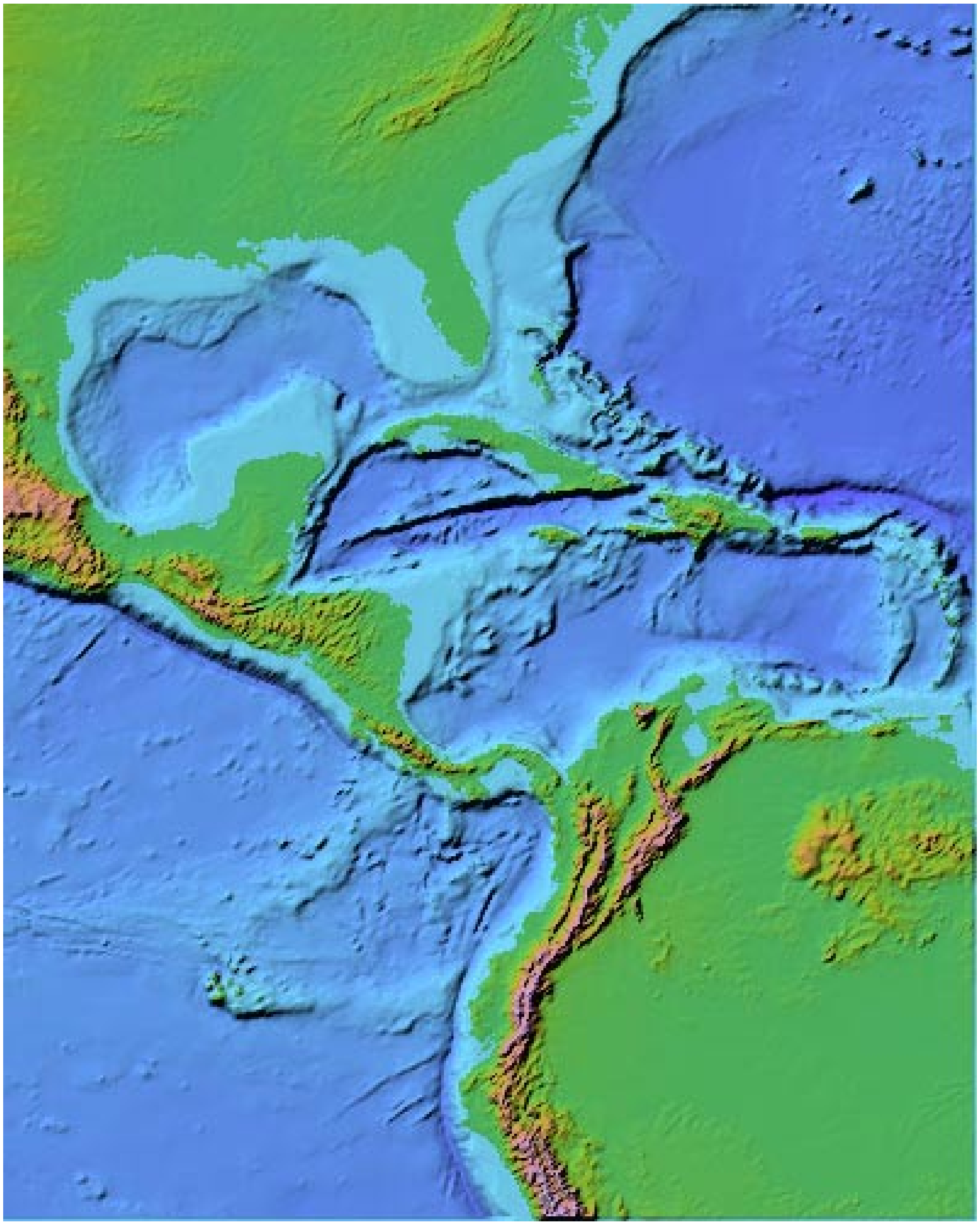
TerrainBase Path: c:\tbase.bin

Replace value if Below -9998 0

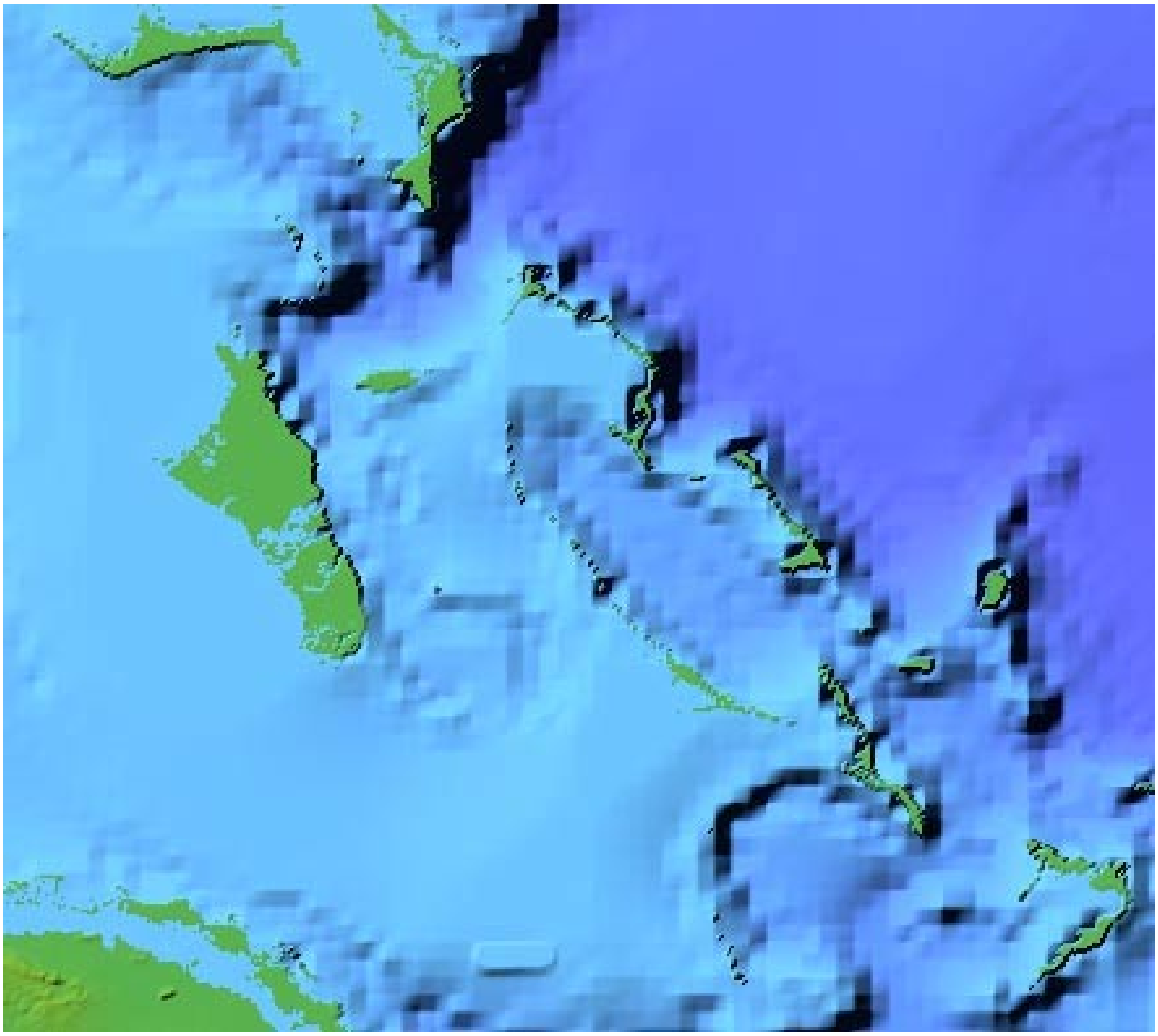
Clip result between -100000 and -0.1

The "clip result" parts are to ensure that the TerrainBase data won't show up as land. By clipping the data to between very deep and just below the surface, we ensure that interpolation errors in the data set won't cause spurious bits of land to appear.

The result now has seafloor data, as shown below.



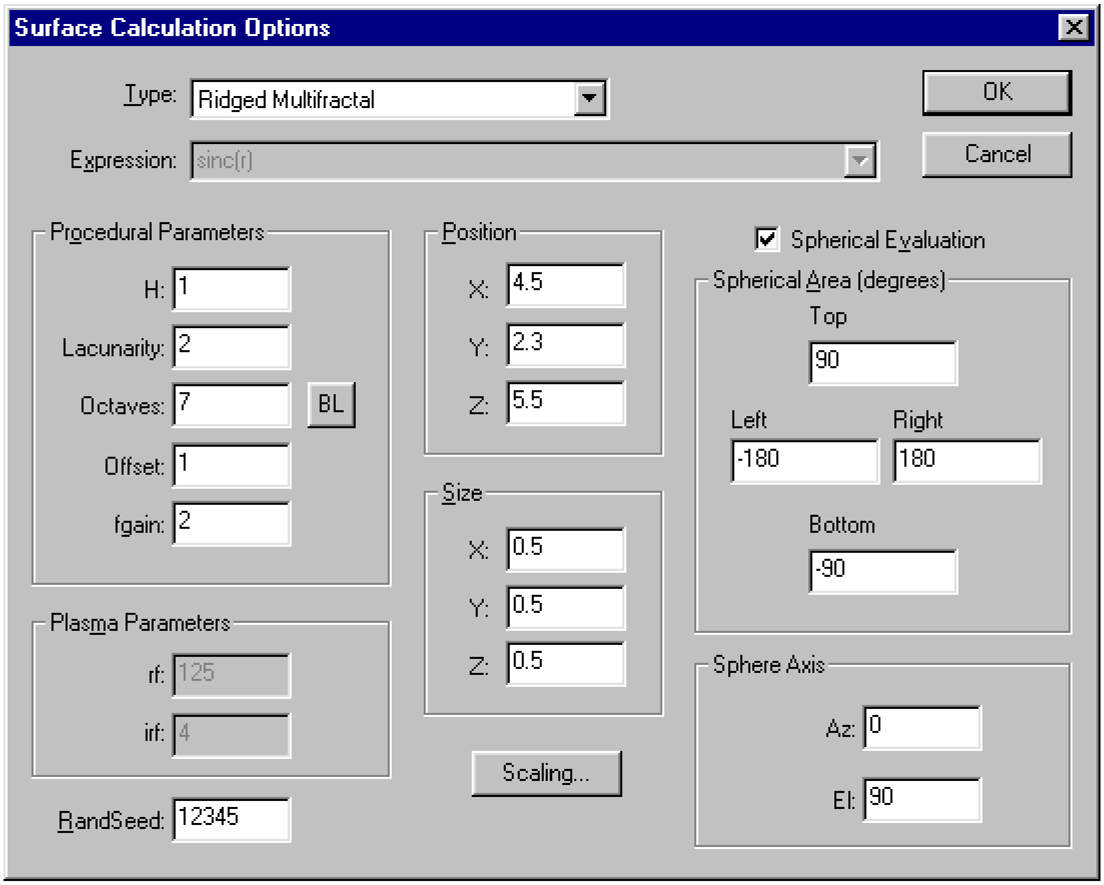
The results are much prettier for coastal maps. Unfortunately, using the TerrainBase data has a major drawback: the data is fairly low resolution (about 10 km per pixel) and Wilbur uses a simple linear interpolation scheme to extract the data. The image below shows the area in the Caribbean east of Cuba at roughly 2/3 scale; the faceting in the sea floor is painfully obvious. Also, notice that the two data sets don’t quite align perfectly for small details.



4. A Synthetic World

So far, we've focused on our own world map and getting data from other's data sets into Wilbur, then coloring. It's now time to create a map of a completely artificial world. We'll make a procedural height field that's mapped to a sphere, adjust the data to make it appear more like a real world, and finally, render it to disk (showing various height data bits).

Create a height field using one of the procedural terrain models. The "Ridged Multifractal" type seems to give the best results. The size used for the example will be 512 samples wide by 256 samples high and the edges of the area will be 90 for the top, -180 for the left edge, 180 for the right edge, and -90 for the bottom. The values used during the surface evaluation are shown in the screenshot below (and be sure to set the “Scaling” dialog values as well):



Surface Calculation Options

Type: Ridged Multifractal

Expression: sinc(r)

Procedural Parameters

H: 1

Lacunarity: 2

Octaves: 7

Offset: 1

fgain: 2

Plasma Parameters

rf: 125

irf: 4

RandSeed: 12345

Position

X: 4.5

Y: 2.3

Z: 5.5

Size

X: 0.5

Y: 0.5

Z: 0.5

☑ Spherical Evaluation

Spherical Area (degrees)

Top: 90

Left: -180

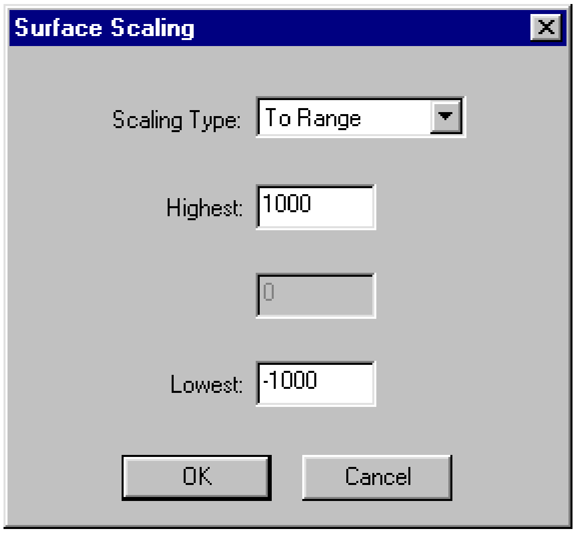
Right: 180

Bottom: -90

Sphere Axis

Az: 0

El: 90



Surface Scaling

Scaling Type: To Range

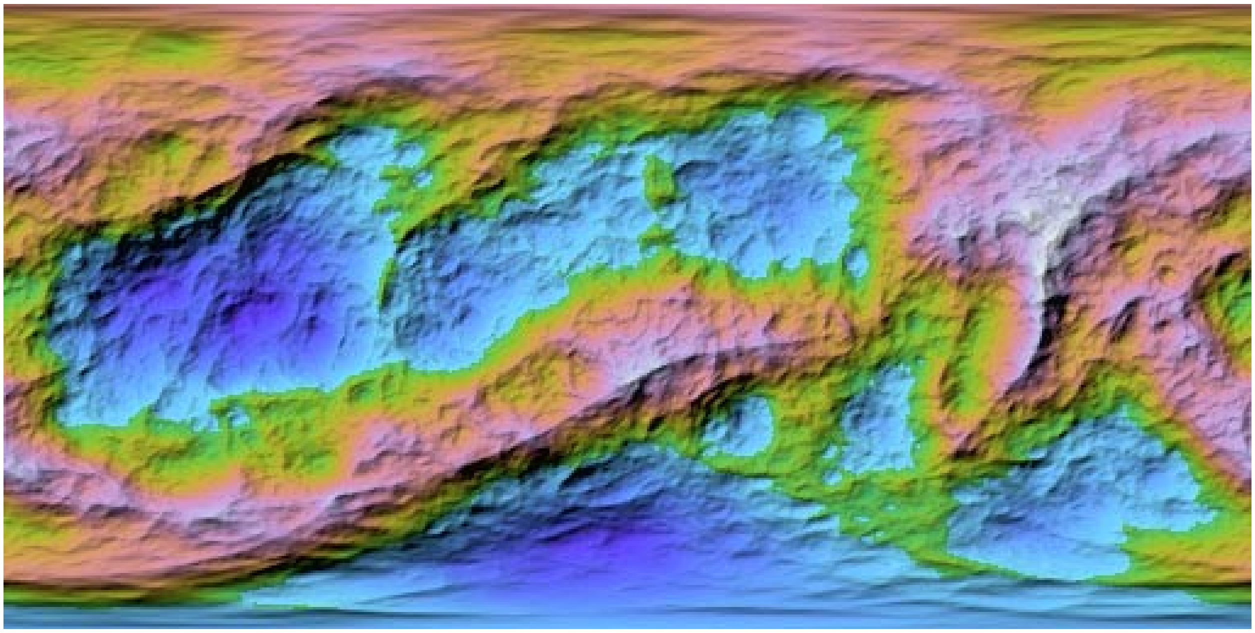
Highest: 1000

0

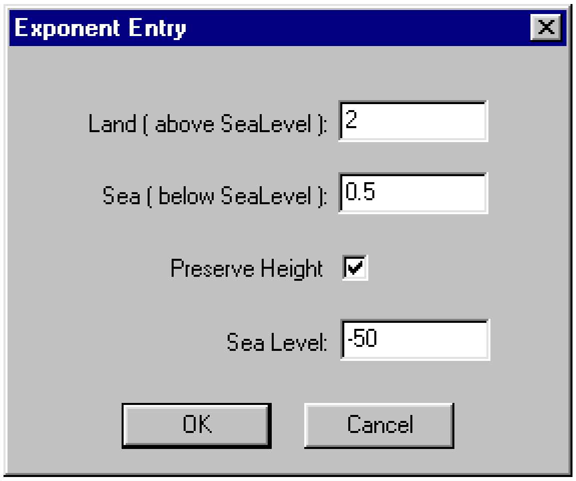
Lowest: -1000

In the context of spherical evaluation, the position values represent the center of the spheroid used for the calculations. The Size parameters are the X, Y, and Z radii, respectively, of the evaluation ellipsoid. Smaller values will give larger land masses.

