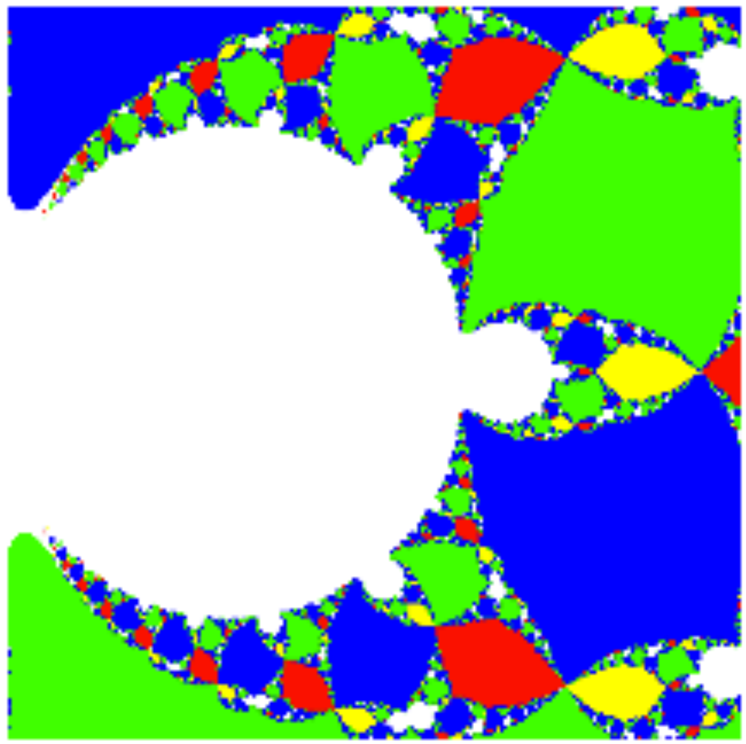
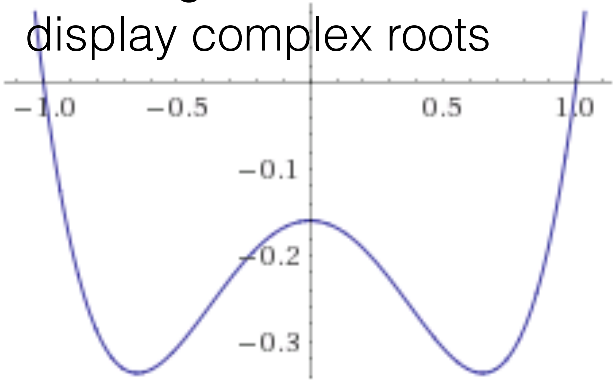
Newton Basins



