

Orbit Diagram for $Q_c = x^2 + c$

We are going to plot the orbit diagram for Q_c using a simple BASIC program. We will have 360 different c values (seed values) equally spaced between -2 and .25. We'll do 200 iterations for each c value, but in order to see the ultimate behavior of each plot we will leave out the first 50 plots of each c value. This simple little function produces a complex and beautiful orbit diagram!

Note: The following BASIC source code listing comes from Robert Devaney's book "Chaos, Fractals, and Dynamics." Everyone should have a copy of this amazing book!

```
REM program ORBITDGM
CLS
SCREEN 9
COLOR 1, 7
FOR c = -2 TO .25 STEP .00625
  x = 0
  m = 160 * (c + 2)
  FOR i = 0 TO 200
    x = x * x + c
    IF i < 50 GOTO 11
    n = 75 * (2 - x)
    PSET (m, n)
11 NEXT i
NEXT c
END
```

If you are a BASIC programmer this listing will seem pretty simple. If not, here are a few words of explanation.

REM

REM means remark. This is a note to yourself or someone else who may read the listing. It tells a little bit about the program. The BASIC compiler will disregard any line that starts with "REM."

It's good to make lots of remarks. If you come back to the source code years later they will help to refresh your memory about, "What the heck was I thinking?"

Also, it's a good technique to remark out some parts of the program to see what effect leaving out those commands has. Then, if you decide to add them back in, you don't have to re-enter the code. You just remove the "REM."

CLS = Clear Screen

SCREEN 9 = Make the graphics screen 640 x 350 pixels

COLOR a,b = Set the foreground color to "a" and the background color to "b"

FOR c = -2 TO .25 STEP .00625

Here's where we get 360 different c values. Check it out on your calculator $2.25/360 = 0.00625$

x=0

We are going to start with an x value of 0 and increment from there

Now, when we plot points on the screen we say PSET (m,n) where m = the horizontal or "x" value and n = the vertical or "y" value.

Hopefully that's not too confusing. We are so used to thinking in terms of x/y coordinates!

In this listing we are going to PSET (m,n). So that's like saying, "put a dot at the coordinates (m,n).

```
m=160*(c+2)
```

This will set the starting value of m to zero. (Remember we have already said the first value of c will be -2. So, we are scaling the x axis to the screen). I mean, we want the graph to start on the left hand side of the screen.

```
For i=0 TO 200
```

Starts us out with an "i" value of zero, and we will increment it up to 200 (that's the number of iterations we will compute for each seed value, but as seen later, we'll through the first 50 out).

```
x=x*x+c
```

This is the quadratic equation $Q_c = x^2 + c$. So here's where we compute the value of the function for this given value of c. Are we getting close or what!

```
IF i<50 GOTO 11
```

We skip the first 50 iterations and don't plot them.

```
n=75*(2-x)
```

This is the 'vertical' or "y" value of the plot.

```
PSET (m,n)
```

OK, this is the big one! This command says, "Go ahead and put a pixel at (m,n). So, finally we have written something to the screen!

```
11 NEXT i
```

Increment i

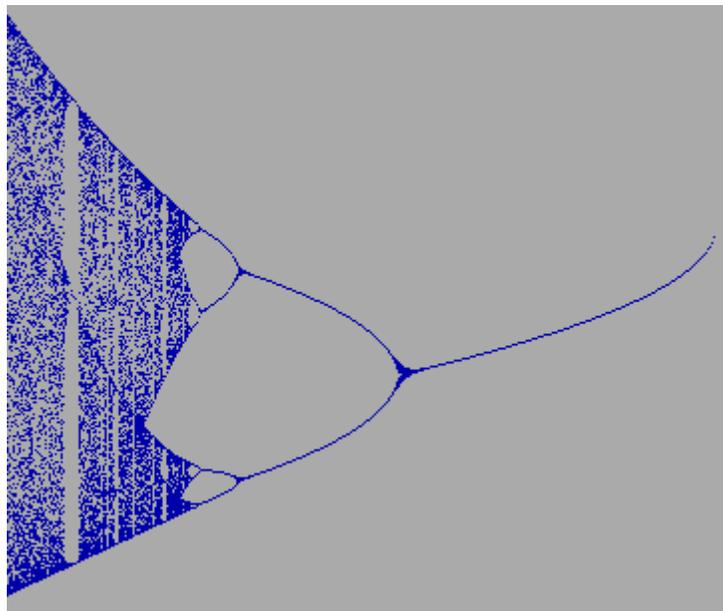
NEXT c

Now we start over with the next "seed value." This is the next of the 360 values of c that we will plot the orbit diagram for.

END

Once we have plotted orbit diagrams for each of the values of c we are done!

Here's a screenshot of the orbit diagram.



Orbit Diagram for $Q_c = x^2 + c$ from -2 to .25

Some interesting things to note are:

1. The orbit for each c value appears to be completely unpredictable.
2. Orbits for some c values are much more complex than others. This supports the concept of "sensitive dependence on initial conditions."
3. There is a very evident overall pattern.