

Matlab

Newton's Method

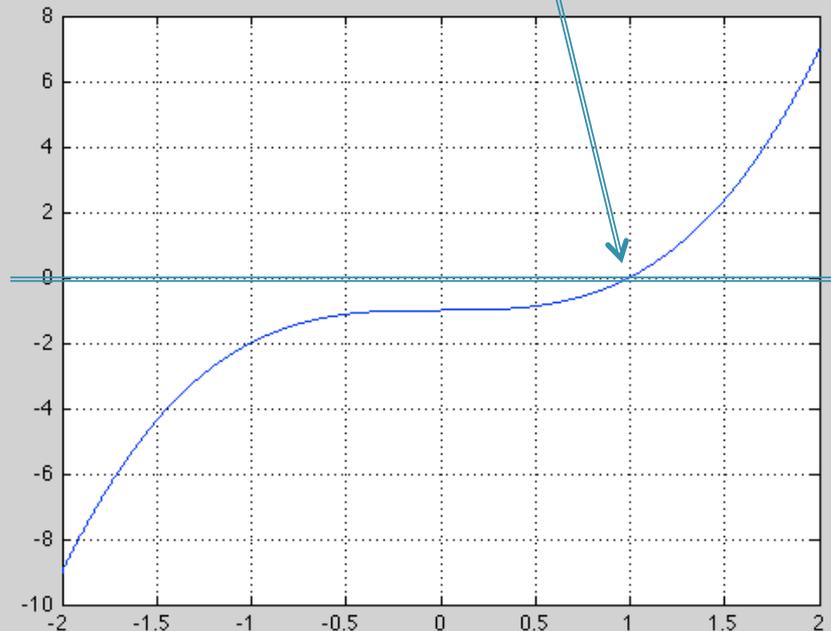
Newton's method is a numerical technique for finding the (approximate) root(s) of the equation

$$f(x) = 0$$

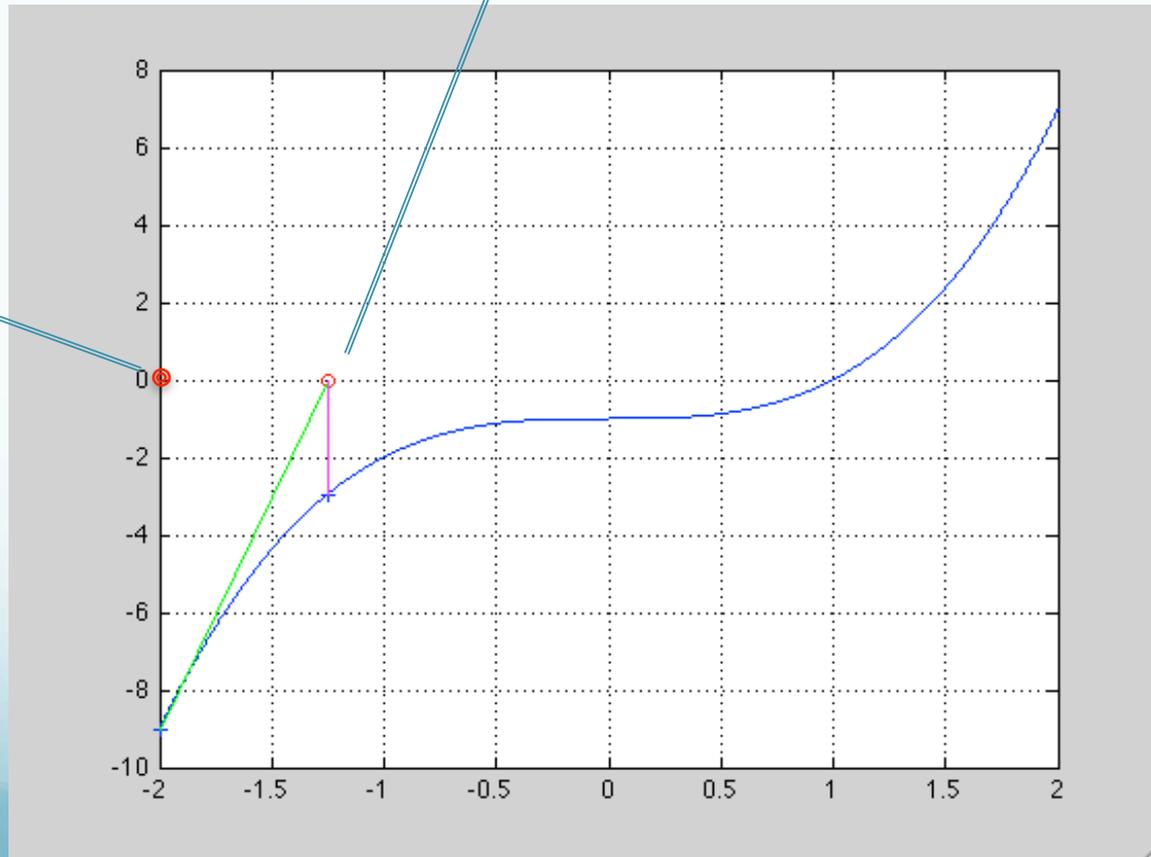
Below we see a graph of the equation

$$f(x) = x^3 - 1 = 0$$

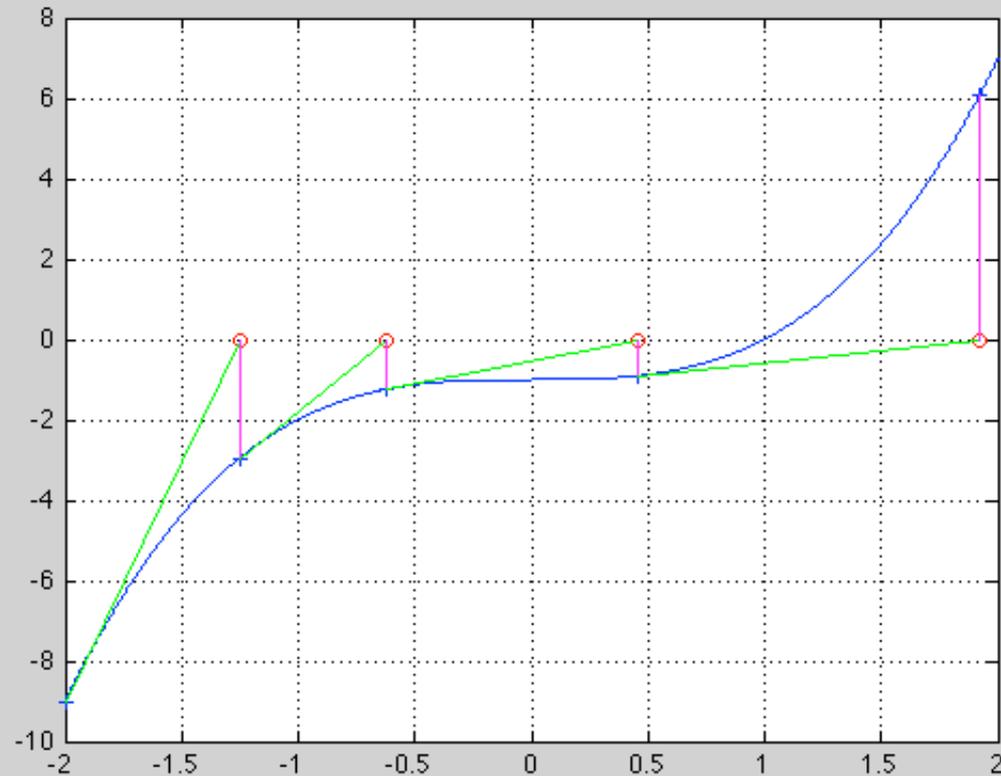
which has a real root at $x = 1$



The idea of Newton's method is to guess the root and then "iterate" – generating new approximations to the root by using the intercept of the tangent line at the current approximation.



This process is repeated until an acceptable approximation to the root is obtained (or not).



If we know the analytic form of the function we can use that to calculate the derivative (tangent).

If we do not know the analytic form of the function, we can calculate the derivative (tangent) from the data ($\Delta y / \Delta x$).

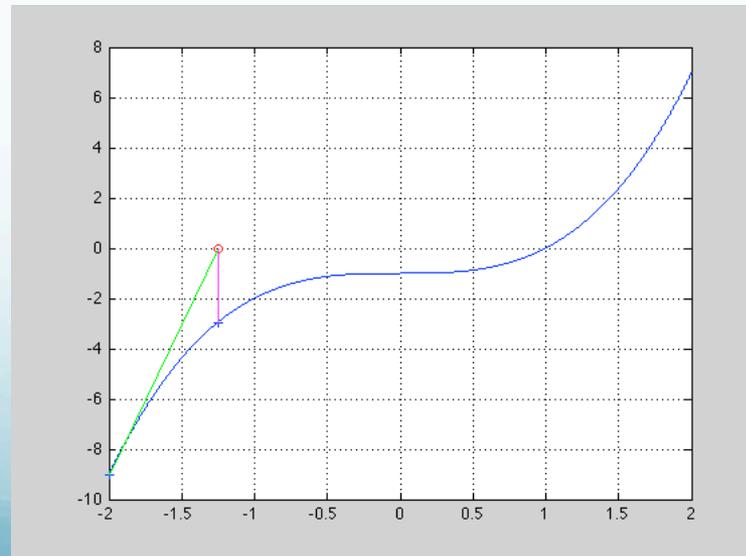
For the next approximation, we have

$$y = mx + b$$

the equation for a line and

$$m = \frac{df(x)}{dx} = 3x^2$$

the analytic equation for the derivative



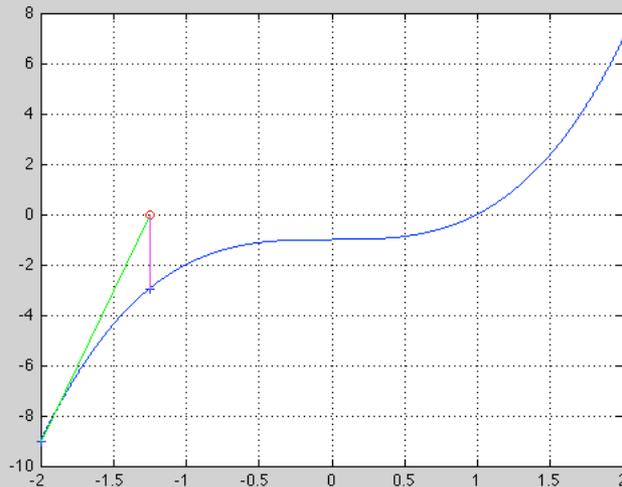
$$f(x_0) = x_0^3 - 1 = y_0 = mx_1$$

$$f(x_1) = x_1^3 - 1 = y_1 = mx_2$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0$$