The lighting defaults should give a picture very similar to the one shown below:



It's not real convincing yet. There are no continental shelves and the land is awfully rough near sea level over large areas. Fortunately, noth problems can be addressed with a single operation. Just use *Surface->Point Process->Exponential* on the surface with the parameters shown below (the result is shown below that):

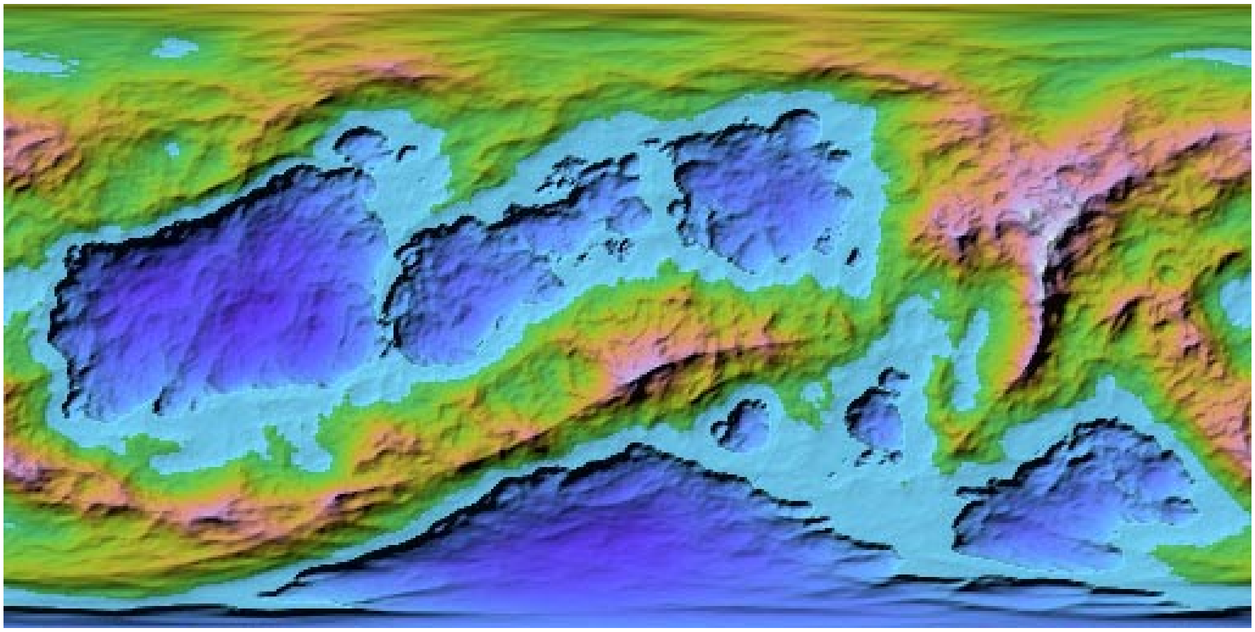


Land (above SeaLevel): 2

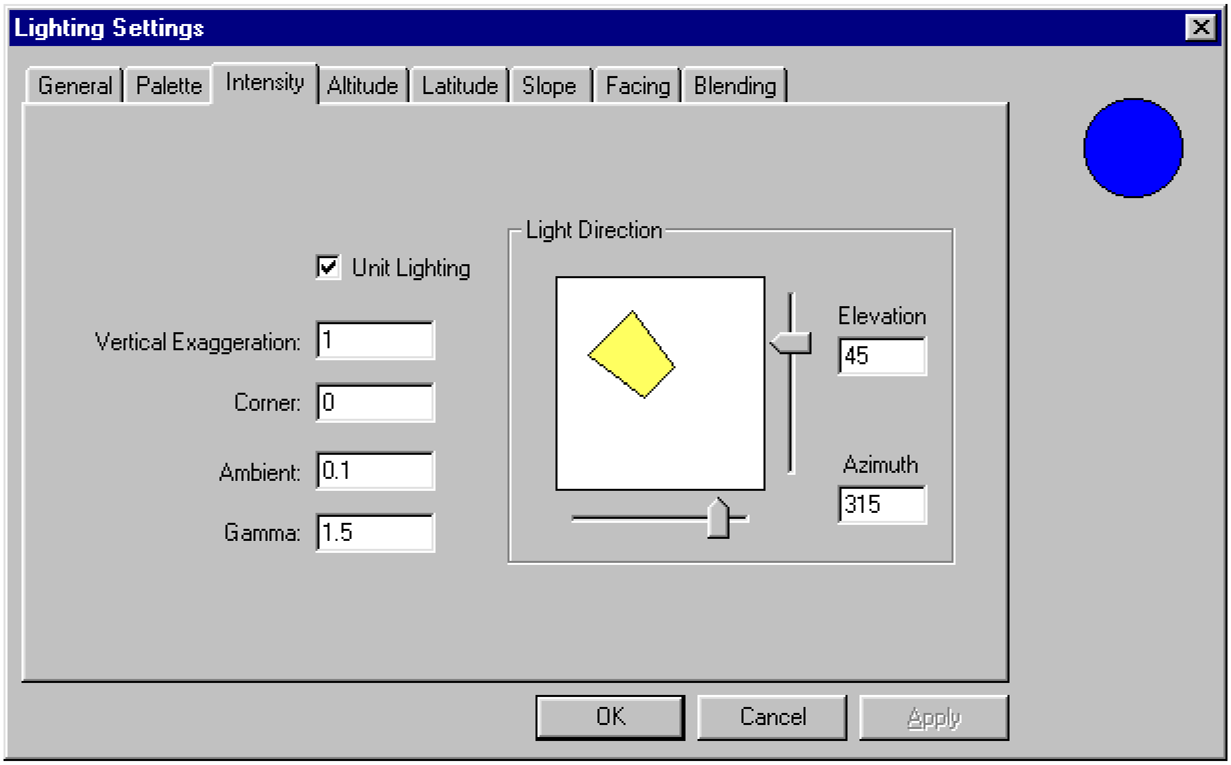
Sea (below SeaLevel): 0.5

Preserve Height ☑

Sea Level: -50



The results can be adjusted in many ways using the lighting parameters. For example, we'll add ice caps and some coloring based on the latitude of the world in question. The property sheet pages for controlling the lighting are shown below, followed by the resulting image:



☑ Unit Lighting

Vertical Exaggeration: 1

Corner: 0

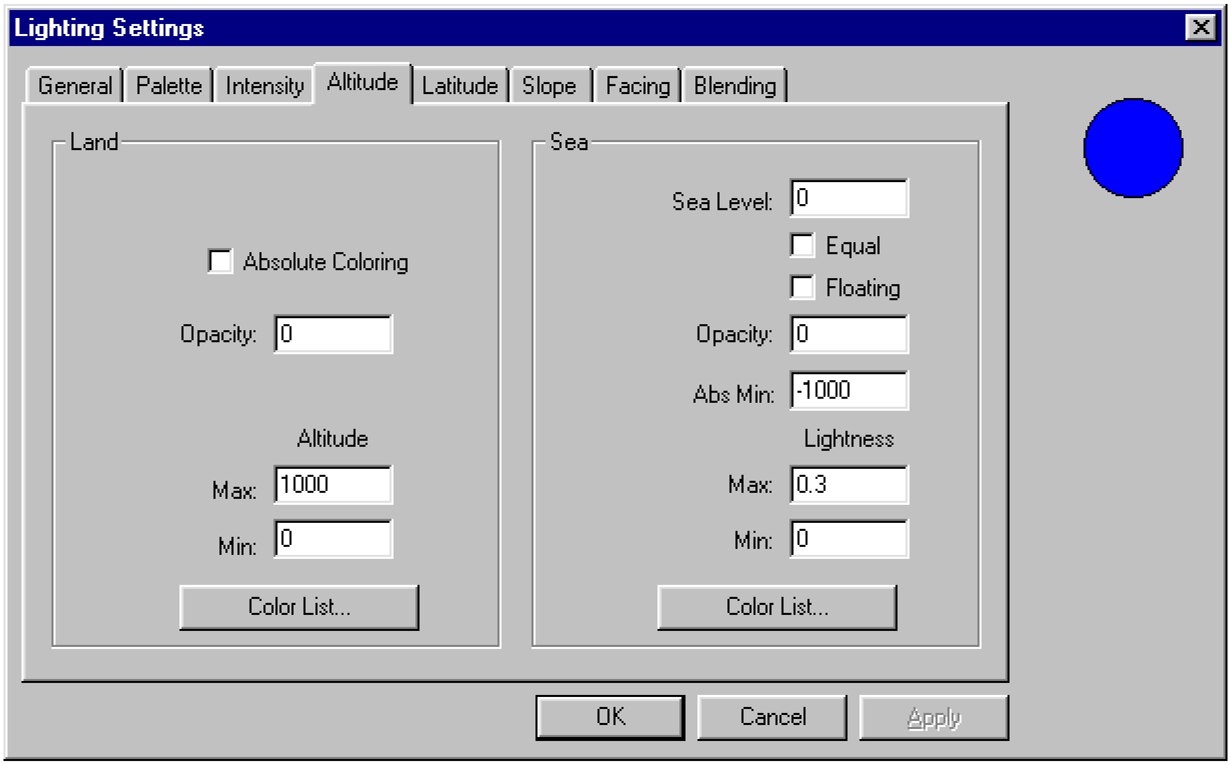
Ambient: 0.1

Gamma: 1.5

Light Direction

Elevation: 45

Azimuth: 315



Land

□ Absolute Coloring

Opacity: 0

Altitude

Max: 1000

Min: 0

Sea

Sea Level: 0

□ Equal

□ Floating

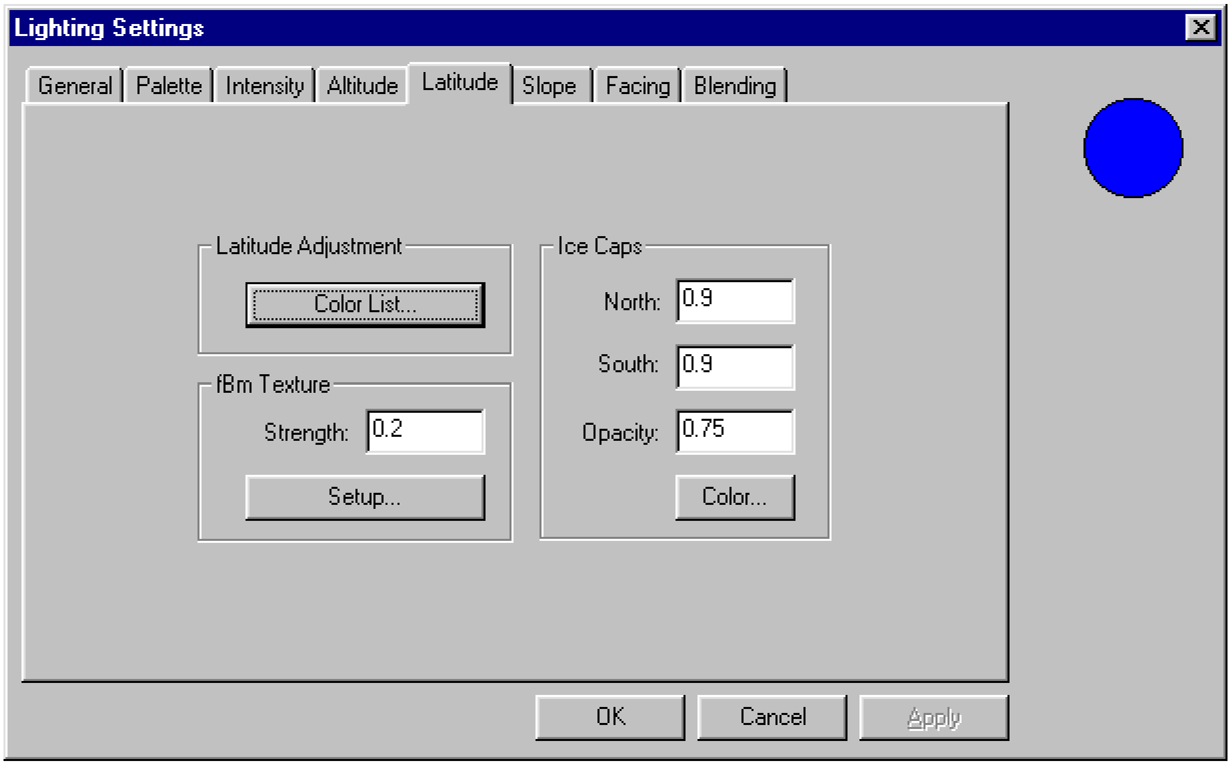
Opacity: 0

Abs Min: -1000

Lightness

Max: 0.3

Min: 0

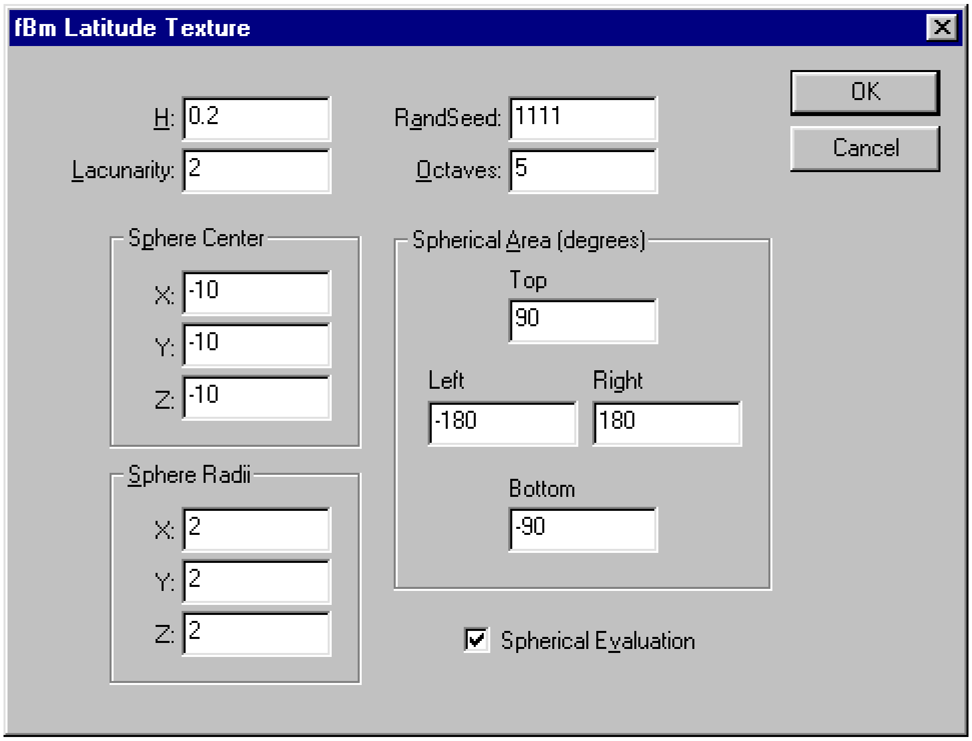


Lighting Settings

Latitude Adjustment

fBm Texture
Strength: 0.2

Ice Caps
North: 0.9
South: 0.9
Opacity: 0.75



fBm Latitude Texture

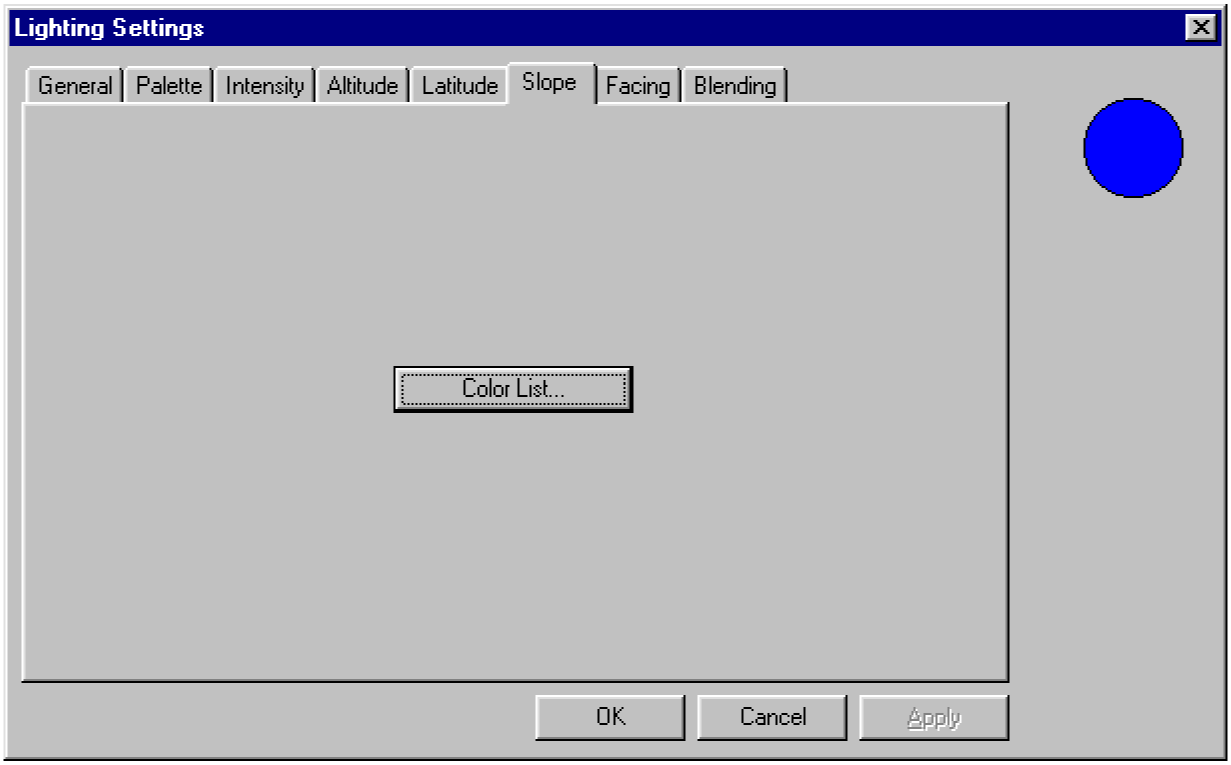
H: 0.2
RandSeed: 1111
Lacunarity: 2
Octaves: 5

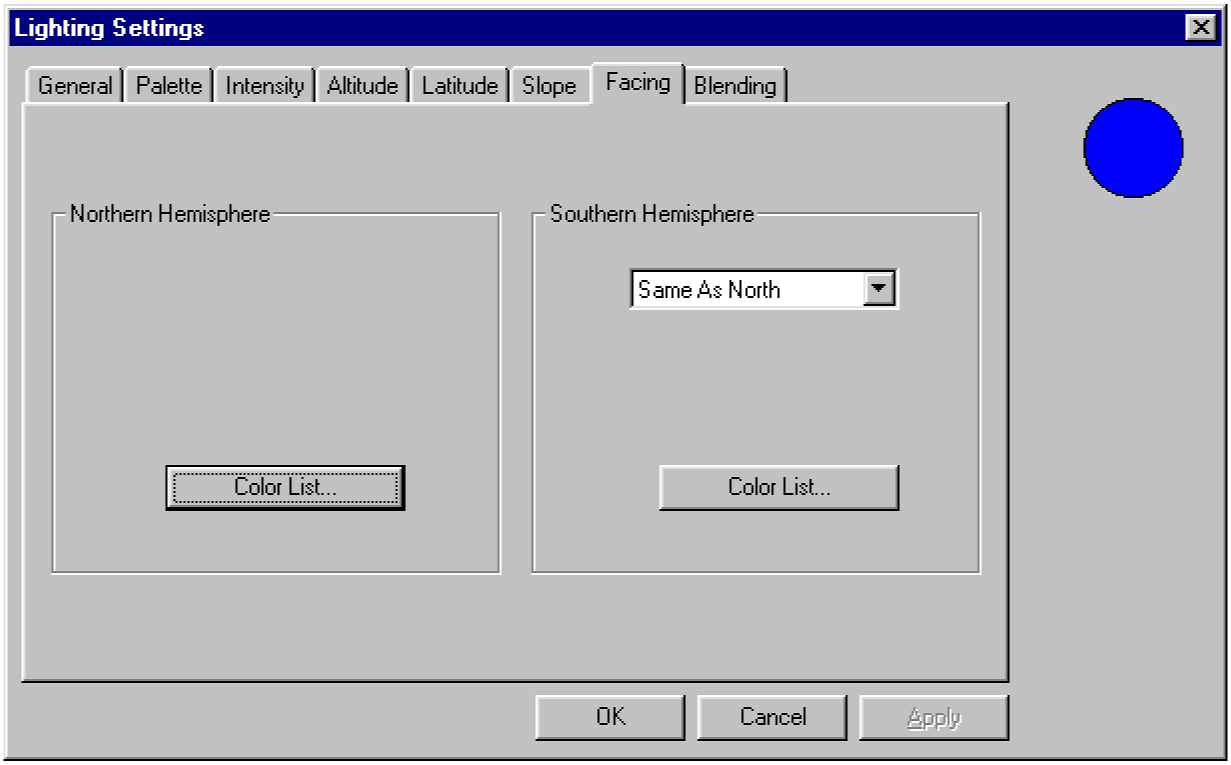
Sphere Center
X: -10
Y: -10
Z: -10

Sphere Radii
X: 2
Y: 2
Z: 2

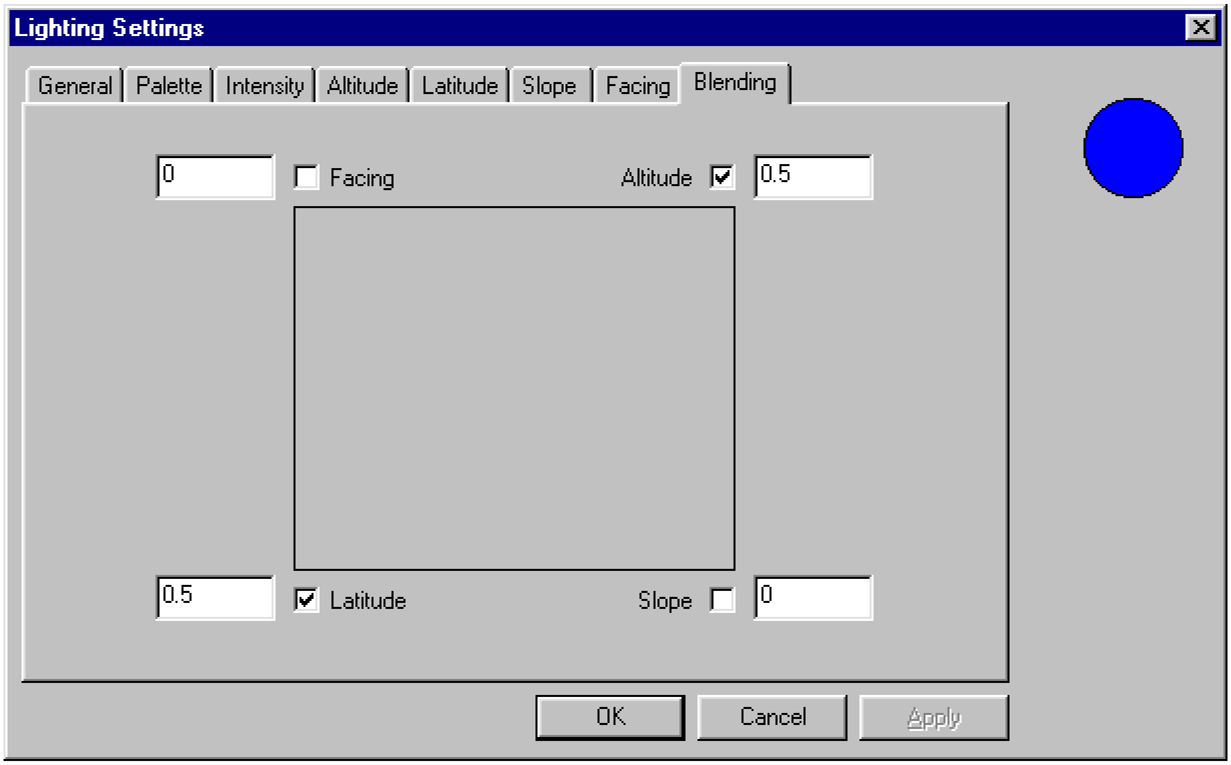
Spherical Area (degrees)
Top: 90
Left: -180
Right: 180
Bottom: -90

☑ Spherical Evaluation





Southern Hemisphere: Same As North

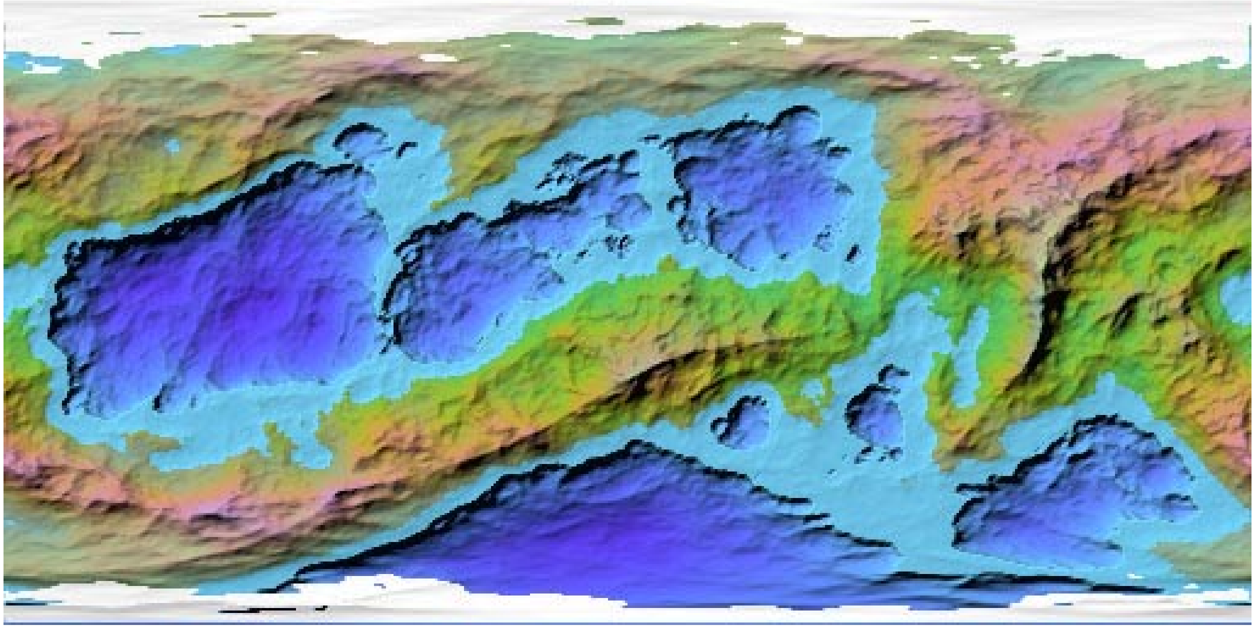


0 □ Facing

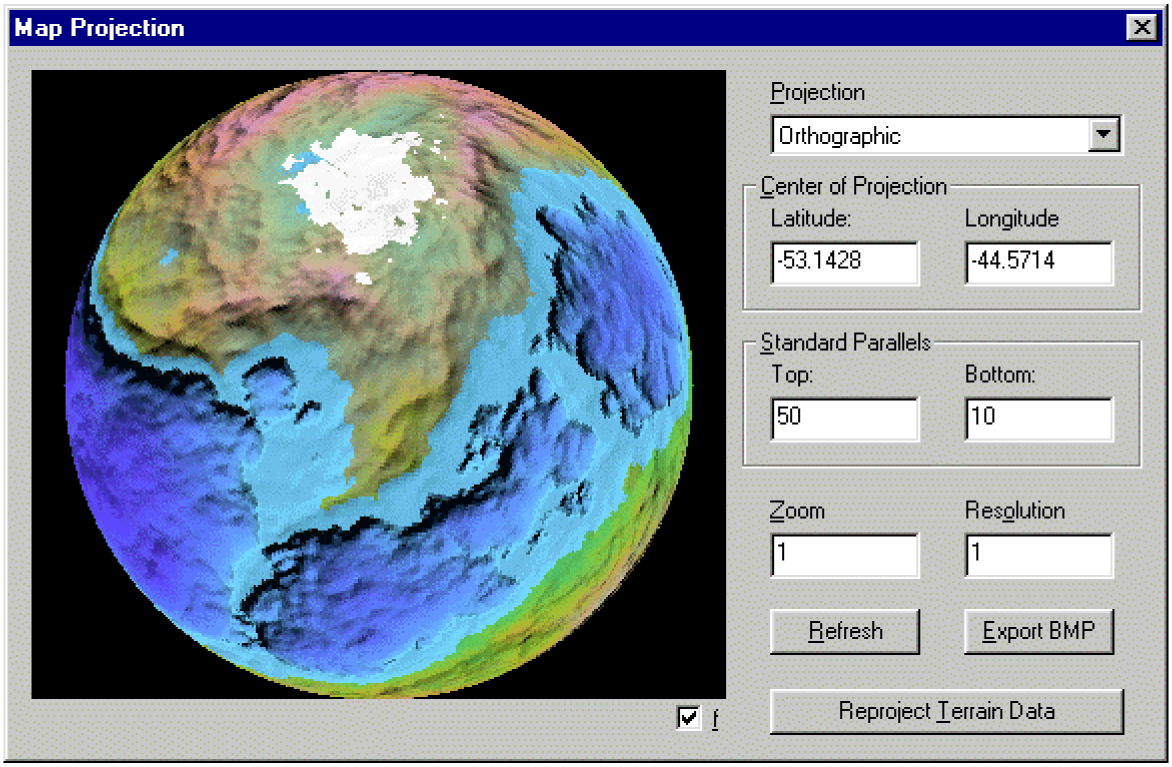
Altitude ☑ 0.5

0.5 ☑ Latitude

Slope □ 0



But what would this flat, distorted map look like on a globe, you might ask? There are many ways to accomplish this feat, but by far the simplest is to use Wilbur own *Misc->Map Projections* dialog. The output isn't the best, but it certainly gets the idea across. Shown below is sort of a north-polar-type view of the above map.



