

Chapter 10

Mandelbrot Set

Fractals, topology, complex arithmetic and fascinating computer graphics.

Benoit Mandelbrot is a Polish/French/American mathematician who has spent most of his career at the IBM Watson Research Center in Yorktown Heights, N.Y. He coined the term *fractal* and published a very influential book, *The Fractal Geometry of Nature*, in 1982. An image of the now famous Mandelbrot set appeared on the cover of *Scientific American* in 1985. This was about the time that computer graphical displays were first becoming widely available. Since then, the Mandelbrot set has stimulated deep research topics in mathematics and has also been the basis for an uncountable number of graphics projects, hardware demos, and Web pages.

To get in the mood for the Mandelbrot set, consider the region in the complex plane consisting of the values z_0 for which the trajectories defined by

$$z_{k+1} = z_k^2, \quad k = 0, 1, \dots$$

remain bounded as $k \rightarrow \infty$. It is easy to see that this set is simply the unit disc, $|z_0| \leq 1$, shown in figure 10.1. If $|z_0| \leq 1$, the repeated squares remain bounded. If $|z_0| > 1$, the repeated squares are unbounded. The boundary of the unit disc is the unit circle, $|z_0| = 1$. There is nothing very difficult or exciting here.

The definition of the Mandelbrot set is only slightly more complicated. It involves repeatedly adding in the initial point. The Mandelbrot set is the region in the complex plane consisting of the values z_0 for which the trajectories defined by

$$z_{k+1} = z_k^2 + z_0, \quad k = 0, 1, \dots$$

remain bounded as $k \rightarrow \infty$. That's it. That's the entire definition. It's amazing that such a simple definition can produce such fascinating complexity.

Copyright © 2008 Cleve Moler
MATLAB[®] is a registered trademark of The MathWorks, Inc.[™]
April 6, 2008

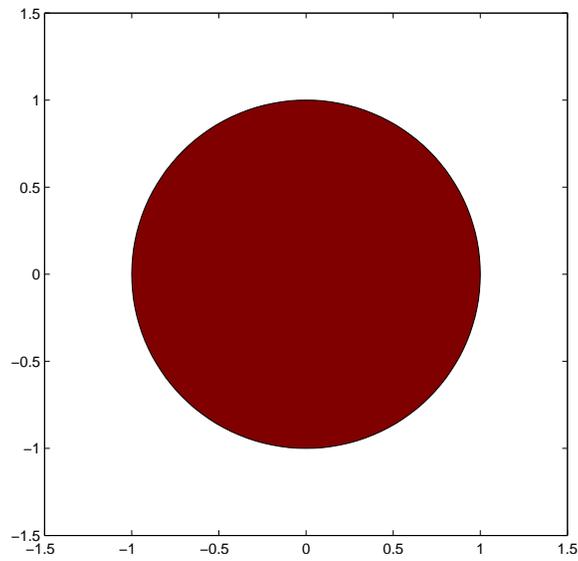


Figure 10.1. *The unit disc is shown in red. The boundary is simply the unit circle. There is no intricate fringe.*

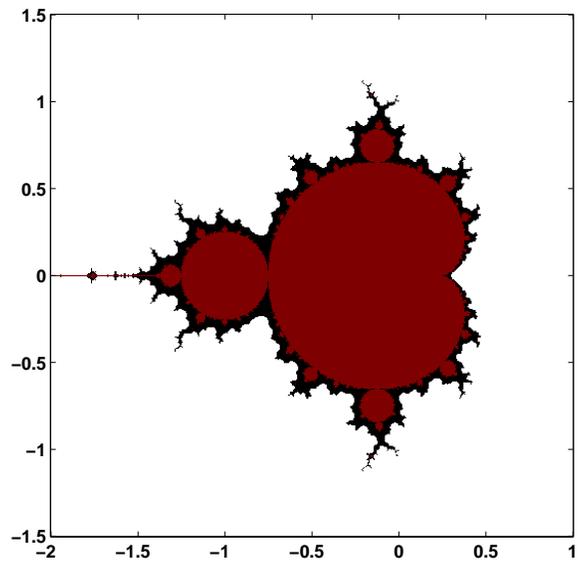


Figure 10.2. *The Mandelbrot set is shown in red. The boundary is very complicated. The fringe just outside the set is a region of rich structure.*