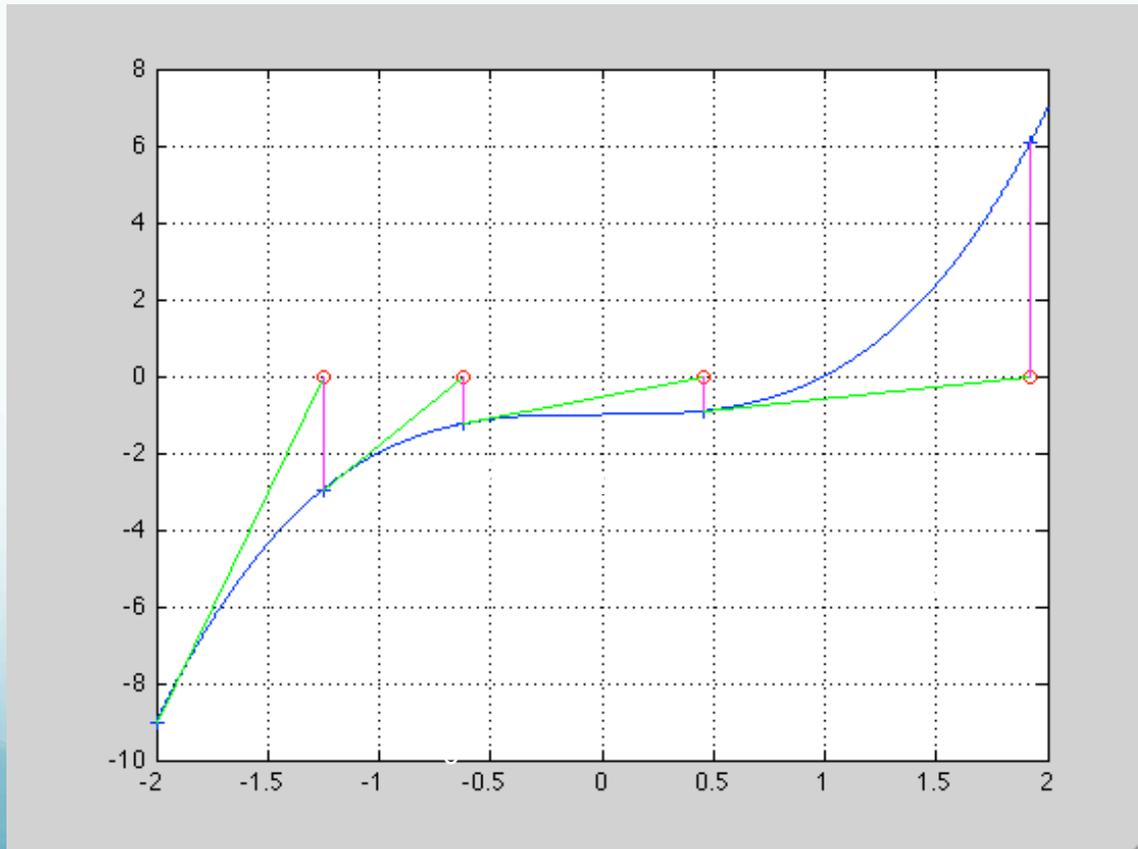
Using this, for

$$x_0 = -2, \quad x_1 \approx 1.25$$

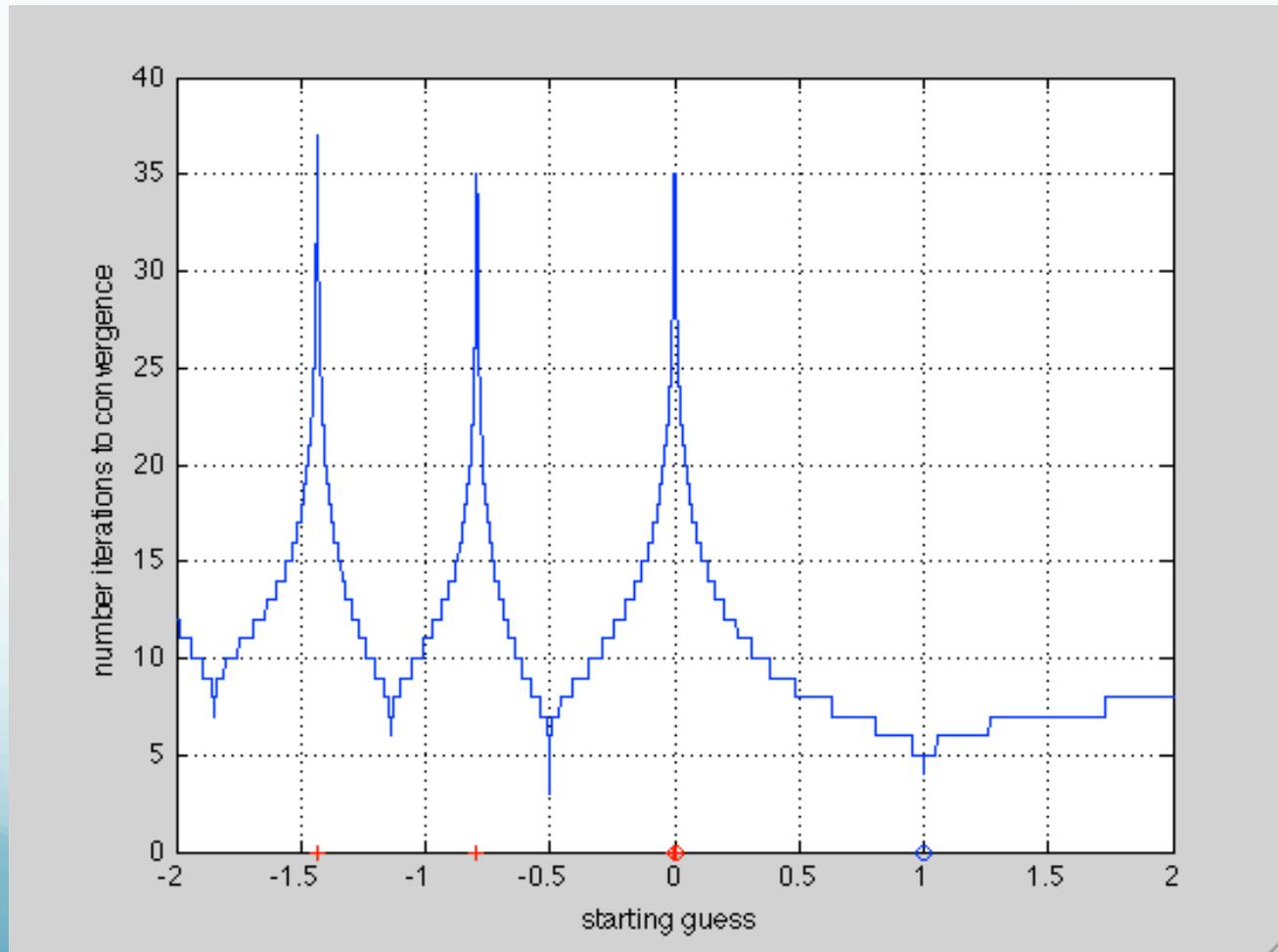


x_n approaches the root $x=0$ for the first few iterations, then diverges slightly. This is due to the small value of the slope at x_3 .

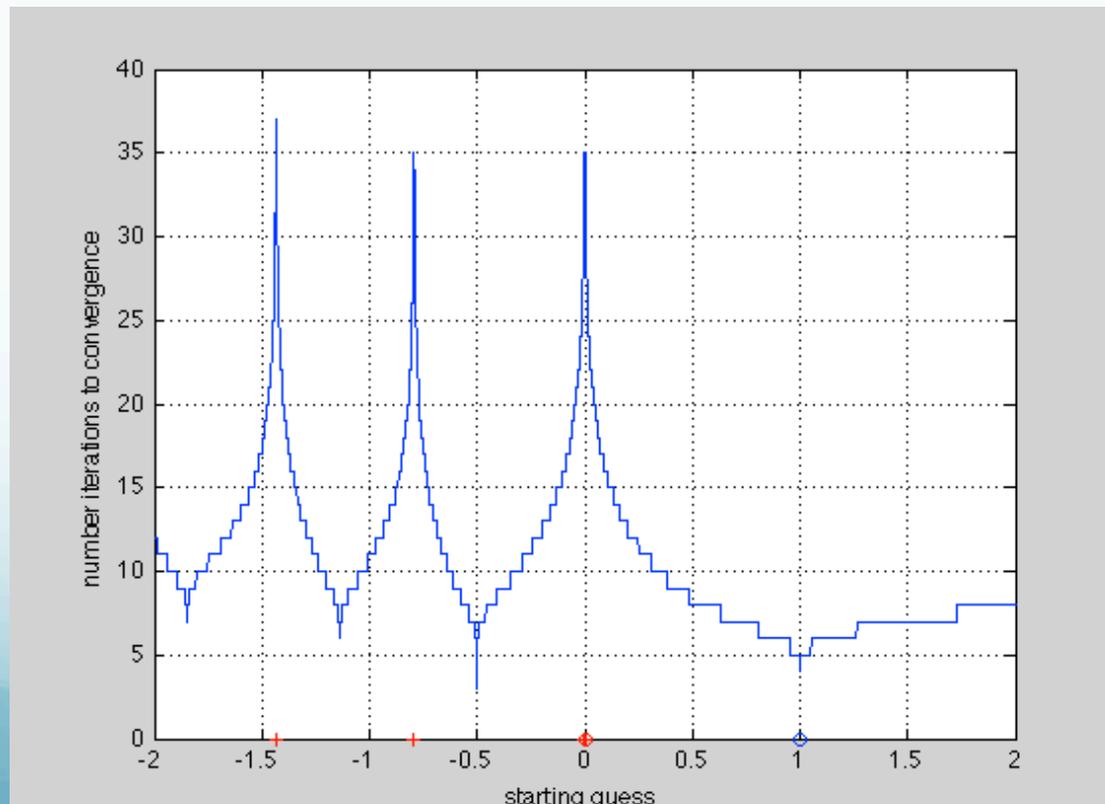
(and this shows the problem when $f'=0$, the new approximation shoots off to infinity. More on this later.)