Projection: Orthographic

Center of Projection

Latitude: -53.1428

Longitude: -44.5714

Standard Parallels

Top: 50

Bottom: 10

Zoom: 1

Resolution: 1

As a side note, the exact ratio of sea to land can be controlled using the histogram display and the offset command. Typically, this operation will be performed before the exponential operation (but using it before the exponential operation will cause the amount of ocean to change after that operation is applied; for exact land/sea ratios, use after the exponential operation). Call up a histogram (*Surface->Histogram*) and move the cursor back and forth across the histogram data, noting the percentile value. This value is the percent of samples below the bin (height). If, for example, the 75th percentile is at 100, then applying an offset (*Surface->Point Process->Offset*) of -100 will force 75% of the data samples to be at or below sea level. In our example world above, adjusting things so that the world has a 75%:25% land:sea ratio gives a map like:

5. Simulated contour maps

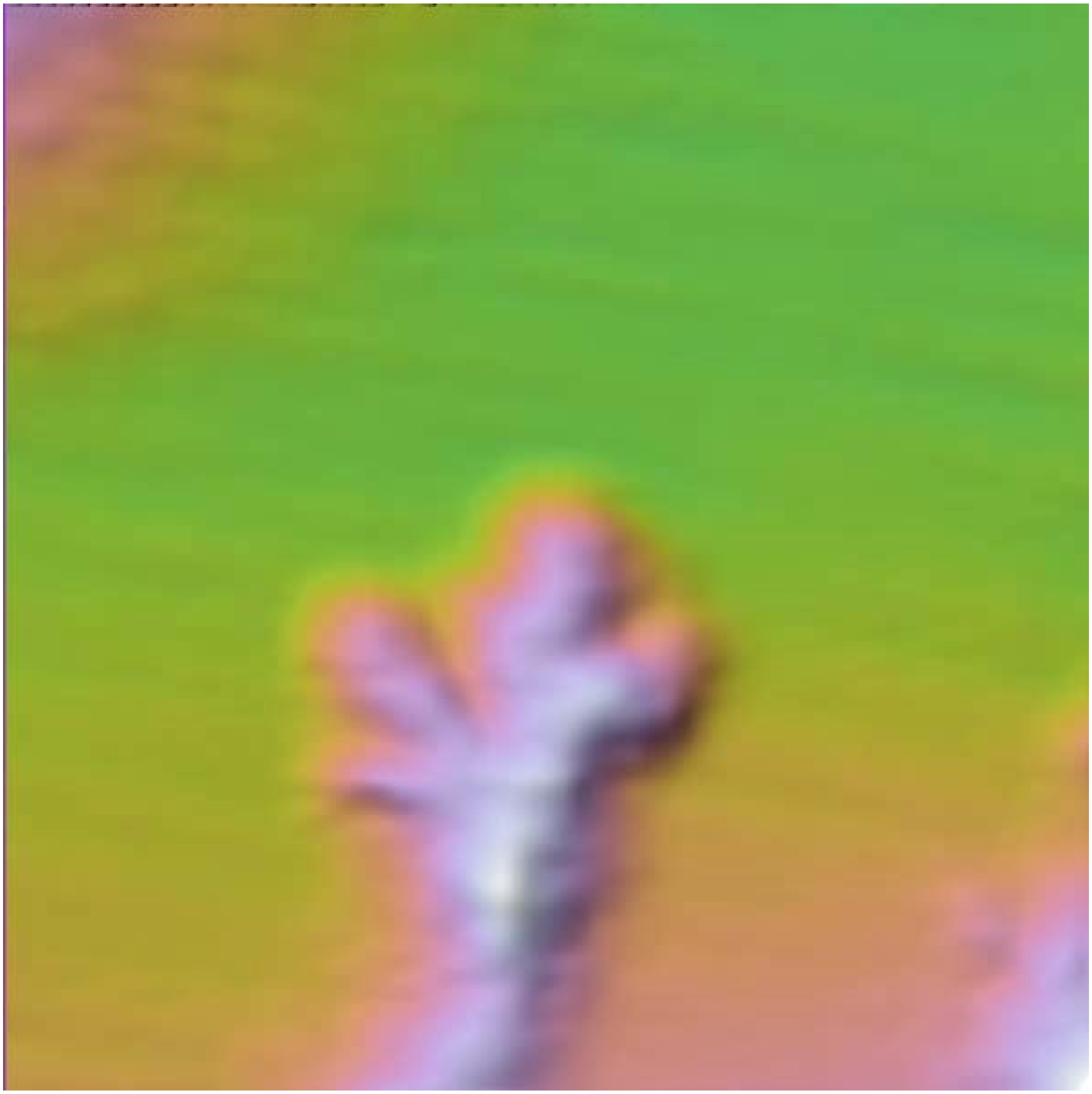
Sometimes, you need a simple contour-style map with no frills. While the current version of Wilbur won't output a true vector-format contour map (except for one oddball CAD program), it can simulate one using a bitmap.

For this example, we'll use a tiny (and I mean tiny) area of the USGS data set: Black Mountain at China Lake, CA. We'll take 100 meter USGS data and convert it to a contour map with 5 meter intervals. Then we'll do a little processing with Photoshop to get the final result.

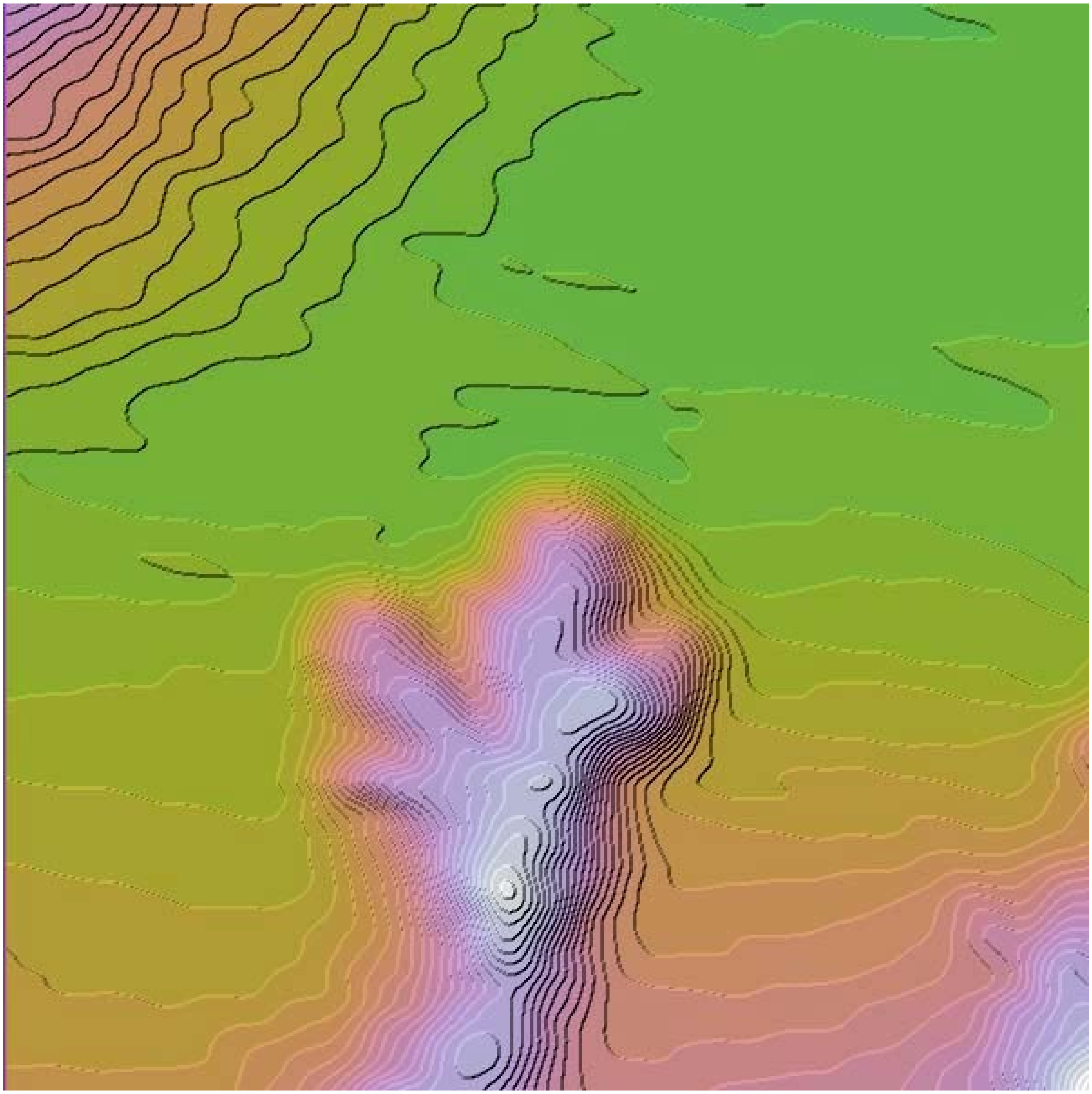
The core concept here is to take the data set and posterize it (which converts it to stairsteps), then shine a light straight down from above and tell the system that there's an enormous vertical exaggeration. What will happen is that wherever there's a change in the height (at the edge of the steps), there will be a black line; the rest will be colored however we defined it. Photoshop can be used to combine two or more maps with differing posterization values to get a more pleasing result.

Import the data set. I used a small (64x64 data set) to get really smooth contour intervals (also because the area I'm interested in is fairly small).

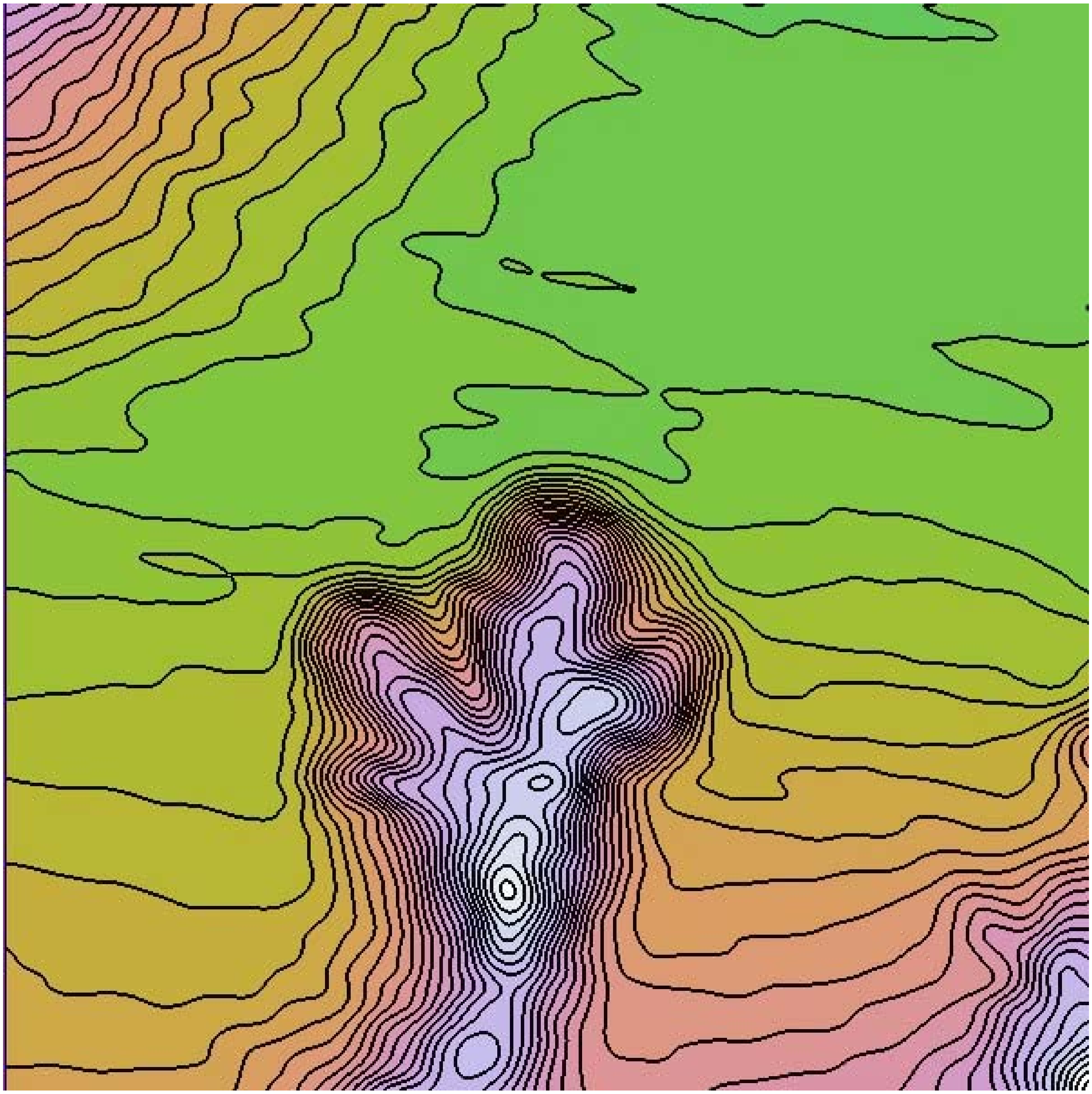
Resample to a resolution you're comfortable with (not really required unless you want to). I used 600x600 with a B-spline approximation for a good working resolution.



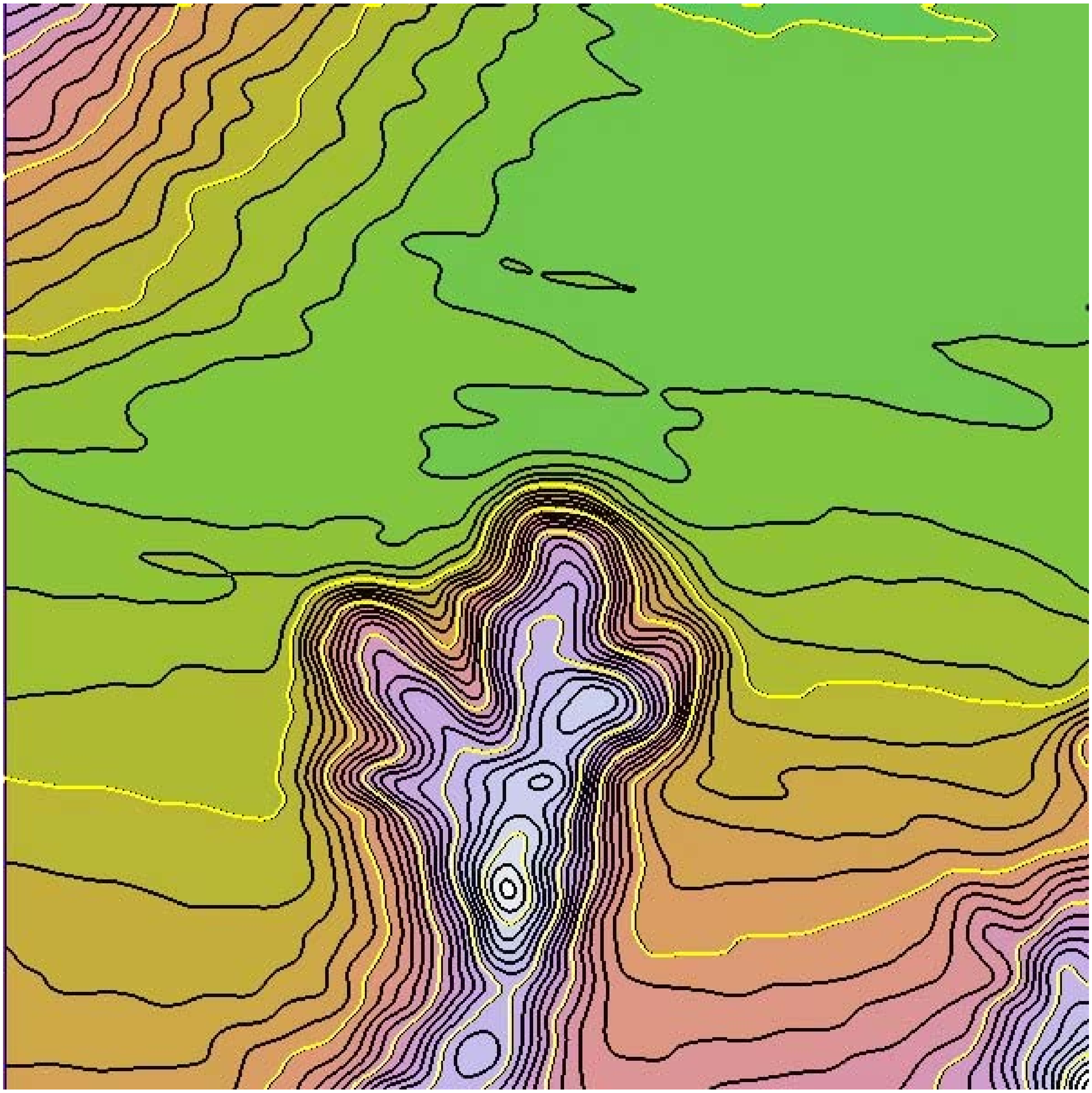
Do a threshold operation with the desired contour interval (10 meters in this case). Note that it's already starting to look good.