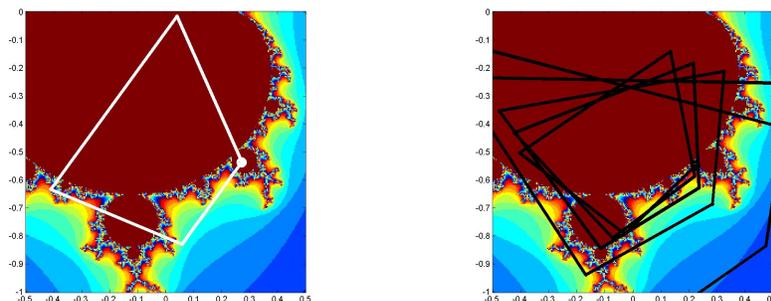


Figure 10.3. *Two trajectories. $z_0 = .25-.54i$ generates a cycle of length four, while nearby $z_0 = .22-.54i$ generates an unbounded trajectory.*

Figure 10.2 shows the overall geometry of the Mandelbrot set. However, this view does not have the resolution to show the richly detailed structure of the fringe just outside the boundary of the set. In fact, the set has tiny filaments reaching into the fringe region, even though the fringe appears to be solid black in the figure. It has recently been proved that the Mandelbrot set is mathematically connected, but the connected region is sometimes so thin that we cannot resolve it on a graphics screen or even compute it in a reasonable amount of time.

To see how the definition works, enter

```
z0 = .25-.54i
z = 0
```

into MATLAB. Then use the up-arrow key to repeatedly execute the statement

```
z = z^2 + z0
```

The first few lines of output are

```
0.2500 - 0.5400i
0.0209 - 0.8100i
-0.4057 - 0.5739i
0.0852 - 0.0744i
0.2517 - 0.5527i
...
```

The values eventually settle into a cycle

```
0.2627 - 0.5508i
0.0156 - 0.8294i
```

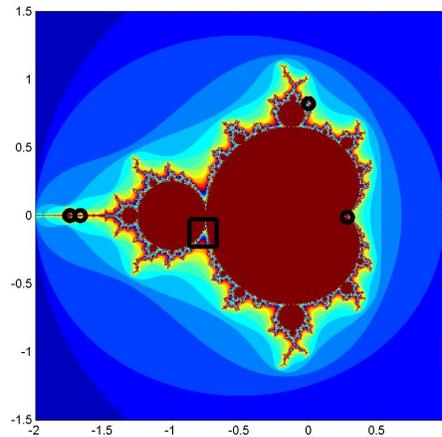


Figure 10.4. *Later figures explore several regions in the fringe of the Mandelbrot set. The square covers “The Valley of the Seahorses.”*

```
-0.4377 - 0.5659i
0.1213 - 0.0446i
0.2627 - 0.5508i
...
```

This cycle repeats forever. The trajectory remains bounded. This tells us that the starting value value, $z_0 = .25 - .54i$, is in the Mandelbrot set. The same cycle is shown in the left half of figure 10.3.

On the other hand, start with

```
z0 = .22 - .54i
z = 0
```

and repeatedly execute the statement

```
z = z^2 + z0
```

You will see

```
0.2200 - 0.5400i
-0.0232 - 0.7776i
-0.3841 - 0.5039i
```

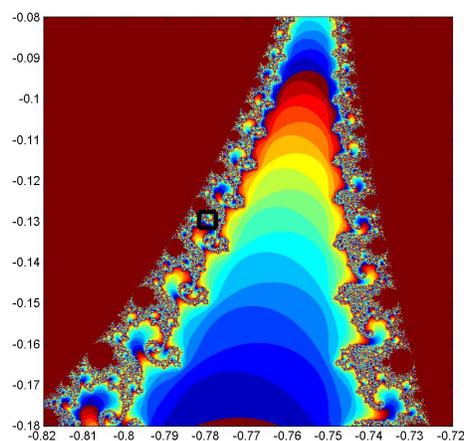


Figure 10.5. *Seahorses #1. Zoom in on the square to explore the “Valley of the Seahorses” in more detail.*

```
0.1136 - 0.1529i
0.2095 - 0.5747i
...
```

Then, after 24 iterations,

```
...
1.5708 - 1.1300i
1.4107 - 4.0899i
-14.5174 -12.0794i
6.5064e+001 +3.5018e+002i
-1.1840e+005 +4.5568e+004i
```

The trajectory is blowing up rapidly. After a few more iterations, the floating point numbers overflow. So this z_0 is not in the Mandelbrot set. The same unbounded trajectory is shown in the right half of figure 10.3. We see that the first value, $z_0 = .25 - .54i$, is in the Mandelbrot set, while the second value, $z_0 = .22 - .54i$, which is nearby, is not.