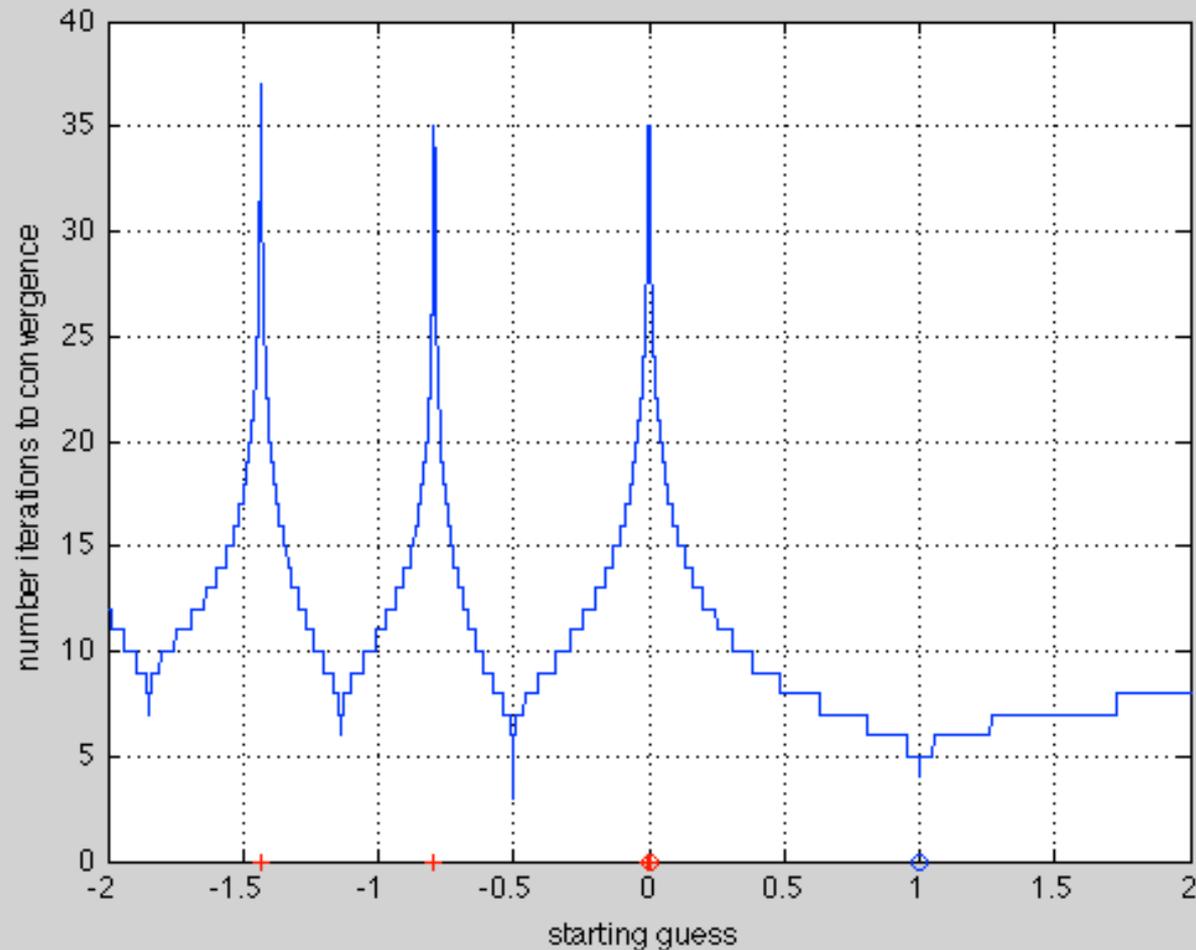
What happens if I chose another starting value?
The plot below shows the number of iterations to arrive at the solution for "all" starting values between -2 and 2.



Notice there are several starting values for which convergence is not reached: the red circle at $x=0$ indicates the method fails there immediately due to the slope being zero, while the red “+” signs indicate convergence has not been reached after 30 iterations.

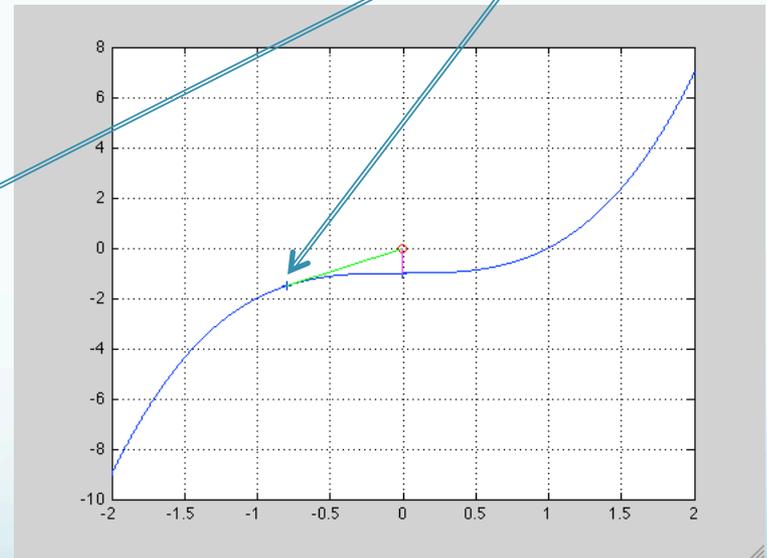
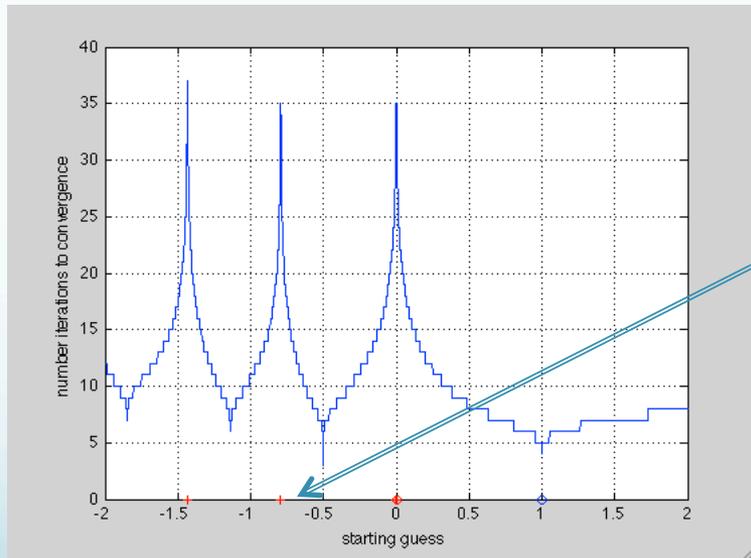


In general, for most starting values there is no simple way to predict the rate (or success) of convergence from the starting value.



We can easily find the first non-zero starting value that diverges – it is the one that produces $x=0$ at the next guess.

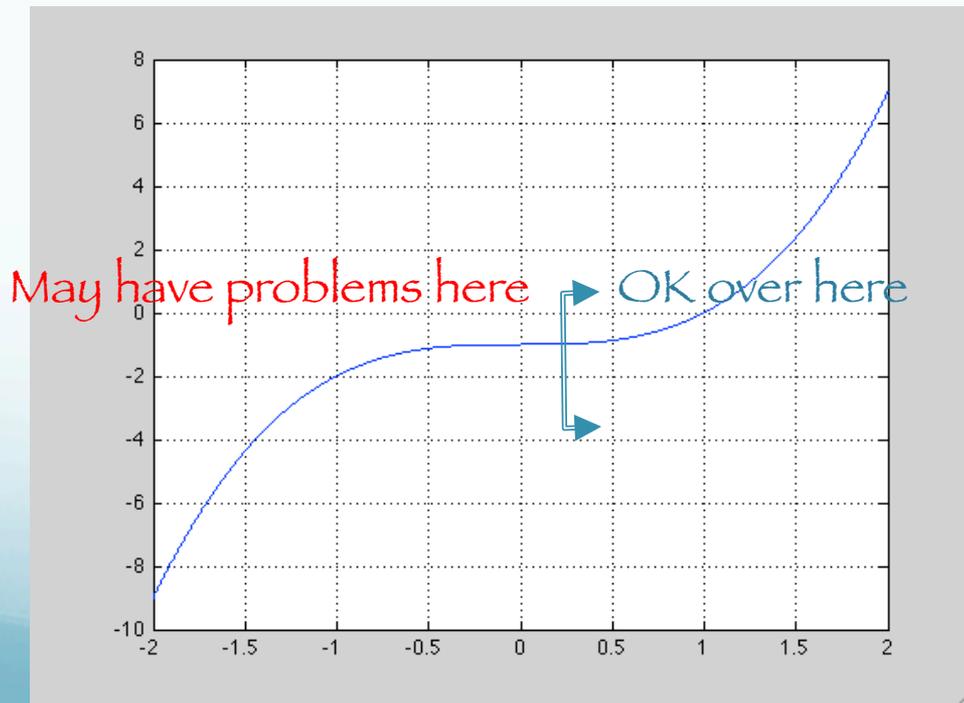
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad x_1 = 0 = x_0 - \frac{x_0^3}{3x_0^2}, \quad x_0 = \sqrt[3]{-0.5} = -0.7937\dots$$



We can continue calculating the “next” bad starting value.