Reset the lighting parameters to 90 degrees elevation and 10000 vertical exagerration. The rest of the parameters don't matter much. Looks pretty good.



For a final touch, reposterize at say 1/5 interval (50 meter in this case), then combine the two in Photoshop to get a very nice contour interval with large and small variations. Another fun way to play is to overlay the contour maps on the basic texture map. When doing multiple overlays with Photoshop, it's best to calculate the overlay contour images as lighted gray scale so that there's no spurious color information to get in the way.



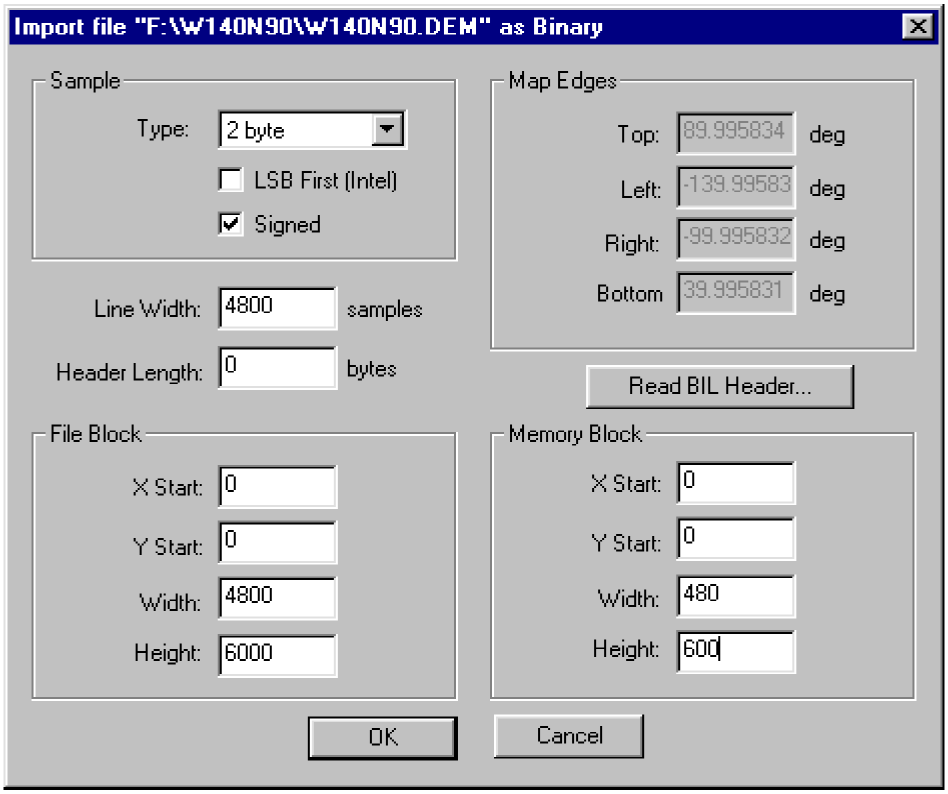
6. Stitching together multiple binary files

People seem to be having difficulties with the concept that the File->Import->Image Subsection command can be used to replace just portions of a map. In this example, we'll use 4 files from the GTOPO30 data set to make a single image that covers the contiguous United States. We'll also get a bunch of other junk in the image, but that's not terribly important here.

Locate the 4 files required. In this case, they are W100N90.DEM (the northeast image), W100N40.DEM (the southeast image), W140N90 (the northwest image), and W140N40 (the southwest image). As indicated in the tutorial that combines GTOPO30 data with TerrainBase data, open and examine each of the four files. To simplify things, we'll just import the whole files at 1/10 resolution and. Because the basic files are 4800x6000 pixels, we'll start with a 960x1200 image (1/10 of (4800+4800) because there are two images across and 1/10 of (6000+6000) because there are two images high.

Surface->Size, set size to 960 width and 1200 height

Import the upper left map section using File->Import->Image Subsection



Import file "F:\W140N90\W140N90.DEM" as Binary

Sample

Type: 2 byte

☐ LSB First (Intel)

☑ Signed

Line Width: 4800 samples

Header Length: 0 bytes

File Block

X Start: 0

Y Start: 0

Width: 4800

Height: 6000

Map Edges

Top: 89.995834 deg

Left: -139.99583 deg

Right: -99.995832 deg

Bottom 39.995831 deg

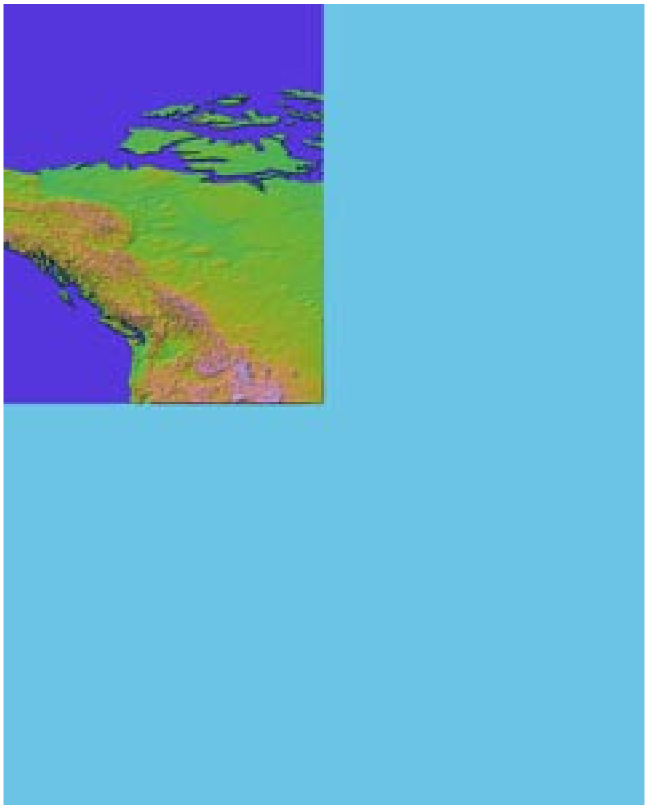
Memory Block

X Start: 0

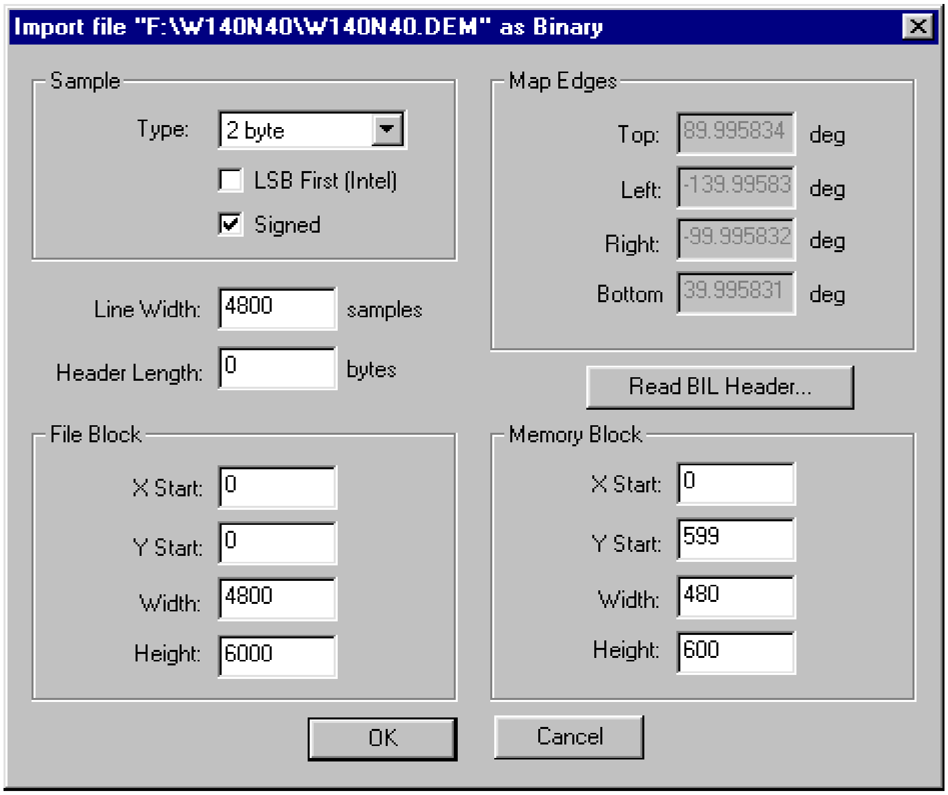
Y Start: 0

Width: 480

Height: 600



Import the lower left map section using File->Import->Image Subsection



Import file "F:\W140N40\W140N40.DEM" as Binary

Sample

Type: 2 byte

☐ LSB First (Intel)

☑ Signed

Line Width: 4800 samples

Header Length: 0 bytes

File Block

X Start: 0

Y Start: 0

Width: 4800

Height: 6000

Map Edges

Top: 89.995834 deg

Left: -139.99583 deg

Right: -99.995832 deg

Bottom 39.995831 deg

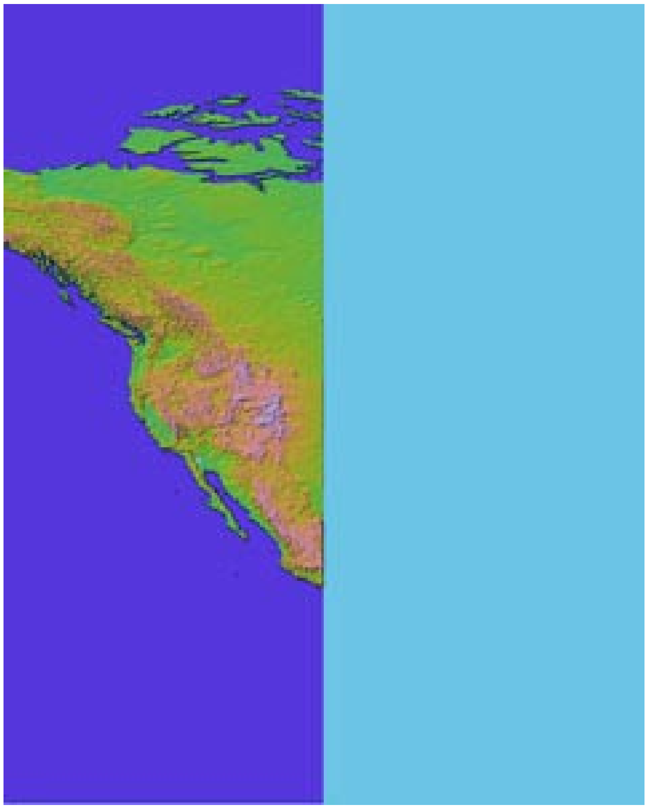
Memory Block

X Start: 0

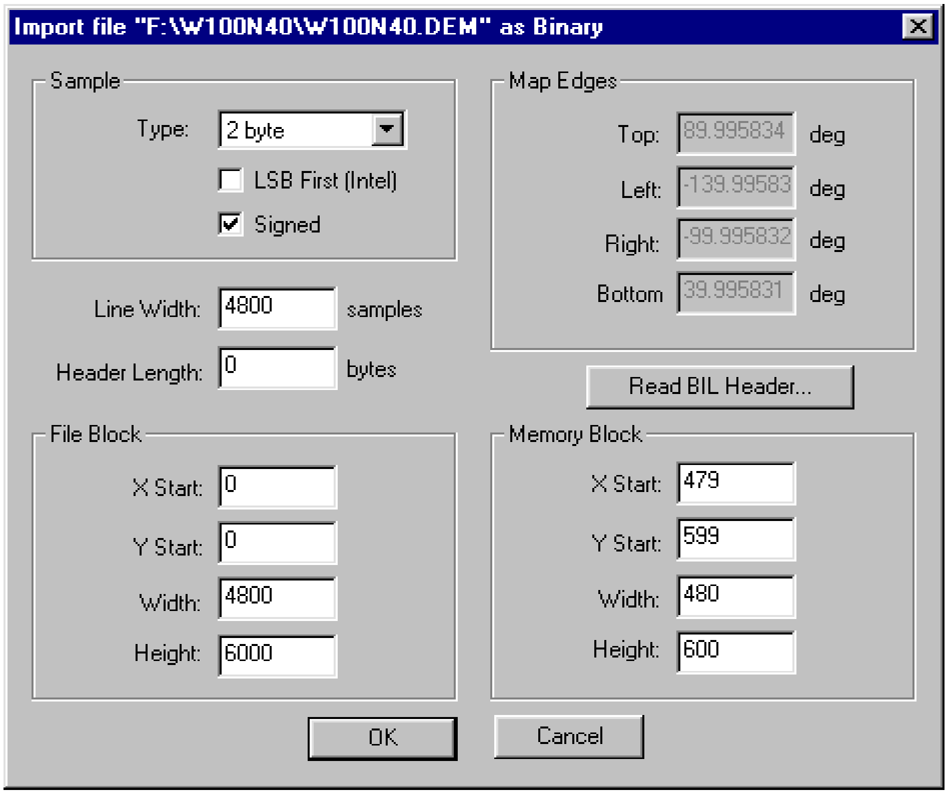
Y Start: 599

Width: 480

Height: 600



Import the lower right map section using File->Import->Image Subsection



Import file "F:\W100N40\W100N40.DEM" as Binary

Sample

Type: 2 byte

☐ LSB First (Intel)