qdrive Δ qnewt Δ myownpolyder

Note: figure does not display complex roots



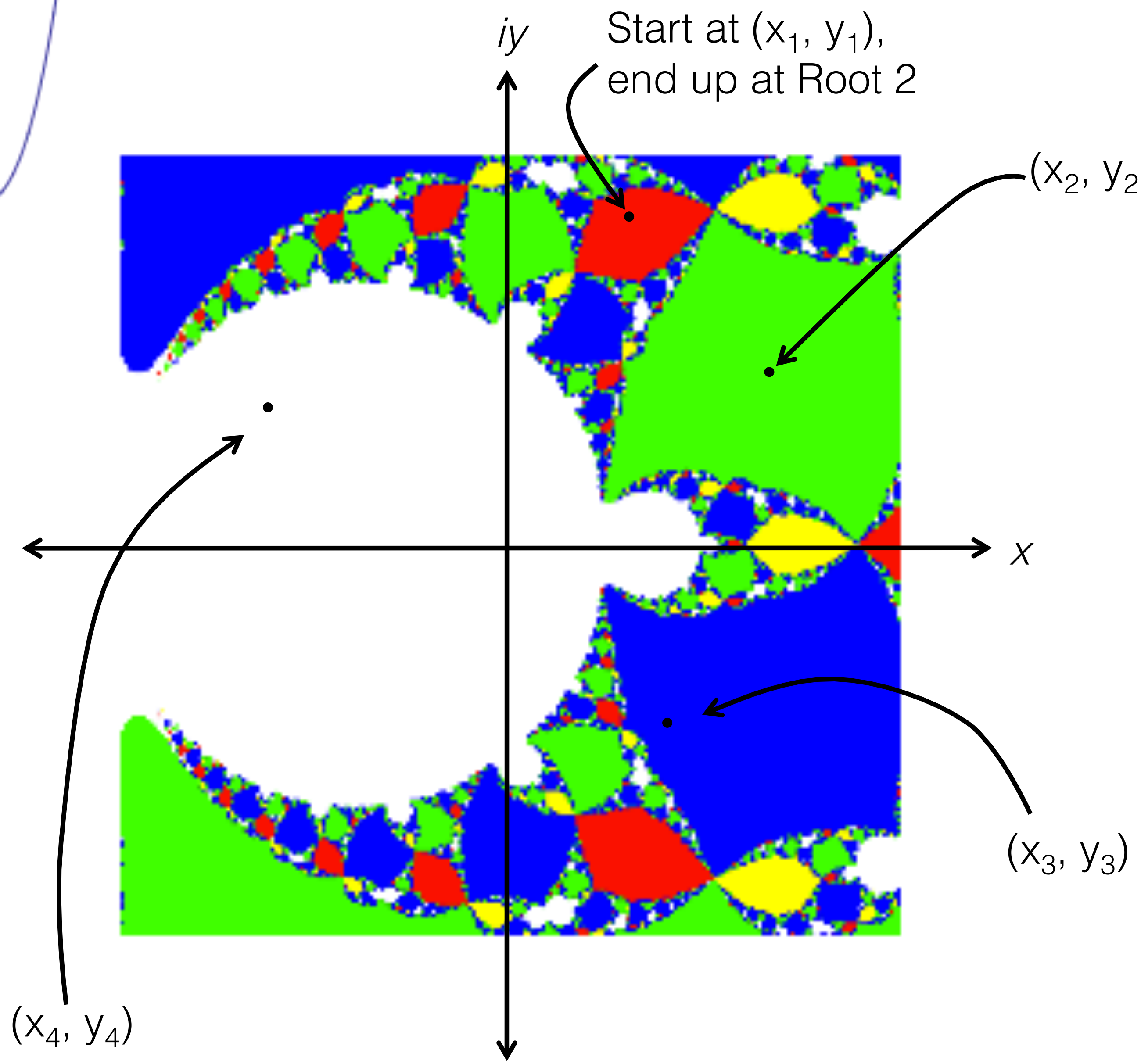
Root 1: Yellow

Root 2: Red

Root 3: green

Root 4: blue

I claim that you only need four color cases. What does it mean when we see white on the plot?