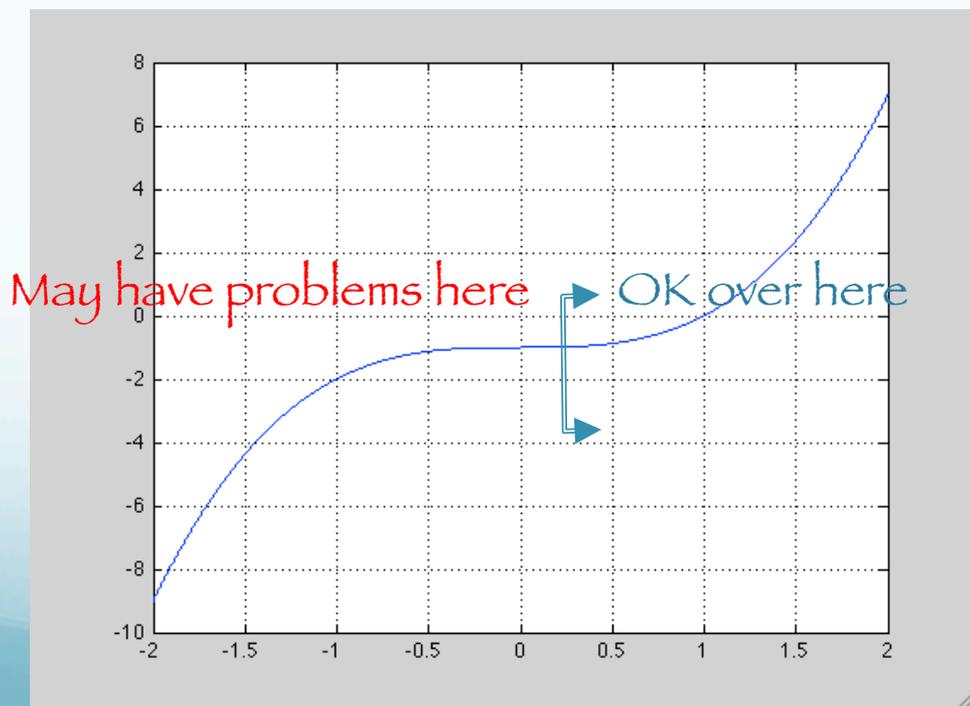
As long as there are no places where the derivative is zero between where you are and the solution, you are OK (you will converge).

If there is a place with zero slope between you and the solution, you will have problems if you get near that spot.



In this case, where we know the function and derivative, you can find the starting values that will have problems by working backwards.

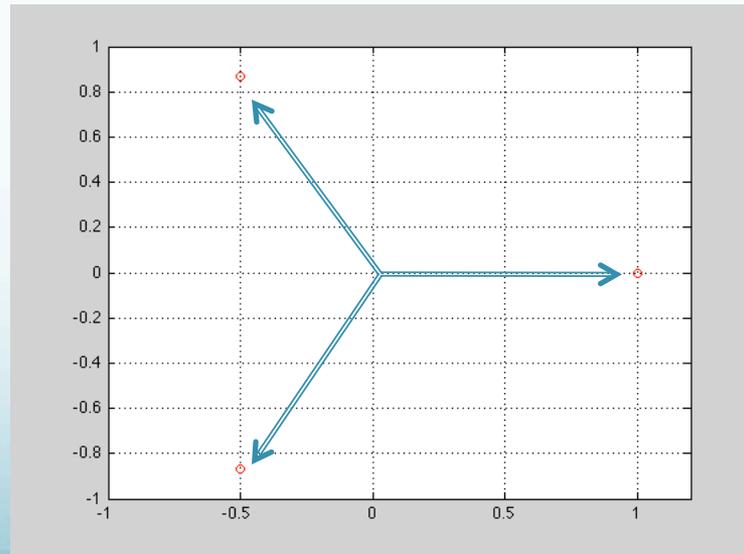
If you don't have $f(x)$ analytically, it is harder to identify the bad starting values.



This is not the full solution however.

So far we have only considered the real part of the problem.

We know that $x^3 - 1 = 0$ has 3 solutions.
(magnitude = 1, at 120° around the origin.)



So now let's try Newton's method again, but now letting x be complex (and calling it z), and taking starting points in the 4×4 box around the origin as starting points.

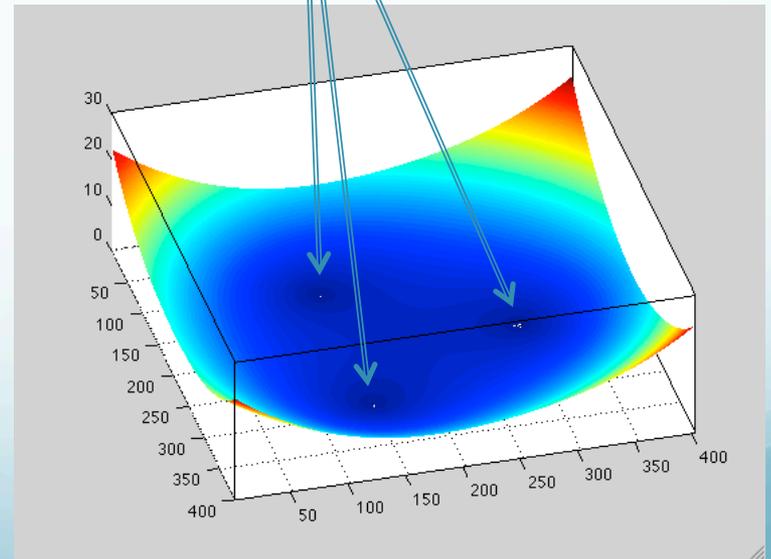
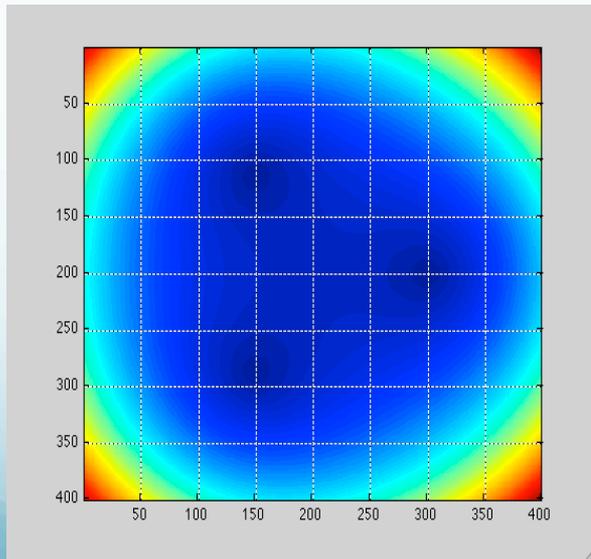
(many presentations of Newton's method say f has to be a real valued function, but this restriction is not needed. Just use the complex derivative, which is like the gradient – it will also have a direction.)

We will count how many iterations to convergence plus keep track of which root Newton's method converges to. We will indicate the root by color (so we will have 3 colors), and use brightness to indicate the speed of convergence (brighter is faster).

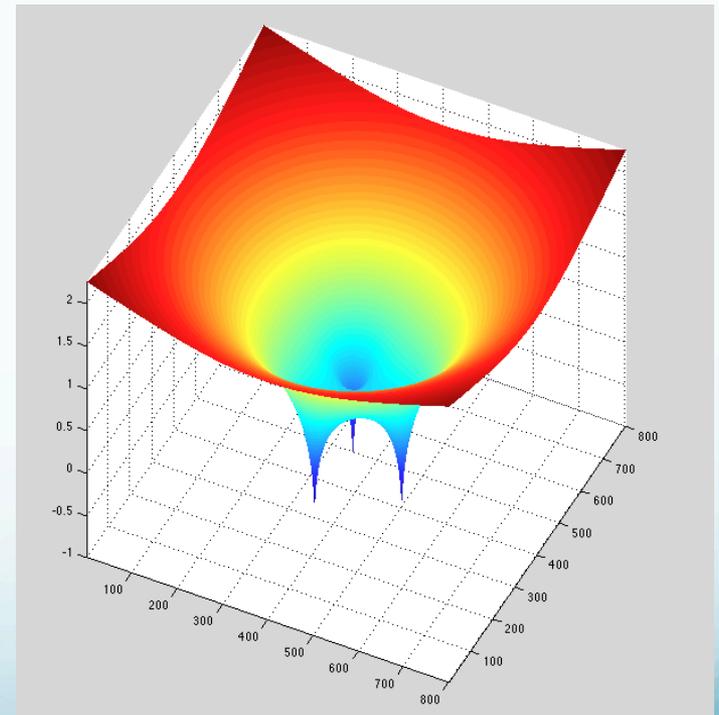
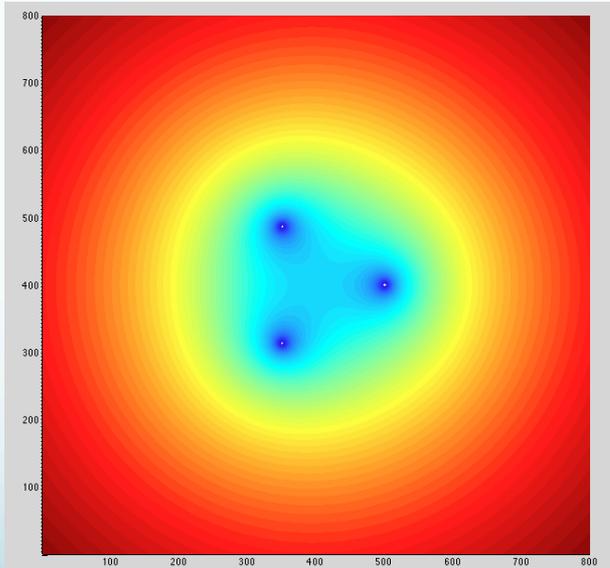
Here are some plots of the absolute value of $f(z)$

($f(z)$ is a complex number so we can't plot it directly).