☑ Signed

Line Width: 4800 samples

Header Length: 0 bytes

File Block

X Start: 0

Y Start: 0

Width: 4800

Height: 6000

Map Edges

Top: 89.995834 deg

Left: -139.99583 deg

Right: -99.995832 deg

Bottom 39.995831 deg

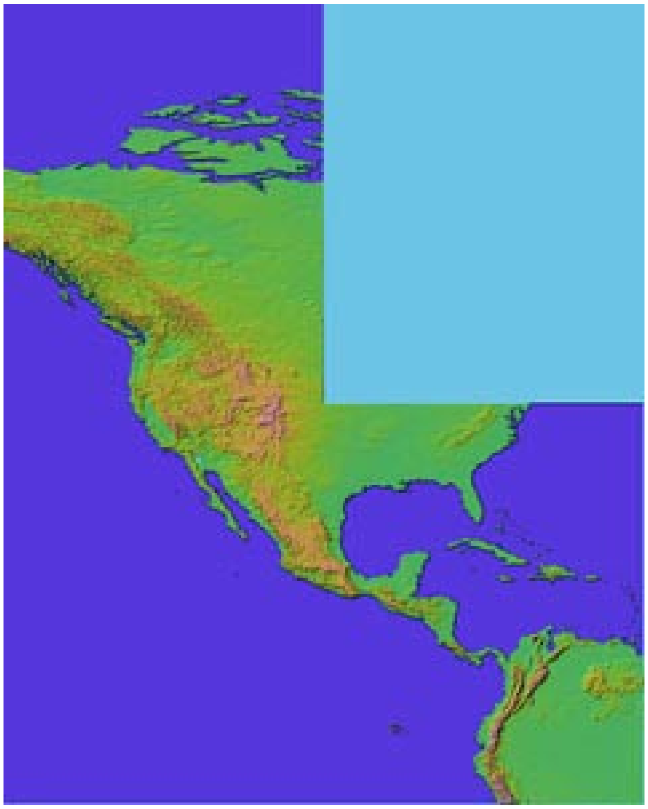
Memory Block

X Start: 479

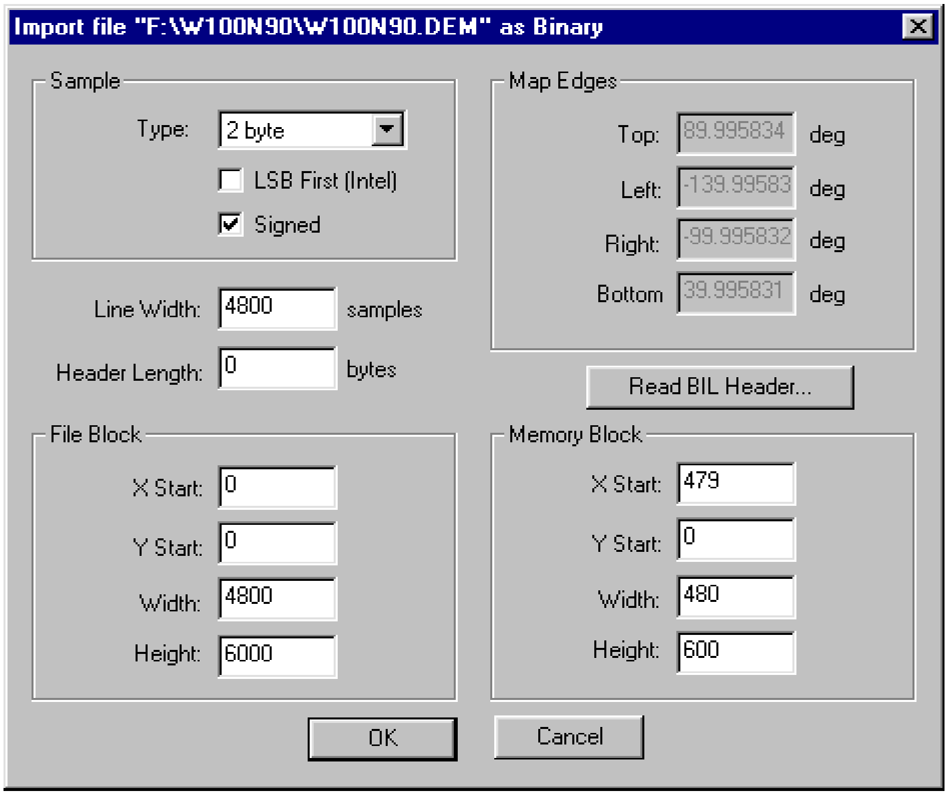
Y Start: 599

Width: 480

Height: 600



Import the upper right map section using File->Import->Image Subsection



Import file "F:\W100N90\W100N90.DEM" as Binary

Sample

Type: 2 byte

□ LSB First (Intel)

☑ Signed

Line Width: 4800 samples

Header Length: 0 bytes

Map Edges

Top: 89.995834 deg

Left: -139.99583 deg

Right: -99.995832 deg

Bottom 39.995831 deg

File Block

X Start: 0

Y Start: 0

Width: 4800

Height: 6000

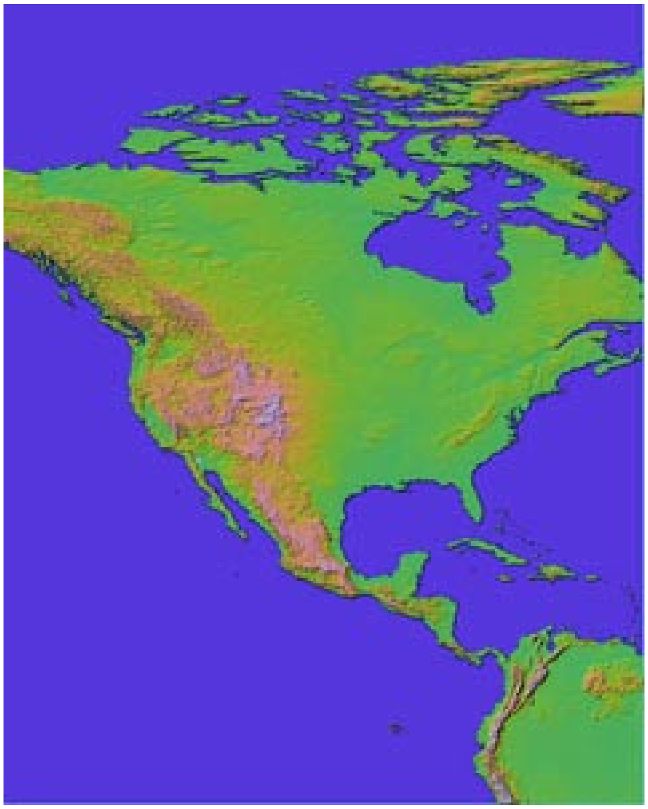
Memory Block

X Start: 479

Y Start: 0

Width: 480

Height: 600



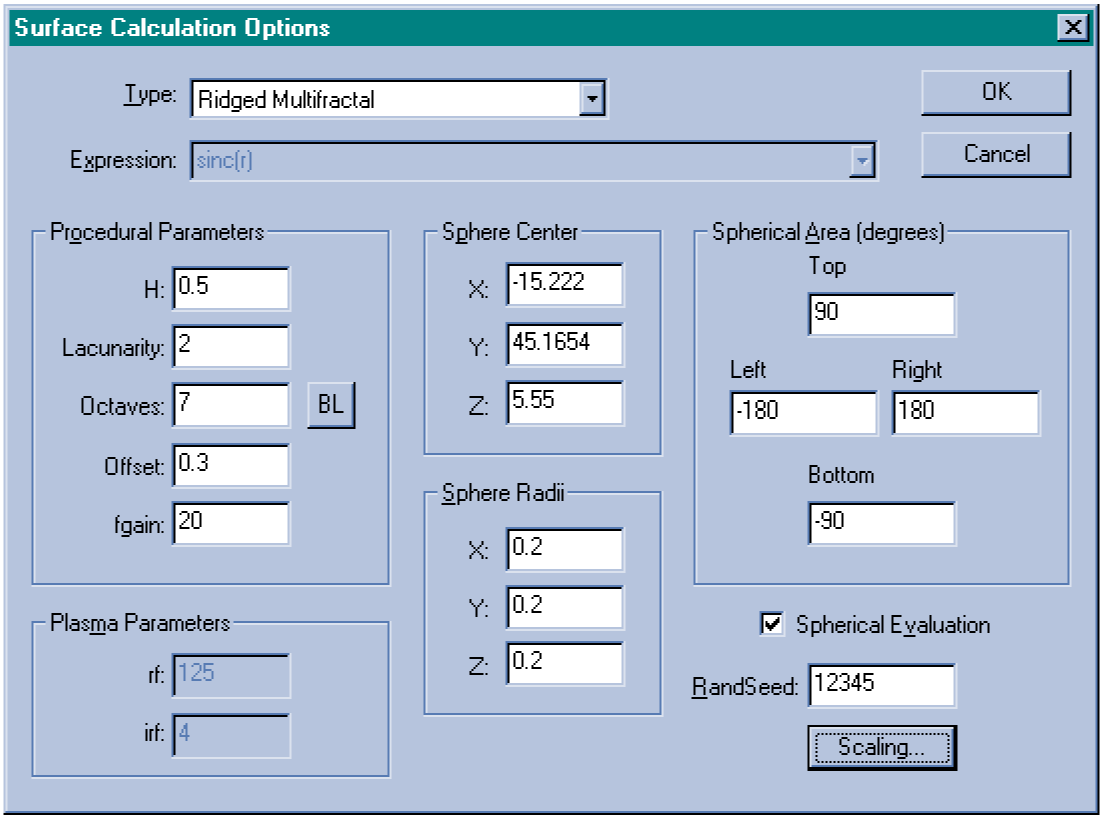
Save the image as an MDR file or some such.

Please note that it’s possible to extract just the portions of the source files that you want by using the “File Block” section (which controls where the data comes from in the file) in addition to the “Memory Block” section (which controls where the data goes when it’s in memory).

7. Making a world with Wilbur and Campaign Cartographer 2

This page assumes the presence of version 1.23 or higher of Wilbur and Campaign Cartographer 2 (V5.21 or higher).

- Start Wilbur
- Set an image size to 512x256 samples and 90 for the top, -90 for the bottom, 180 for the right, and -180 for the left using the File->New command.
- Calculate a new surface using the Surface->Calculate Height field menu option. Set the dialog to the settings shown below.



Surface Calculation Options

Type: Ridged Multifractal

Expression: sinc(r)

Procedural Parameters

H: 0.5
Lacunarity: 2
Octaves: 7
Offset: 0.3
fgain: 20

Plasma Parameters

rf: 125
irf: 4

Sphere Center

X: -15.222
Y: 45.1654
Z: 5.55

Sphere Radii

X: 0.2
Y: 0.2
Z: 0.2

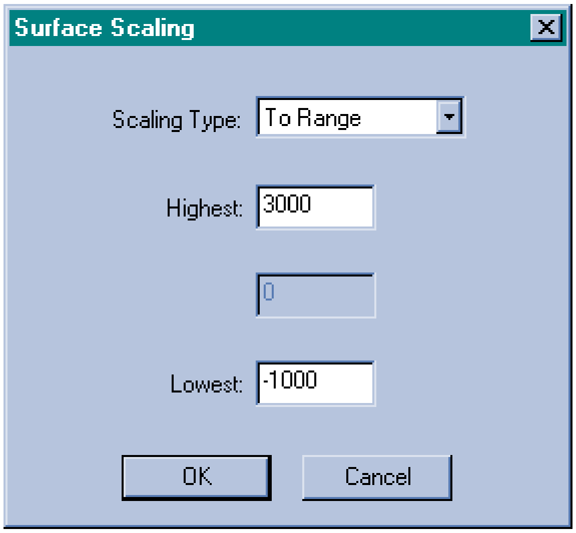
Spherical Area (degrees)

Top: 90
Left: -180
Right: 180
Bottom: -90

☑ Spherical Evaluation

RandSeed: 12345

Press the Scaling button, enter the settings shown below, and press OK. The dialog above will come back and then press OK on that one.



Surface Scaling

Scaling Type: To Range

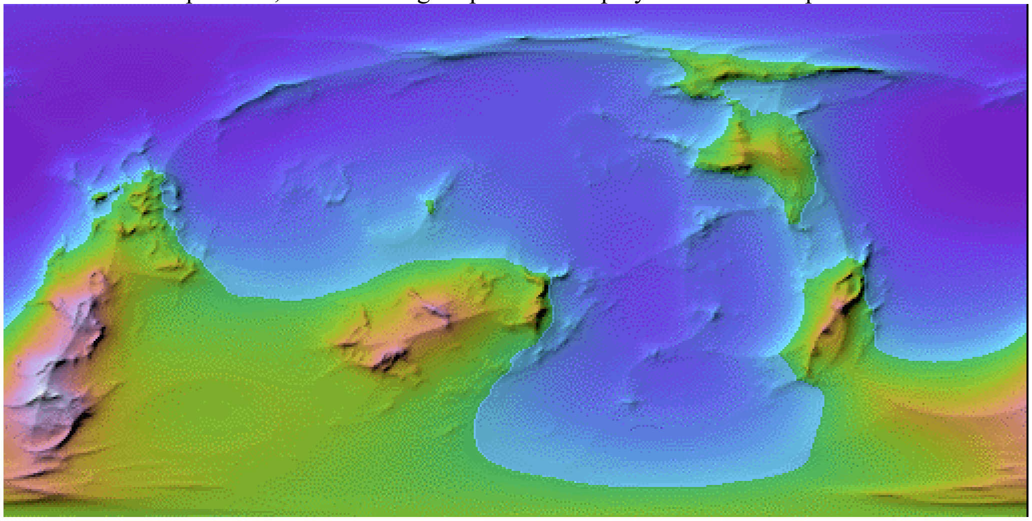
Highest: 3000

0

Lowest: -1000

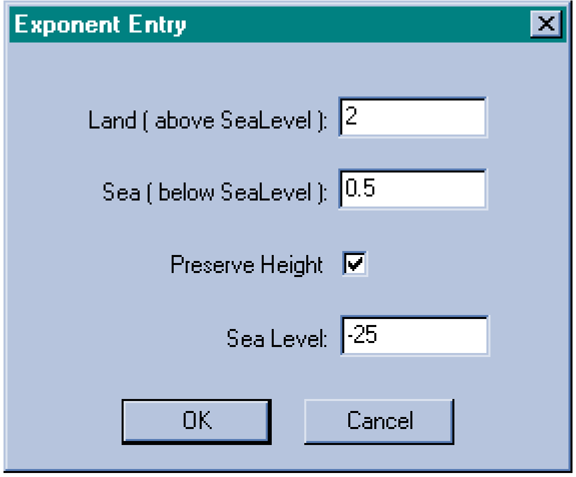
These settings will control the appearance of the world. For example, each value entered for the RandSeed parameter will generate a completely new world. For each RandSeed Value Sphere Center X,Y,Z specify the center of a sphere (of radii Sphere Radii X,Y,Z). The Spherical Area parameters allow a portion of the current settings to be zoomed in on. We'll do that later.

- After a little computation, the following map will be displayed in the main part of the window:



Looks pretty good, but we can do better. If you look at a globe, you'll notice that there are things called continental shelves. We'll simulate some.