The algorithm doesn't have to wait until z reaches floating point overflow. As soon as z satisfies

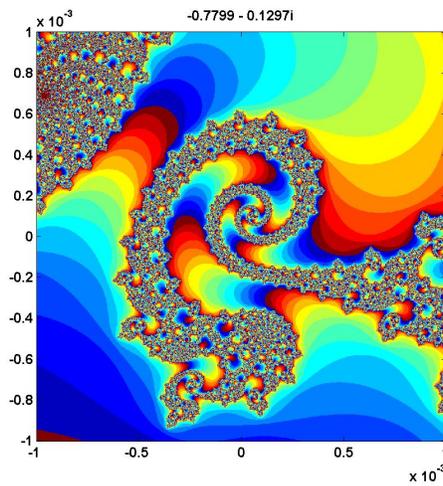


Figure 10.6. *Seahorses #2. More detail in the “Valley of the Seahorses”.*

```
abs(z) >= 2
```

subsequent iterations will essentially square the value of $|z|$ and it will behave like 2^{2^k} .

Try it yourself. Put these statements on one line.

```
z0 = ...
z = 0;
while abs(z) < 2
    z = z^2+z0;
    disp(z),
end
```

Use the up arrow and backspace keys to retrieve the statement and change `z0` to different values near `.25-.54i`. If you have to hit `<ctrl>-c` to break out of an infinite loop, then `z0` is in the Mandelbrot set. If the `while` condition is eventually `false` and the loop terminates without your help, then `z0` is not in the set.

The number of iterations required for z to escape the disc of radius 2 provides the basis for showing the detail in the fringe. Let's add an iteration counter to the loop and make the code into a function, `M(z0)`. A quantity we call `depth` specifies the maximum iteration count and thereby determines both the level of detail and

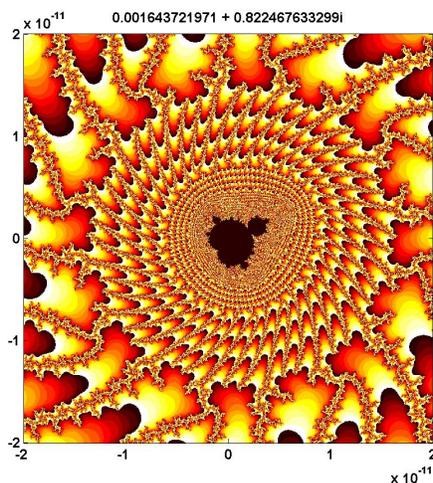


Figure 10.7. The “Buzzsaw” requires a magnification factor of 10^{11} to reveal a miniature copy of the Mandelbrot set.

the overall computation time. Typical values of `depth` are several hundred or a few thousand.

```
function k = M(z0)
z = 0;
k = 0;
while abs(z) < 2 && k < depth
    z = z^2+z0;
    k = k + 1;
end
```

The maximum value of $M(z_0)$ is `depth`. If the value of $M(z_0)$ is less than `depth`, then z_0 is outside the set. Large values of $M(z_0)$ indicate that z_0 is in the fringe, close to the boundary. If $M(z_0)$ reaches `depth` then z_0 is declared to be in the Mandelbrot set.

MATLAB array operations allow us to run this loop on a large grid of complex numbers simultaneously and accumulate an array of iteration counters. Most of the time is spent squaring complex numbers and then counting the number whose absolute value exceeds two.

Here is a small table of $M(z_0)$ as z_0 ranges over complex values near $0.22-0.54i$.

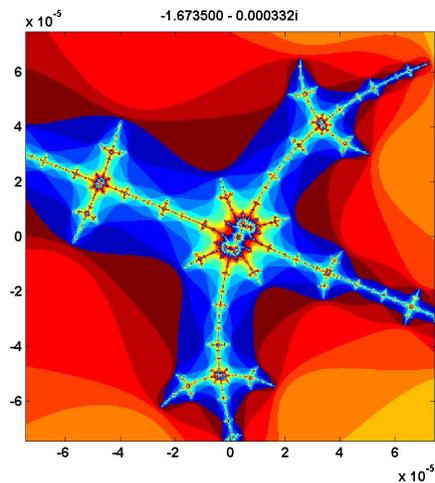


Figure 10.8. Our “West Wing” is located just off the real axis in the thin far western portion of the set, near $\text{real}(z) = -1.67$.