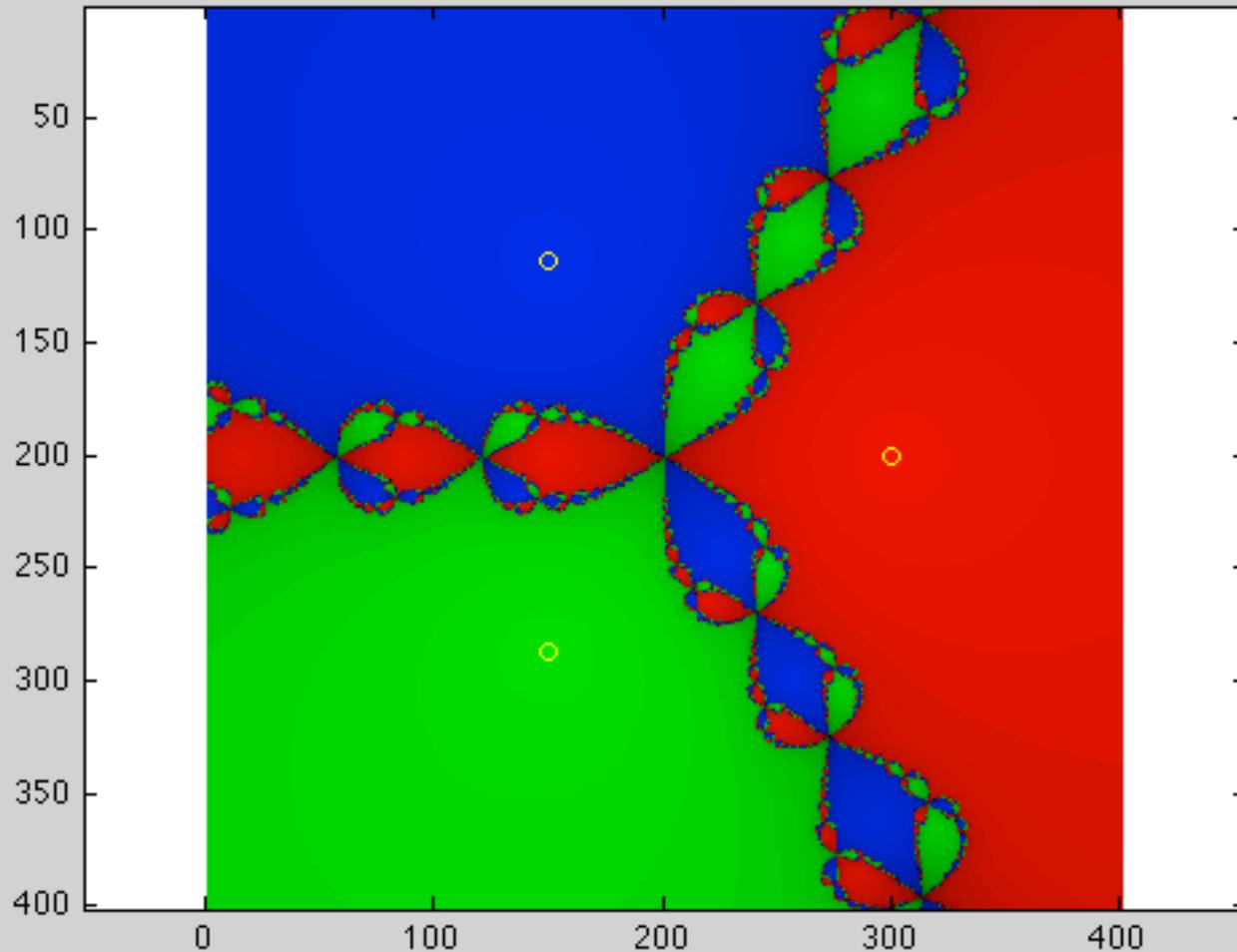
We are aiming for the 3 low spots.



Plot log to see better.

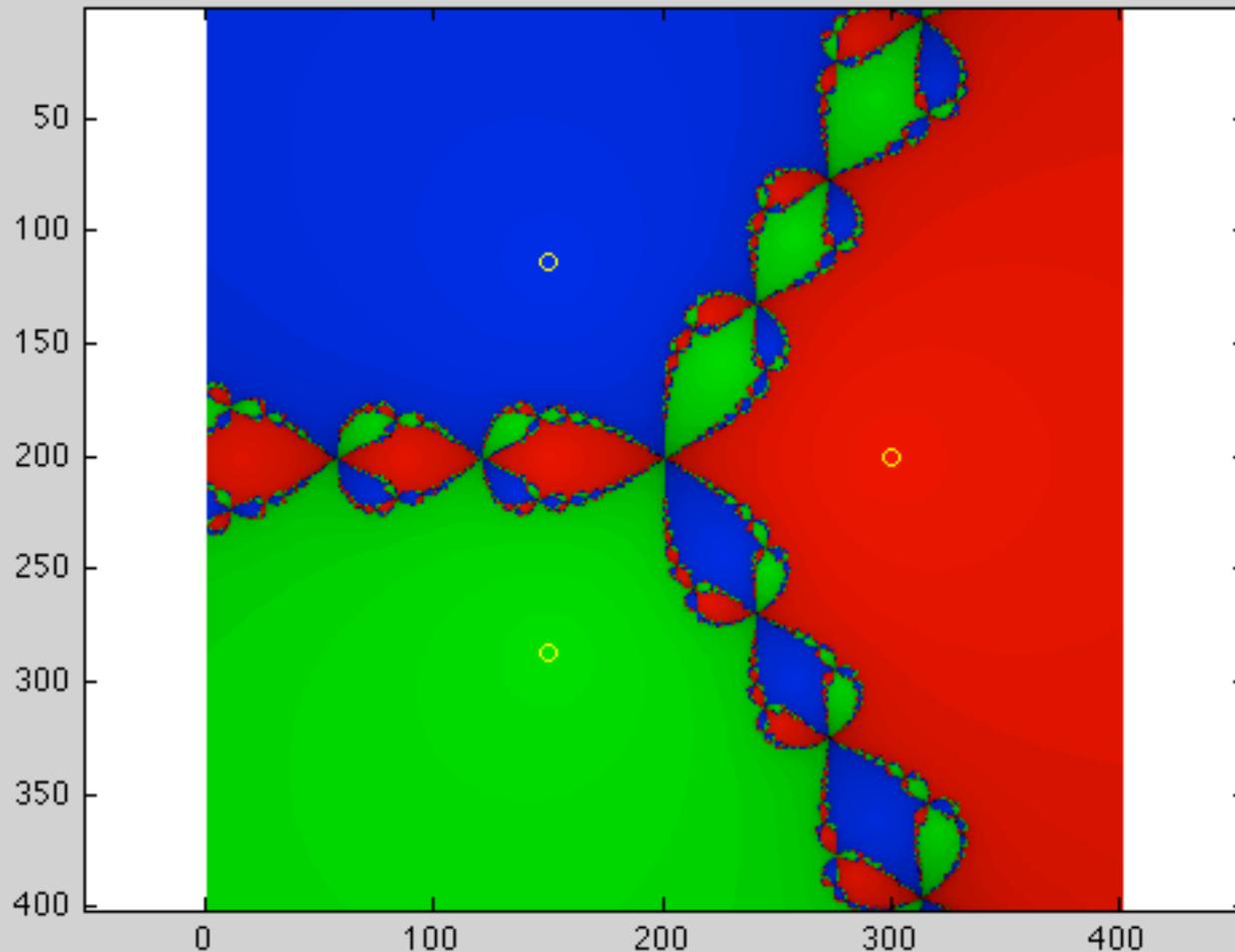


Here is what we get when we apply Newton's method and keep track of to where (color) and how fast (brightness/intensity) we converge..