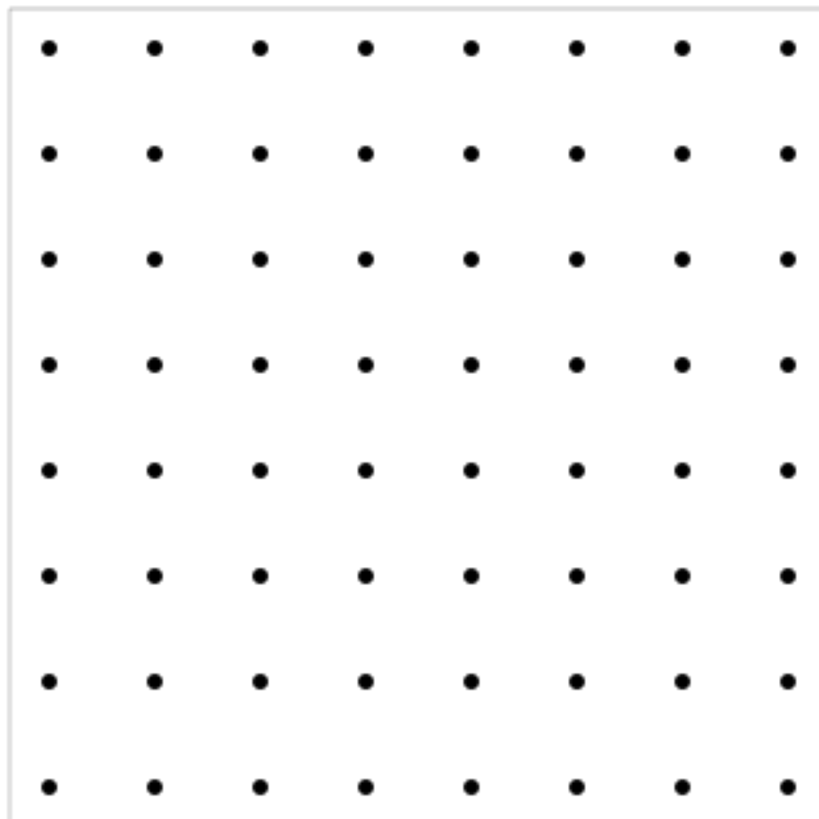


It would be extremely expensive to invoke a for loop and run through every point in the window.

In fact, unless we were working with a finite set of points, the loop would go on forever, and the picture would become more and more detailed

So we make a lattice that covers the plane (window specified by xt and yt) and *perform Newton's method on all of the points at once!*

Then we have a huge collection (matrix) of the roots that Newton's method finds for each starting point in the lattice.



Outline

```
function qdrive
    *run qnewt on all 4 quartics*
return

function qnewt(q,xt,yt,maxiter)
    *We will unpack this one*
return

function d=myownpolyder(q)
    *Write your own derivative-taking function (one-liner)*
return
```

qdrive

```
function qdrive

 qnewt([1 0 -0.84 0 -0.16],[.45 .0001 .55],[-.05 .0001 .05],20)

      [1 0 -0.84 -0.1 -0.16]
      [1 -0.1 -0.84 0 -0.16]
      [1 -0.1i -0.84 0 -0.16]

return
```

all of the plotting is taken care of in qnewt! The driver really just specifies the quartics and the dimensions of the grid.

qnewt(q,xt,yt,maxiter)

Where are these specified?

```
function qnewt(q,xt,yt,maxiter)
```

1. create meshgrid

```
x = .15:.0025:.55;
y = -.15:.0025:.15;
[X,Y] = meshgrid(x,y);
Z = X+i*Y;
```

2. take derivative of your q using
   dq= mypolyder(q)
3. Run Newton on entire lattice at once

```
for k=1:maxiter,
    Z = Z - (Z.^2-1).*(Z.^2+0.16)./(4*Z.^3 - 1.68*Z);
end
```

What are the numerator and denominator in terms of dq?

Why the '.' ?

4. Find roots of q. There is a built-in MATLAB function for this:
   R = roots(q);

We know that there are 4 roots of q. So in what form must R be?

5. Use "find" to see which points in Z end up at which root, and plot in the corresponding colors (see next slide).

```
return
```

plotting in qnewt

R is a 1x4 vector of the roots of q.

We will evaluate each of these roots independently, and see which points in Z ended up at that root.

0.1 is the tolerance you will use

1.
```
[i1,j1] = find(abs(Z-1)<0.1);
```

1 is an example root from the notes. How can we write each of the roots in terms of R for the four cases?

2. Plot: use indices i1 and j1 to access points in the x-y plane; color those points whatever color you like. Use markersize 1.
   - Hint for accessing points in the x-y plane: where have we discretized the plane/created a lattice? (see slide 8, step 1).

3. hold on, and repeat for the next 3 roots.

dq = myownpolyder(q)

$$2z^4 - 4z^3 + (2-i)z^2 - iz + 10$$

q = [2 -4 (2-i) -i 10]

```
dq = [8 -12 4-2i -i]
```

How can we automate the process of arriving from q to dq? What same operations are performed *every time*?

dq = myownpolyder(q)

$$2z^4 - 4z^3 + (2-i)z^2 - iz + 10$$

q = [2 -4 (2-i) -i 10]

x4 x3 x2 x1

```
dq = [8 -12 4-2i -i]
```

DIES

-kill the last element of q

-multiply the first four elements of q by the corresponding exponents [4 3 2 1]