We have set `depth` = 512.

	0.205	0.210	0.215	0.220	0.225	0.230	0.235	0.240	0.245
-0.520	512	512	512	512	512	512	44	512	512
-0.525	512	512	512	512	512	36	51	512	512
-0.530	512	512	512	512	35	31	74	512	512
-0.535	512	512	512	512	26	28	57	512	512
-0.540	512	139	113	26	24	73	56	512	512
-0.545	512	199	211	21	22	25	120	512	512
-0.550	33	25	21	20	20	25	63	85	512
-0.555	34	20	18	18	19	21	33	512	512
-0.560	62	19	17	17	18	33	162	40	344

We see that about half of the values are less than `depth`; they correspond to points outside of the Mandelbrot set, in the fringe near the boundary. The other half of the values are equal to `depth`, corresponding to points that are regarded as in the set. If we were to redo the computation with a larger value of `depth`, the entries that are less than 512 in this table would not change, but some of the entries that are now capped at 512 might increase.

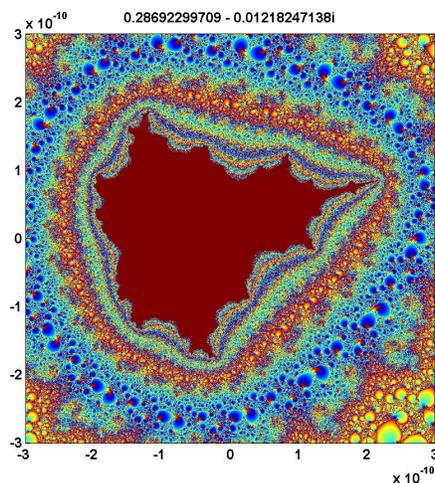


Figure 10.9. *Geode.* This colorful image requires a 2048-by-2048 grid and $depth = 4096$.

The function values $M(z_0)$ can be used as indices into an RGB color map of size `depth`-by-3. The first row of this map specifies the color assigned to any points on the z_0 grid that lie outside the disc of radius 2. The next few rows provide colors for the points on the z_0 grid that generate trajectories that escape quickly. The last row of the map is the color of the points that survive `depth` iterations and so are in the set. The map used in figure 10.2 emphasizes the set itself and its boundary. The map has 12 rows of white at the beginning, one row of dark red at the end, and black in between. Images that emphasize the structure in the fringe are achieved when the color map varies cyclicly over a few dozen colors. One of the exercises asks you to experiment with color maps.

The `exm` toolbox function `mandelbrot` is your starting point for exploration of the Mandelbrot set. With no arguments, the statement

```
mandelbrot
```

starts with the full set. The statement

```
mandelbrot(center,width,grid,depth,cmapindx)
```

explores the Mandelbrot set in a square region of the complex plane with the specified `center` and `width`, using a `grid`-by-`grid` grid, an iteration limit of `depth`, and the color map with index `cmapindx`. The default values of the parameters are

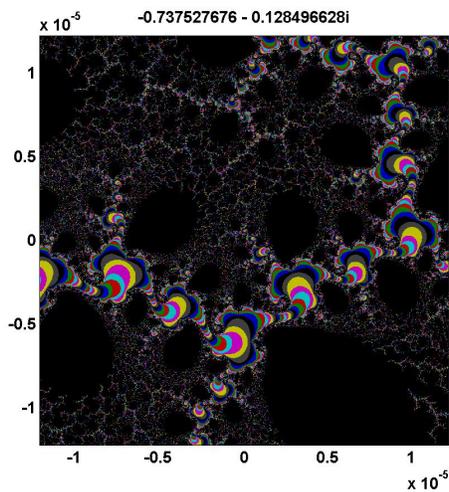


Figure 10.10. *Nebula. Interstellar dust.*

```
center = -0.5+0i
width = 3
grid = 512
depth = 256
cmapindx = 1
```

In other words,

```
mandelbrot(-0.5+0i, 3, 512, 256, 1)
```

This generates figure 10.4, without the outlines of the subregions. On my laptop, the computation takes about one second.