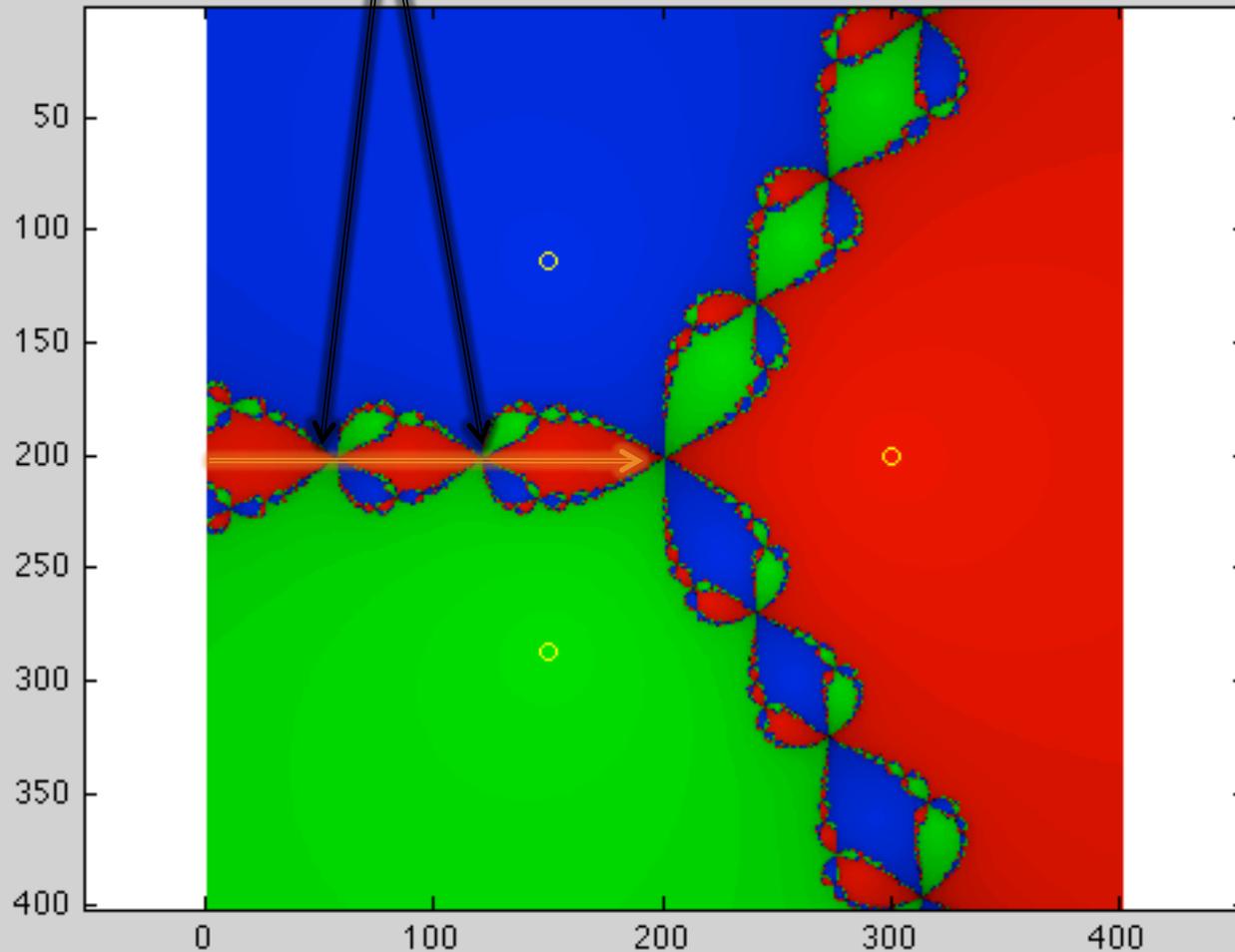


Notice that for most starting values the convergence is well behaved.

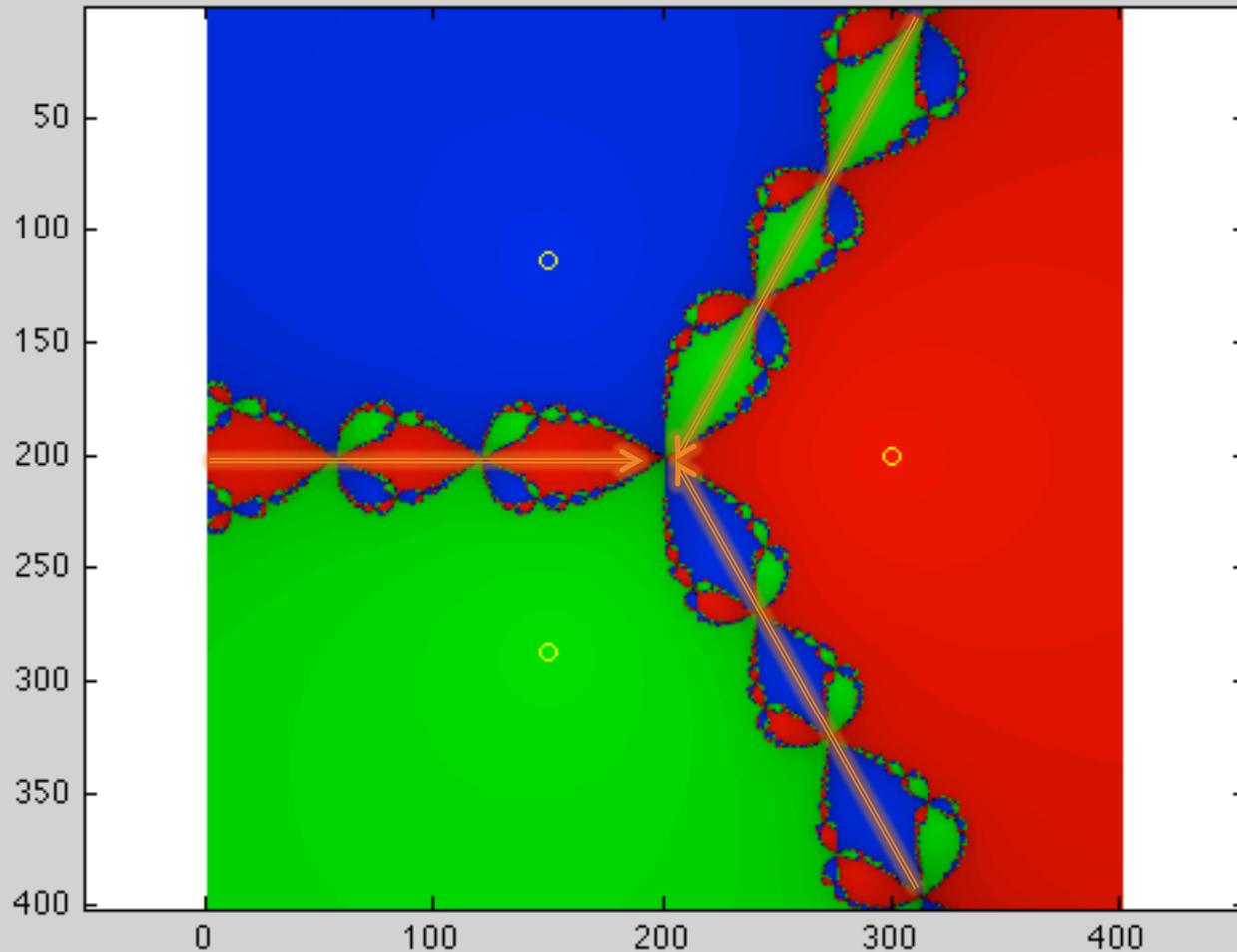
(the roots are arranged every 120° , convergence is relatively rapid, and the regions converging to each root are pie like sectors of 120° "width".)



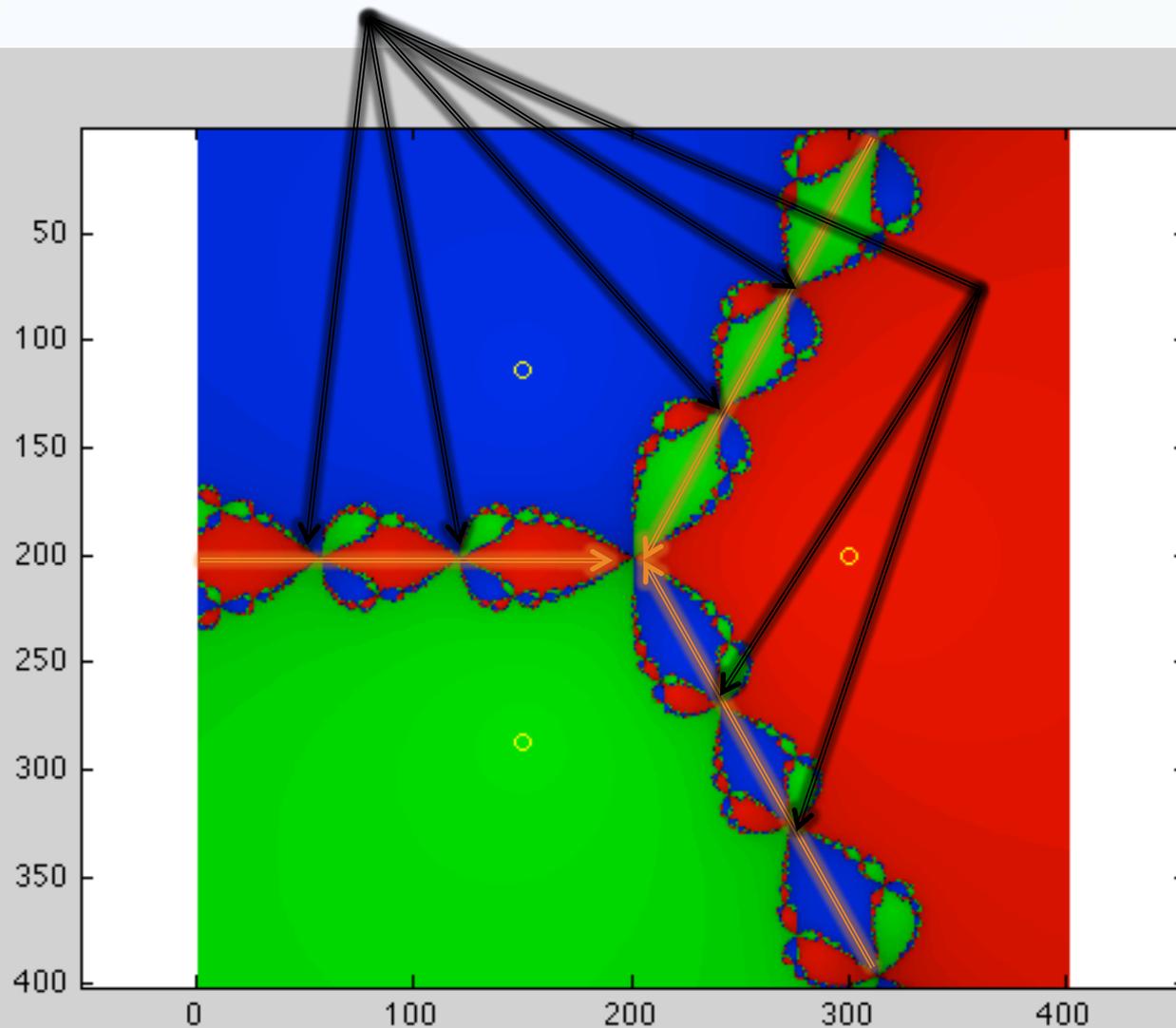
But there are some problematic places.
Remember we had some problems along the
negative real axis for the real case (where $f=0$).