- Select Surface->Point Process->Exponent. Enter the settings below:



Exponent Entry

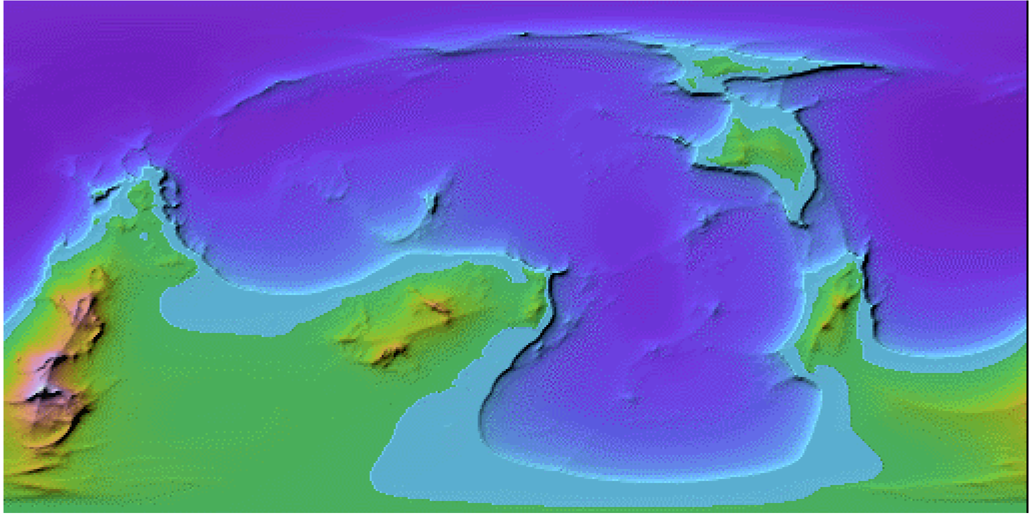
Land (above SeaLevel): 2

Sea (below SeaLevel): 0.5

Preserve Height ☑

Sea Level: -25

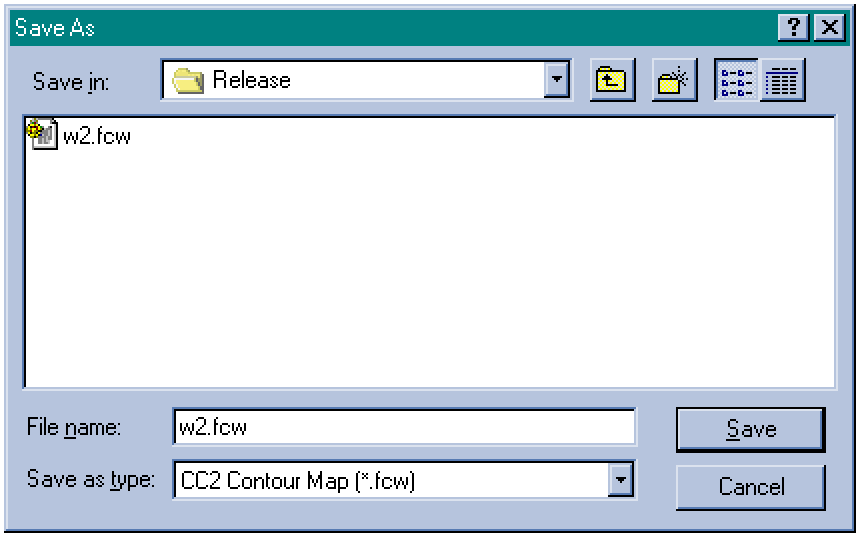
After a little computation, we find that the surface has changed to the following:



Looks good enough to me. We'll output that one to disk as a bitmap and as a CC2 contour file.

- Select File->Save As. On the Save as Type drop list at the bottom of the dialog, select "BMP Texture" then enter a file name and click Save.

- Select File->Save As. On the Save as Type drop list at the bottom of the dialog, select "CC2 Contour Map", then enter a File Name to save as shown below:



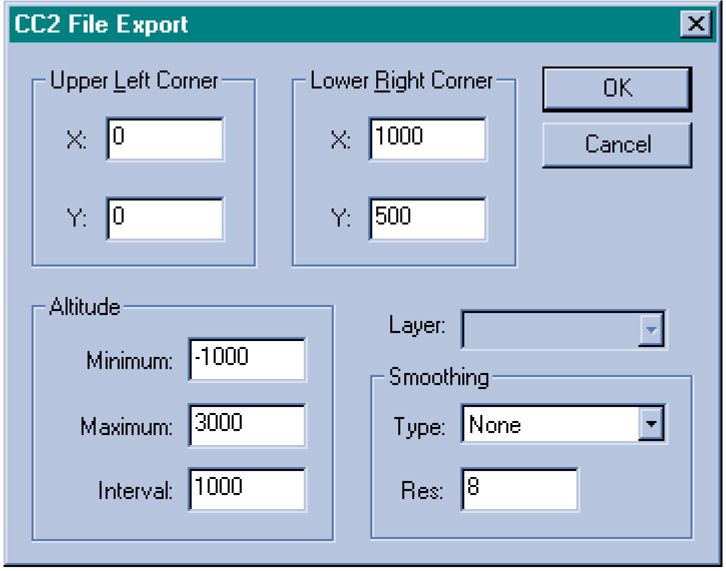
Save in: Release

w2.fcw

File name: w2.fcw

Save as type: CC2 Contour Map (*.fcw)

After you press Save, another dialog will pop up asking for more information. Enter the info shown below:



Upper Left Corner
X: 0
Y: 0

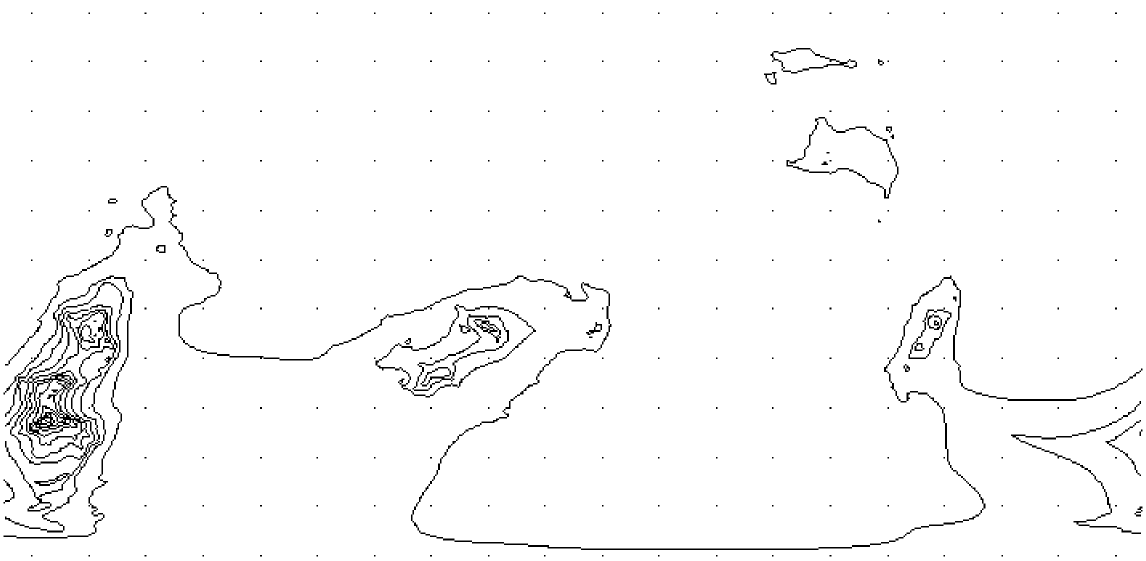
Lower Right Corner
X: 1000
Y: 500

Altitude
Minimum: -1000
Maximum: 3000
Interval: 1000

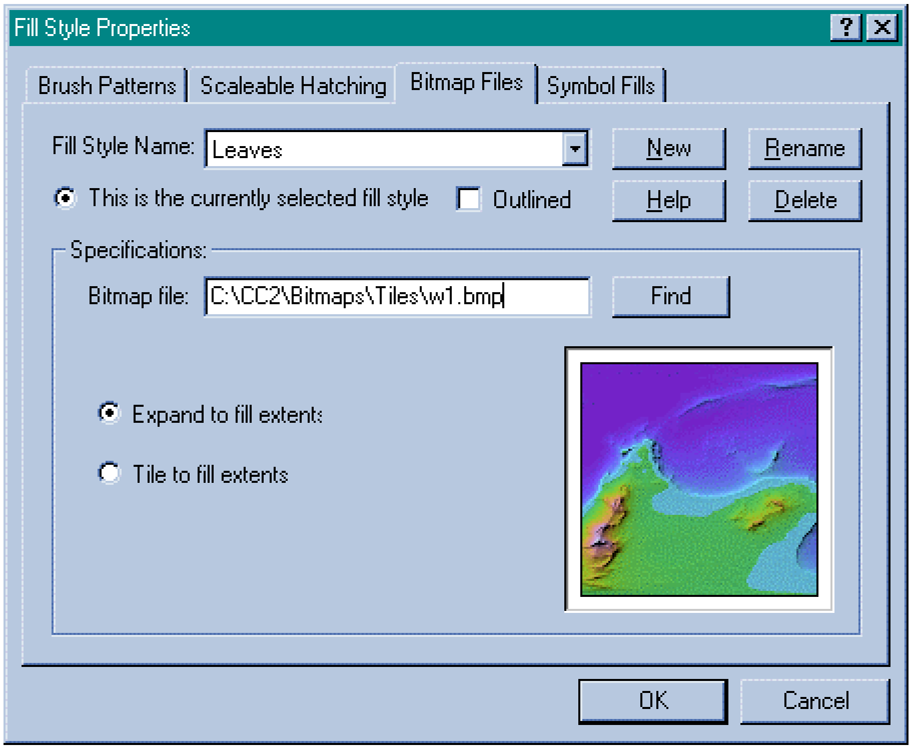
Layer:

Smoothing
Type: None
Res: 8

- Start CC2 and open the file that you saved. The drawing portions of CC2 should show something like that shown below:



- Now we want to put the BMP image that we saved as the background for the contours.
- Draw a box from 0,0 to 1000,500 (quickest way is to type "box 0,0 1000,500" without the quotes). This box will tightly bound the world. I'll try to remember to draw the first and box automatically in the next version.
- Change the fill style to a bitmap that fills the box (quickest way is to type "change f e" select the box, hit return, then hit return again, and then again one last time to get the fill styles dialog to appear). Set up the fill style to use the bitmap that you saved and make sure the "Expand to fill extents" button is selected, as is the "This is the currently selected fill style" button. The dialog should be set up as shown below:



Fill Style Properties

Brush Patterns | Scaleable Hatching | Bitmap Files | Symbol Fills

Fill Style Name: Leaves

● This is the currently selected fill style □ Outlined

Specifications:

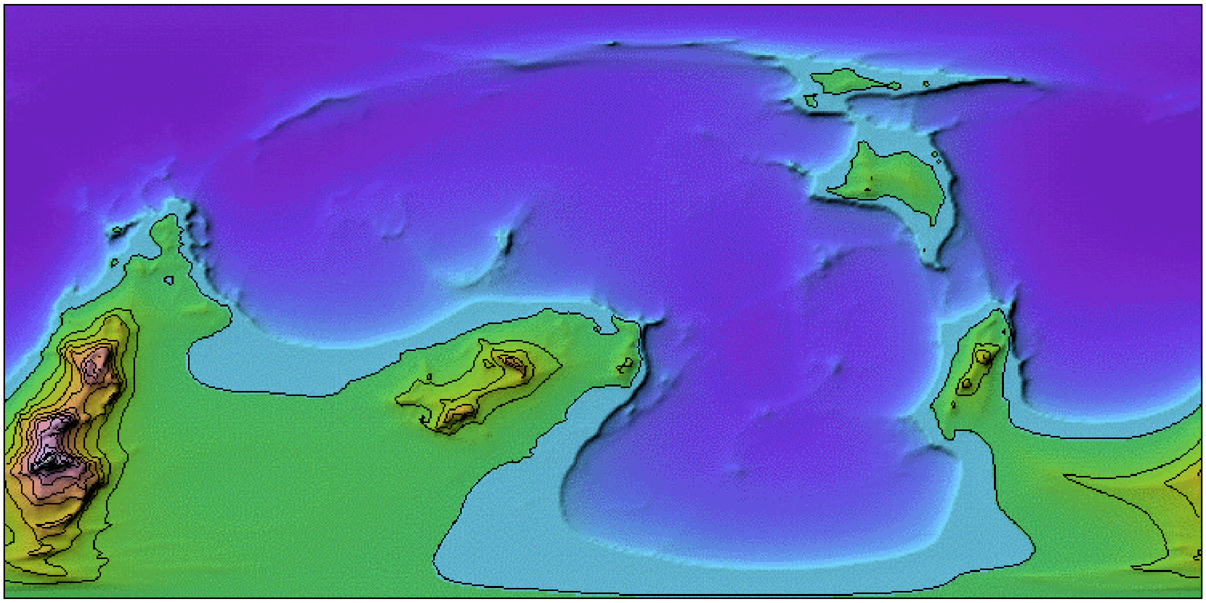
Bitmap file: C:\CC2\Bitmaps\Tiles\w1.bmp

● Expand to fill extents

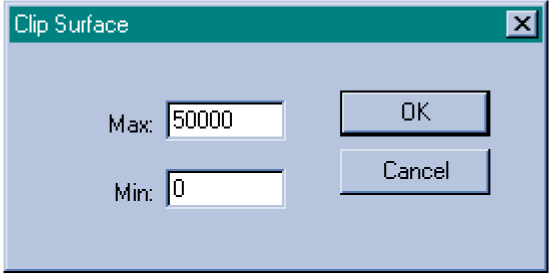
○ Tile to fill extents

- Send that box to the back using the Edit->Back command and applying it to the box just drawn. If you don't have a back command on the Edit menu, you don't have the current patch version.

- The final result should be as shown below. It's a lot of work, but the results are pretty nice.



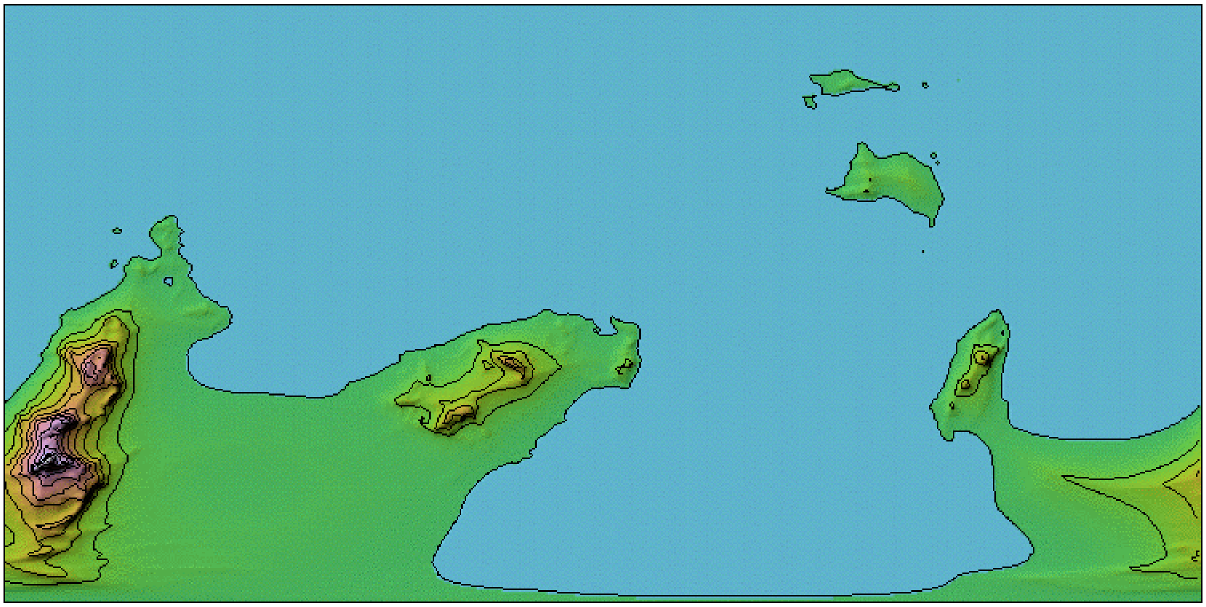
- If you don't want to see the ocean depths do the following before saving the files in Wilbur: Select Surface->Point Process->Height Clip and enter the parameters shown below:



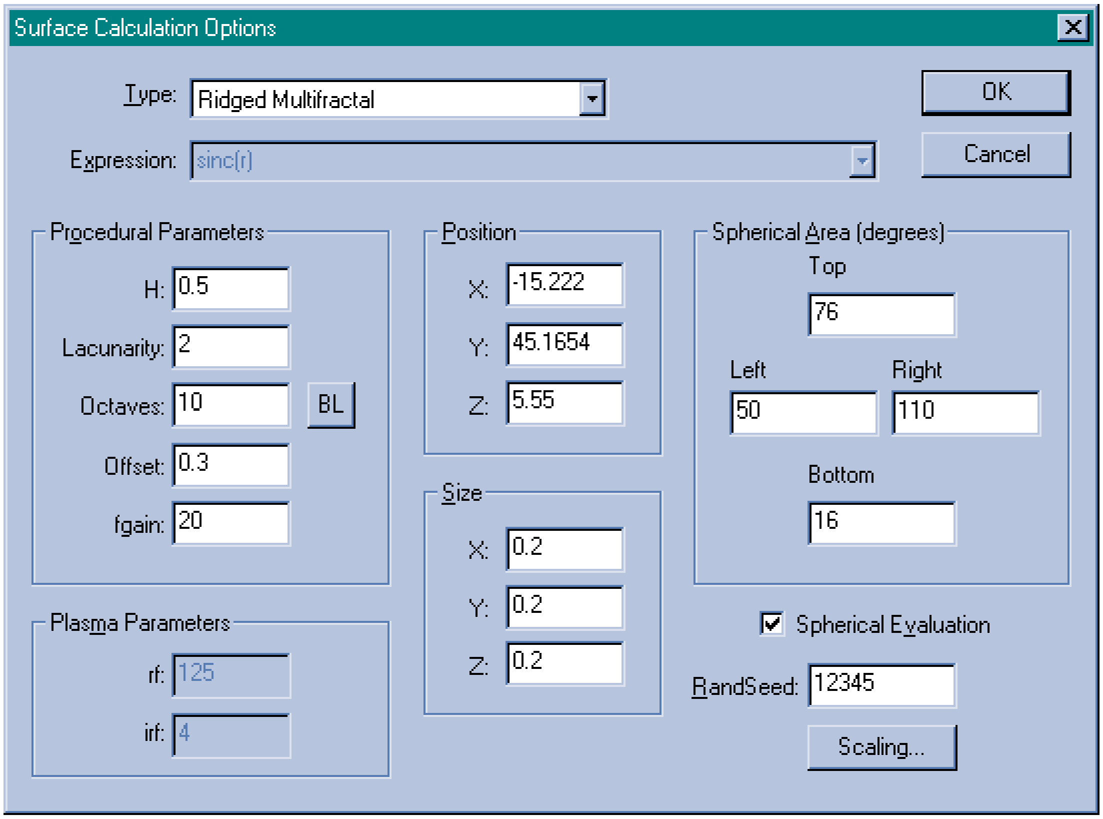
Max: 50000

Min: 0

After pressing OK, every point above 50000 will be replaced with 50000 (and since all points on this surface are below 50000, none of them will be clipped off), and every point below 0 (sea level) will be replaced with 0. Save the files as normal. After the rest of the processing, the results should look like that shown below.



- For a little magic, let's zoom in on the island group in the map. We won't have the heights exactly correct in this demonstration, but it will get the point across.
- Set an image size to 512x512 samples and 90 for the top, -90 for the bottom, 180 for the right, and -180 for the left using the File->New command.
- Calculate a new surface using the Surface->Calculate Height field menu option. Set the dialog to the settings shown below (and set the scaling to -650 and +700).



Surface Calculation Options

Type: Ridged Multifractal

Expression: sinc(r)

Procedural Parameters

H: 0.5

Lacunarity: 2

Octaves: 10

Offset: 0.3

fgain: 20

Plasma Parameters

rf: 125

irf: 4

Position

X: -15.222

Y: 45.1654

Z: 5.55

Size

X: 0.2

Y: 0.2

Z: 0.2

Spherical Area (degrees)

Top: 76

Left: 50

Right: 110

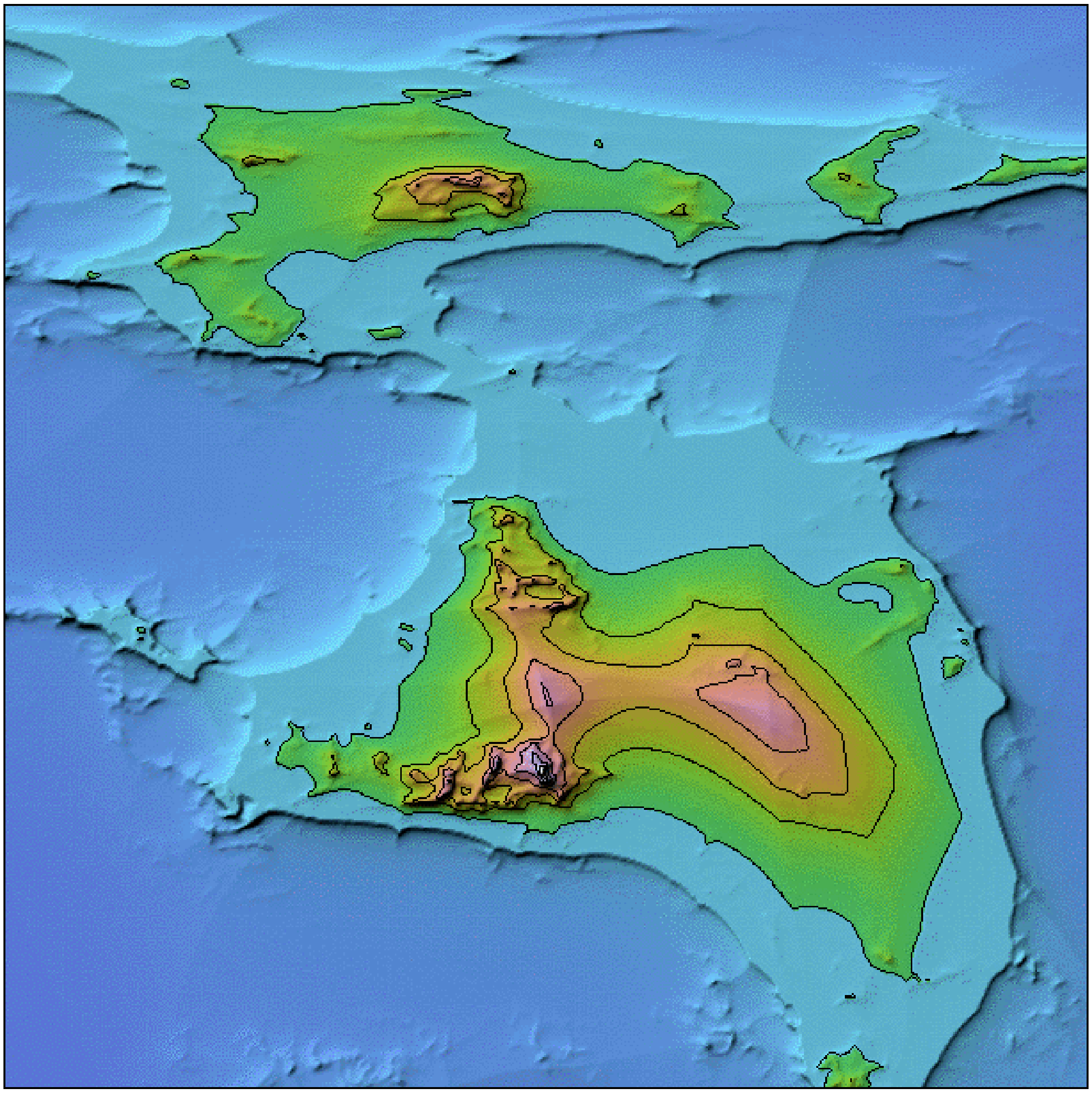
Bottom: 16

☑ Spherical Evaluation

RandSeed: 12345

Note that the only difference between this one and the one at the start of this document is that this one has a different spherical area selected and a higher value for octaves. In general, octaves should go up as the zoom area gets smaller. This parameter controls the roughness of the surface.

- After the rest of the processing (use 0,0 and 511,511 for the CC2 export and drawing the box in CC2), the final map will look something like that shown below:



That's all well and good, you say, but where do I get this software? Wilbur is available free of charge at http://www.ridgenet.net/~jslayton/software.html and Campaign Cartographer 2 is available at http://www.profantasy.com.

8. Fun with expressions

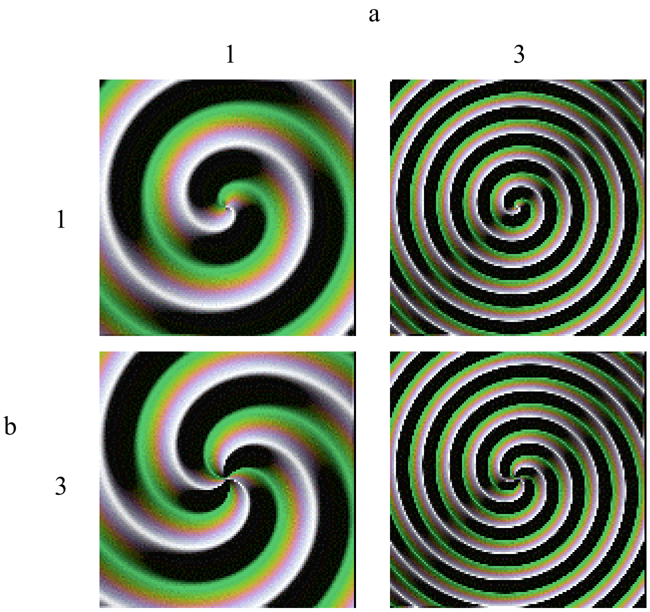
The expression ("Math Function") setting in the texture generation is perhaps the most amusing part of the system. Many odd effects can be generated. These same effects may be turned into brushes using the advanced brush form.

8.1. Spirals

Start: -5,-5,-5; Width: 10,10,10

Expr: cos(r*a+t*b) where a and b are values or expressions that control the shape of the spiral. A controls the number of loops available in a given area (a=1 means 1 turn/pi) while b controls the numbers of arms on the spiral.

Examples:



8.2. Radial Attenuation

May times, it is a good idea to taper the value of the surface off from the center point. The following expressions all result in a nice attenuation around 0.

Expr: rmax-r gives a conical falloff. The half-way point (value of 0.5) is at 0.5.

Expr: rmax-bias(r,n) give a nice falloff that can be adjusted from really pointy (n near 1), through a cone (n=0.5), to really flat and rounded (n near 0). The halfway point varies depending on the value of n.

Expr: exp(-r) gives a nice falloff that's "pointy" at the center. The half-way point (value of 0.5) is at about 0.7.

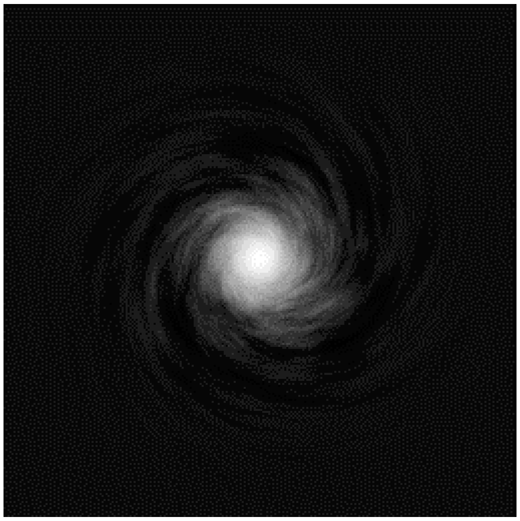
Expr: exp(-r*r) gives a rounded (roughly gaussian) falloff. The half-way point (value of 0.5) is at about 0.84.

8.3. Spiral Galaxy

Expr: fbm(xrot(sqrt(r*10),1,x,y),yrot(sqrt(r*10),1,x,y),r)*exp(-r*r/5)*(1-exp(-r*r))+exp(-r*r)*2

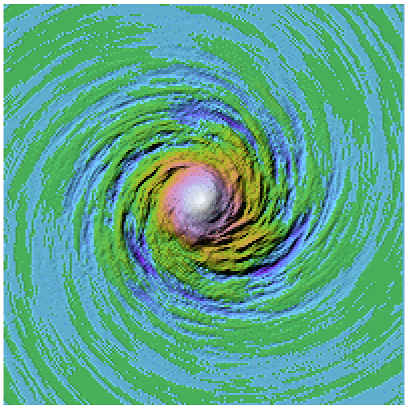
Start: -5,-5,-5; Width: 10,10,10

Generates the "spiral galaxy" shown below:



This image above was scaled to -1 and +1 (min/max) and colored using height coding with an altitude color list of one black and one white color (gamma 0.75).

The image below occurs with the "normal" terrain coloration:



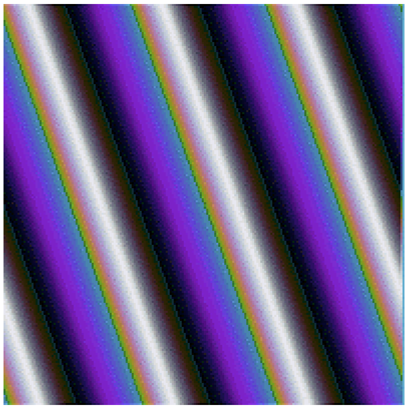
8.4. Corrugations

Sometimes, a simple little corrugated roof is desired. The general formula is

Expr: sin(cos(alpha*pi/180)*x+sin(alpha*pi/180)*y)

This little expression rotates the corrugations to the angle alpha (in degrees). Similar results can be obtained with cos as the first expression component.

An example of the results evaluated in the XY plane from -5,-5 to +5,+5 is shown below:

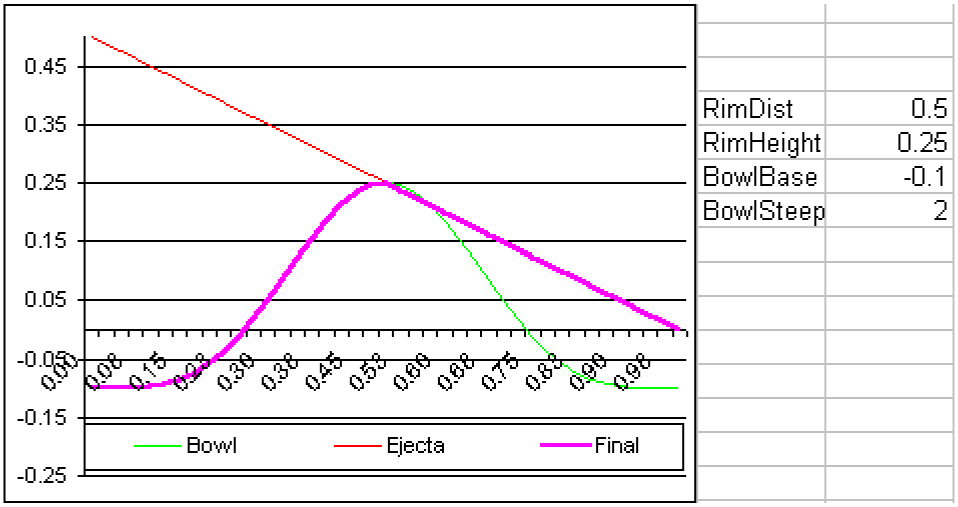


8.5. Craters

Had to be craters, didn't there? The crater function in use is excessively complex for the poor results it gives (much like the rest of this program). Basically, it is based on four parameters:

Parameter	Description
RimDist	Gives the distance from the center of the crater to the highest point on the rim.
RimHeight	Gives the height of the rim relative to the baseline.
BowlBase	Gives the depth of the lowest point of the crater bowl relative to the baseline
Steep	An exponential value that affects the steepness of the inner crater walls. The inner walls are basically calculated as sin(r)^Steep.

A quick diagram should explain it all (or confuse the issue hopelessly):



RimDist	0.5
RimHeight	0.25
BowlBase	-0.1
BowlSteep	2