A simple estimate of the execution time is proportional to

```
grid^2 * depth
```

So the statement

```
mandelbrot(-0.5+0i, 3, 2048, 1024, 1)
```

could take

$$(2048/512)^2 \cdot (1024/256) = 64$$

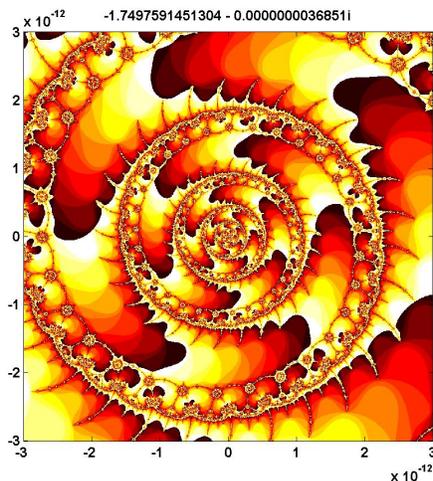


Figure 10.11. A vortex, not far from the West Wing. Zoom in on one of the circular “microbugs” near the left edge.

times as long as the default. However, this is an overestimate and the actual execution time is about half of that.

Most of the computational time required to compute the Mandelbrot set is spent updating two arrays `z` and `kz` by repeatedly executing the step

```
z = z.^2 + z0;
j = (abs(z) < 2);
kz(j) = d;
```

This computation can be carried out faster by writing a function `mandelbrot_step` in C and creating as a MATLAB *executable* or *c-mex* file. Different machines and operating systems require different versions of a mex file, so you should see files with names like `mandelbrot_step.mexw32` and `mandelbrot_step.glnx64` in the `exm` toolbox.

The `mandelbrot` gui turns on the MATLAB zoom feature. The mouse pointer becomes a small magnifying glass. You can click and release on a point to zoom by a factor of two, or you can click and drag to delineate a new region.

The `mandelbrot` gui provides several uicontrols. Try these as you read along. A `listbox` at the bottom of the gui allows you to select any of the predefined regions shown in the figures in this chapter.

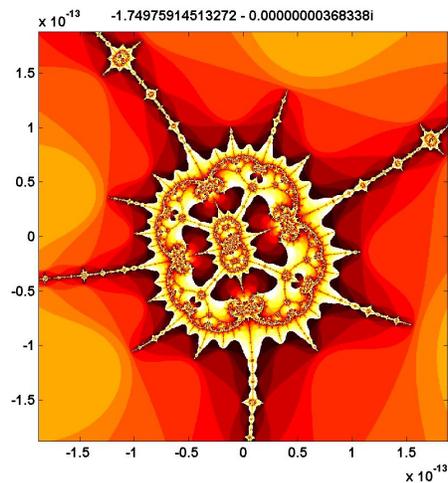


Figure 10.12. A 10^{13} magnification factor reveals a “microbug” within the vortex.

depth. Increase the depth by a factor of $3/2$ or $4/3$, and

grid. Refine the grid by a factor of $3/2$ or $4/3$. The depth and grid size are always a power of two or three times a power of two. Two clicks on the **depth** or **grid** button doubles the parameter.

color. Cycle through several color maps. **jets** and **hots** are cyclic repetitions of short copies of the basic MATLAB **jet** and **hot** color maps. **cmk** cycles through eight basic colors, blue, green, red, cyan, magenta, yellow, gray, and black. **fringe** is a noncyclic map used for images like figure 10.2.

exit. Close the gui.

The portion of the boundary of the Mandelbrot set between the two large, nearly circular central regions is known as “The Valley of the Seahorses”. Figures 10.5 and 10.6 show the results of zooming in on a portion of the valley. Figure 10.5 is obtained by zooming in on the square in figure 10.4. Figure 10.6 is obtained by zooming in on the square in figure 10.6. The two figures can be generated directly with the commands