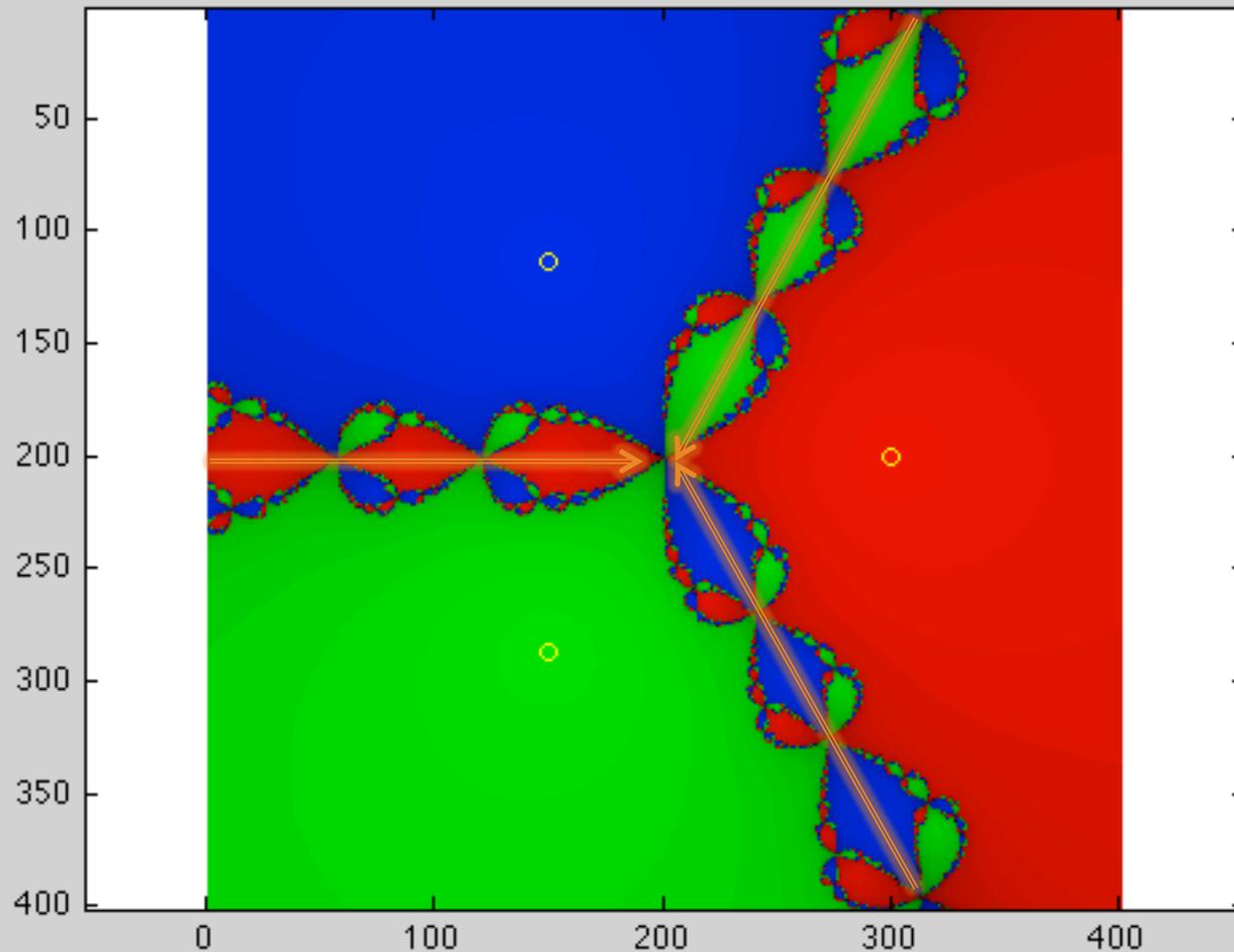
The starting values along the lines pointing towards the roots from the “other half plane” all have similar problems.



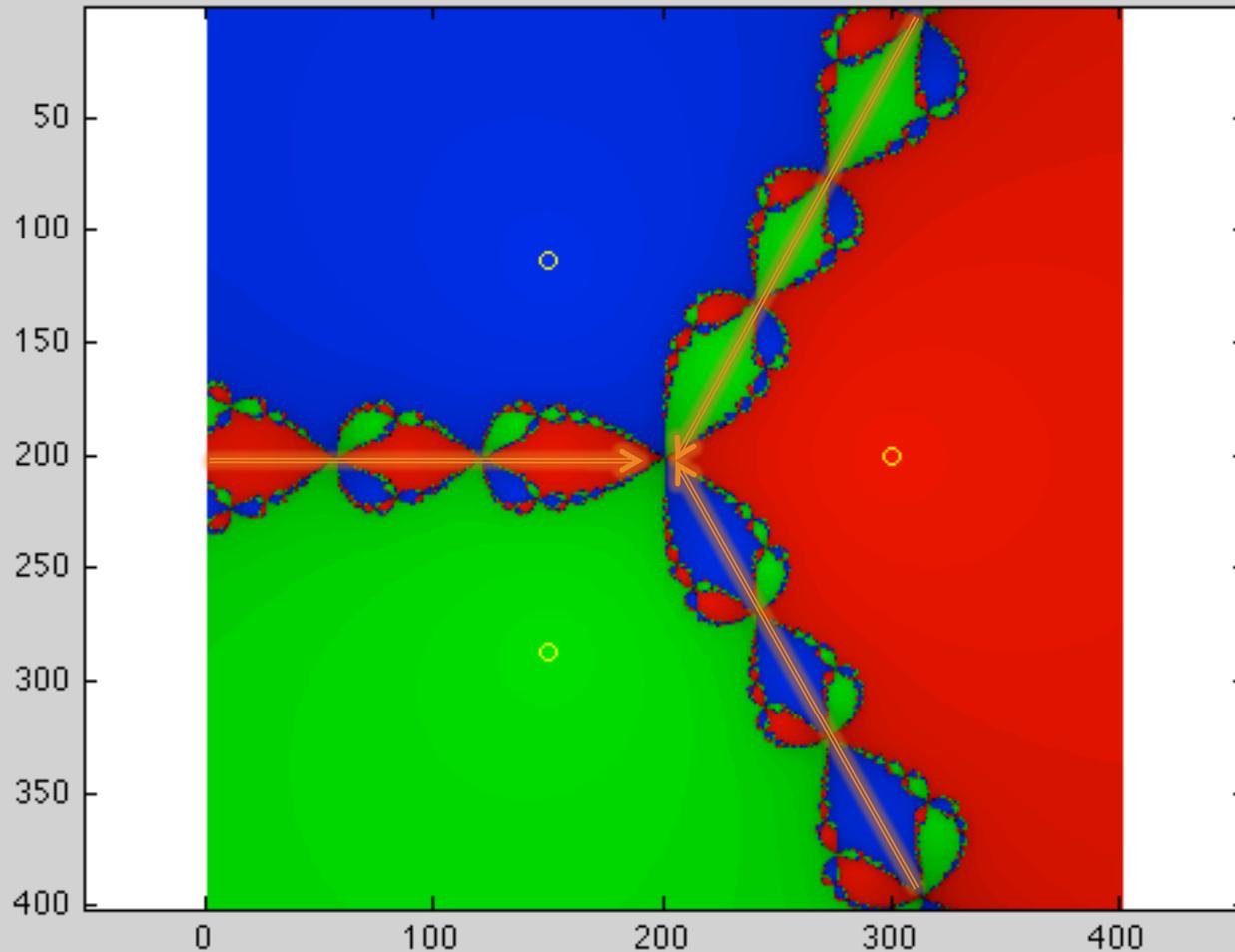
They are part of a structure of places with problems (same problem, they iterate to, or close to, $z=0$, where the slope is zero).



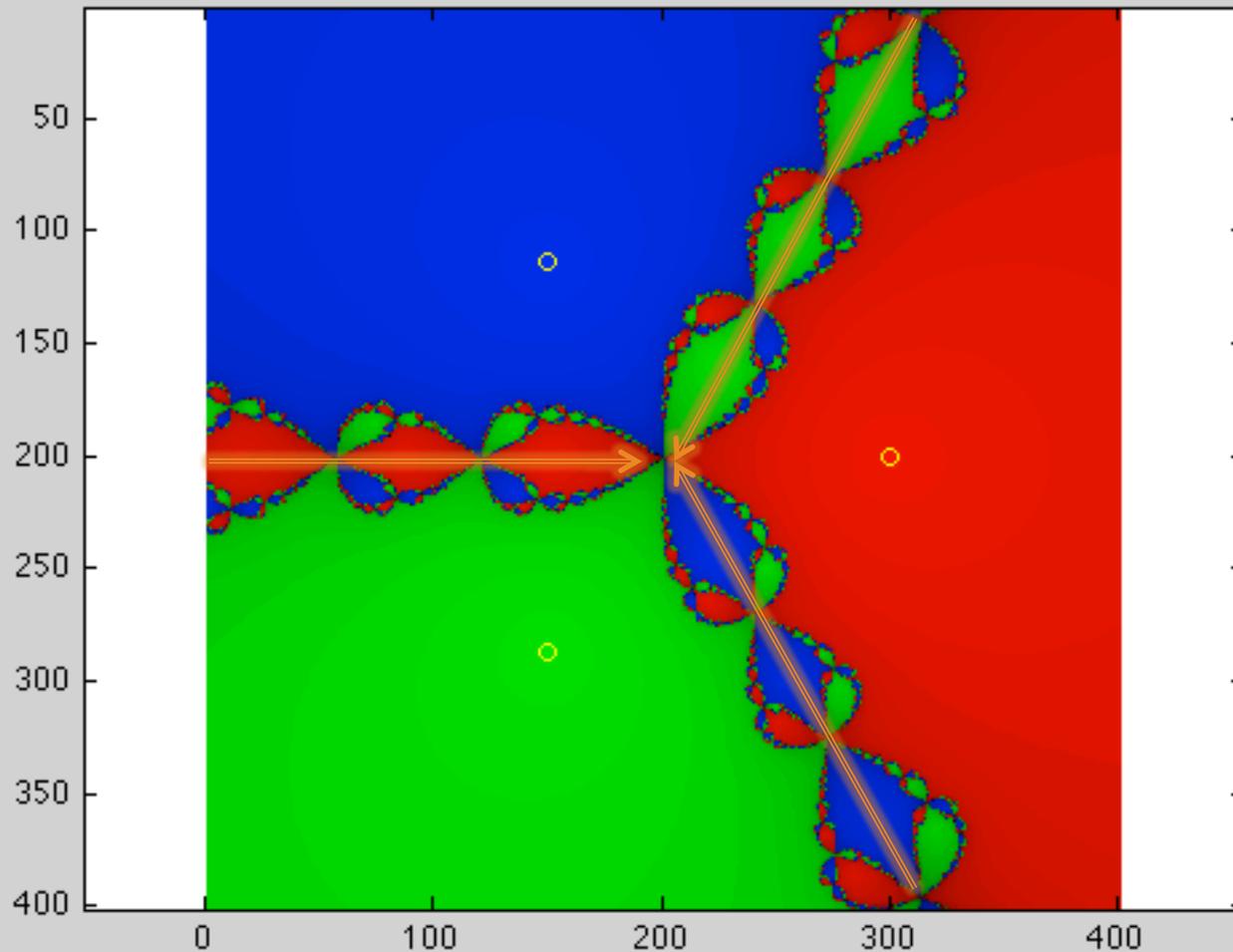
In addition, the division between regions of two different colors contains a region of the third color (except the points of no or slow convergence).