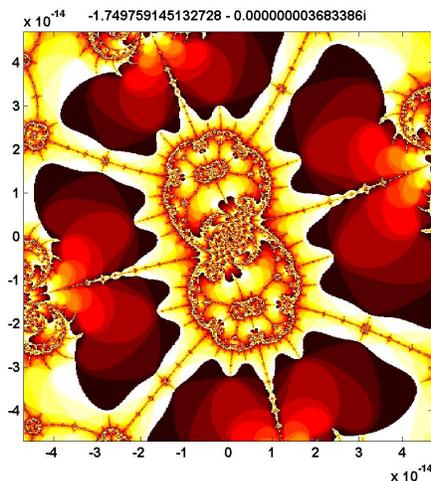


Figure 10.13. *The nucleus of the microbug.*

```
mandelbrot(-.7700-.1300i,0.1,1024,512)
mandelbrot(-.7799-.1297i,0.002,1024,512)
```

We see that a magnification factor of about 10^3 is required for the detailed image in figure 10.6

The Mandelbrot set is *self similar*. Small regions in the fringe reveal features that are similar to the original set. One of the best known examples, shown in figure 10.7, is called the “Buzzsaw”. It can be generated with

```
mandelbrot(0.001643721971153+0.822467633298876i, ...
4.0e-11,1024,2048,2)
```

Taking `width = 4.0e-11` corresponds to a magnification factor of almost 10^{11} . To appreciate the size of this factor, if the original Mandelbrot set fills the screen of your computer, the Buzzsaw is smaller than the individual transistors in your machine’s microprocessor.

We decided to name the image in figure 10.8 the “West Wing” because it resembles the X-wing fighter that Luke Skywalker flies in Star Wars and because it is located near the leftmost, or far western, portion of the set. The magnification factor is a relatively modest 10^4 , so `depth` does not need to be very large. The command to generate the West Wing is

```
mandelbrot(-1.6735-0.0003318i,1.5e-4,1024,160,1)
```

The most intricate and colorful image among our examples is figure 10.9, the “Geode”. It involves a fine grid and a large value of `depth` and consequently requires a few minutes to compute.

```
mandelbrot(0.28692299709-0.01218247138i,6.0e-10,2048,4096,1)
```

We call figure 10.10 “Nebula” because it reminds us of interstellar dust. It is generated by

```
mandelbrot(0.73752777-0.12849548i,4.88e-5,1024,2048,3)
```

Our last three images are obtained by carefully zooming on one location. We call them the “Vortex”, the “Microbug”, and the “Nucleus”.

```
mandelbrot(-1.74975914513036646-0.00000000368513796i, ...
6.0e-12,1024,2048,2)
```

```
mandelbrot(-1.74975914513271613-0.00000000368338015i, ...
3.75e-13,1024,2048,2)
```

```
mandelbrot(-1.74975914513272790-0.00000000368338638i, ...
9.375e-14,1024,2048,2)
```

These examples are just a tiny sampling of the structure of the Mandelbrot set.

Further Reading

We highly recommend a real time fractal zoomer called “XaoS”, developed by Thomas Marsh, Jan Hubicka and Zoltan Kovacs, assisted by an international group of volunteers. See

<http://wmi.math.u-szeged.hu/xaos/doku.php>

If you are expert at using your Web browser and possibly downloading an obscure video codec, take a look at the Wikipedia video

<http://commons.wikimedia.org/wiki/> ...

`Image:Fractal-zoom-1-03-Mandelbrot_Buzzsaw.ogg`

It’s terrific to watch, but it may be a lot of trouble to get working.

Exercises

10.1 *Explore*. Use the `mandelbrot` gui to find some interesting regions that, as far as you know, have never been seen before. Give them your own names.

10.2 *M(z0)*. Reproduce our table of $M(z_0)$ for z_0 for $z_0 = x+i*y$ where $x = .205:.005:.245$ and $y = -.520:-.005:-.560$. First, use `depth = 512`. Then use larger values of `depth` and see which table entries change.

10.3 *p-th power*. Change the iteration to

$$z_{k+1} = z_k^p + z_0$$

for some fixed $p \neq 2$. Better yet, add a button that lets you set p .

10.4 *Color maps*. Add more color maps.

10.5 When the width of the region gets to be smaller than about 10^{-15} , our `mandelbrot` gui does not work very well. Why?

10.6 *Spin the color map*. When you have an interesting region plotted in the figure window, bring up the command window, resize it so that it does not cover the figure window, and enter the command

```
spinmap(10)
```

I won't try to describe what happens – you have to see it for yourself. The effect is most dramatic with the “seahorses2” region. Enter

```
help spinmap
```

for more details.