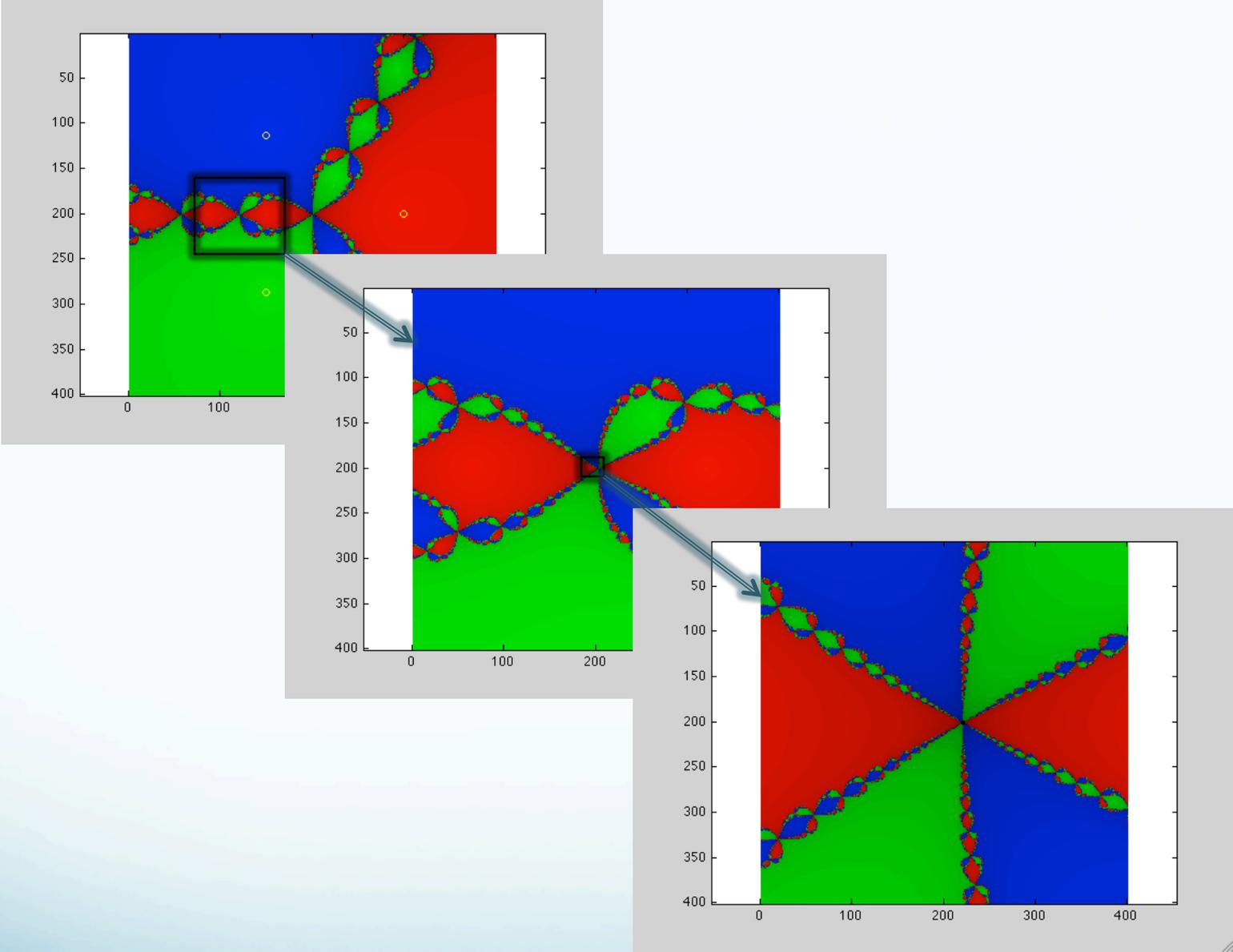


If one watched the path from these starting points, they would get near the problem points on the 3 lines dividing the 3 regions.



This color pattern behavior continues on all “boundaries” between colors, even as one zooms in on the boundaries.



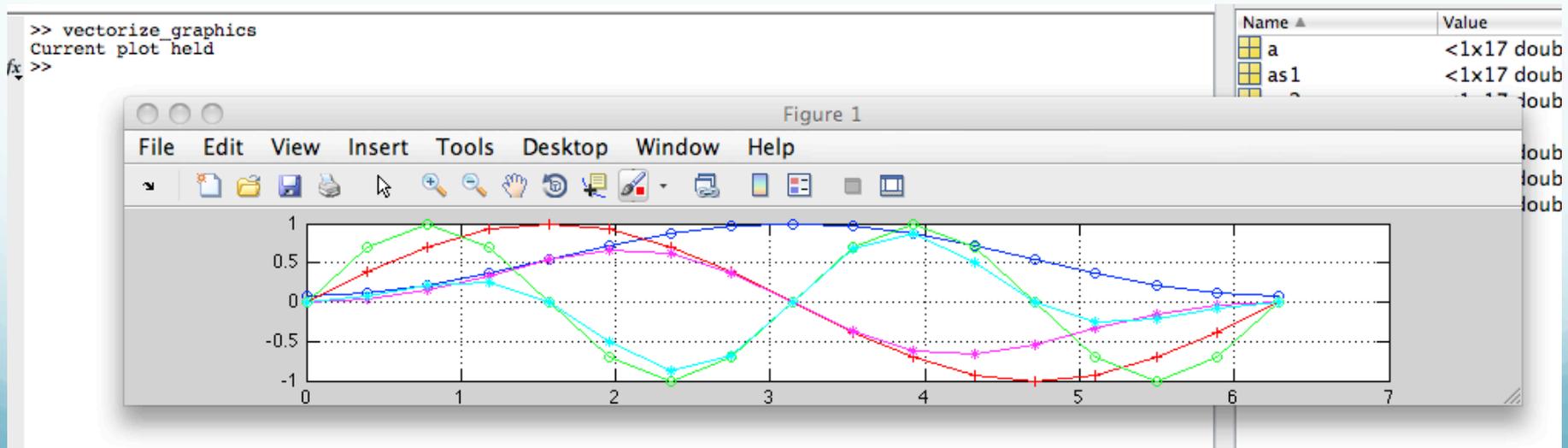


etc.

In most geophysical inversions we do not look carefully at the convergence properties of the system.

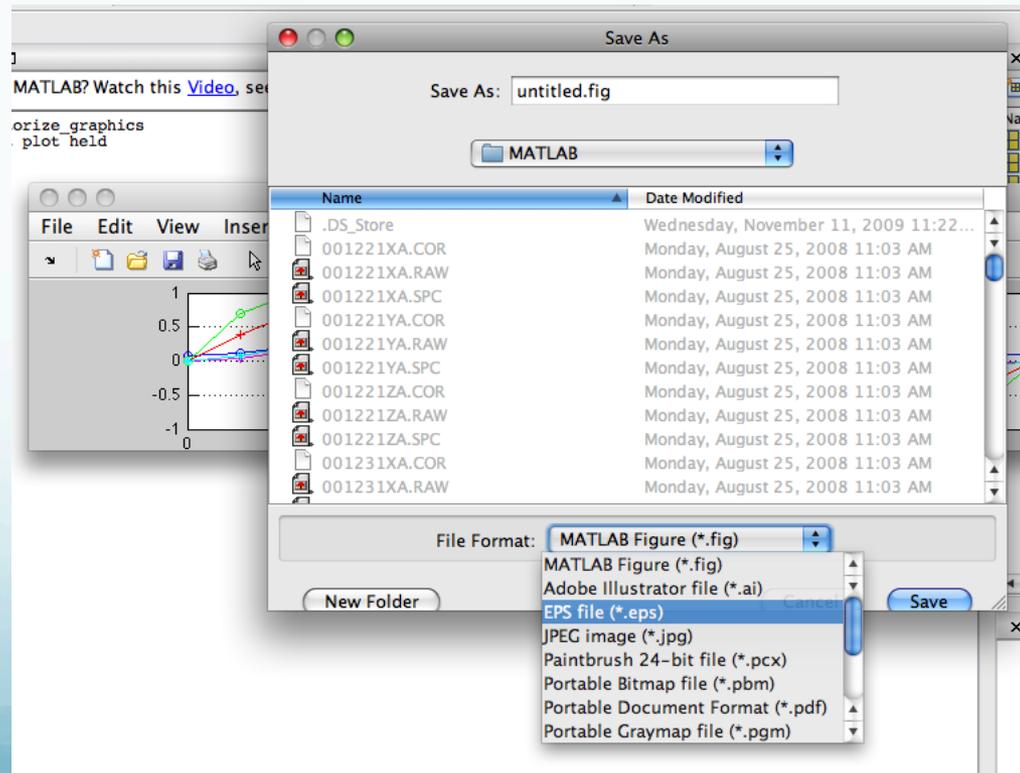
Make figure w/ Matlab etc.

(To make this figure I used “grab” on the MacOS to grab a selected area (the part below), copied it to the clipboard in grab, and then pasted it directly into PowerPoint)

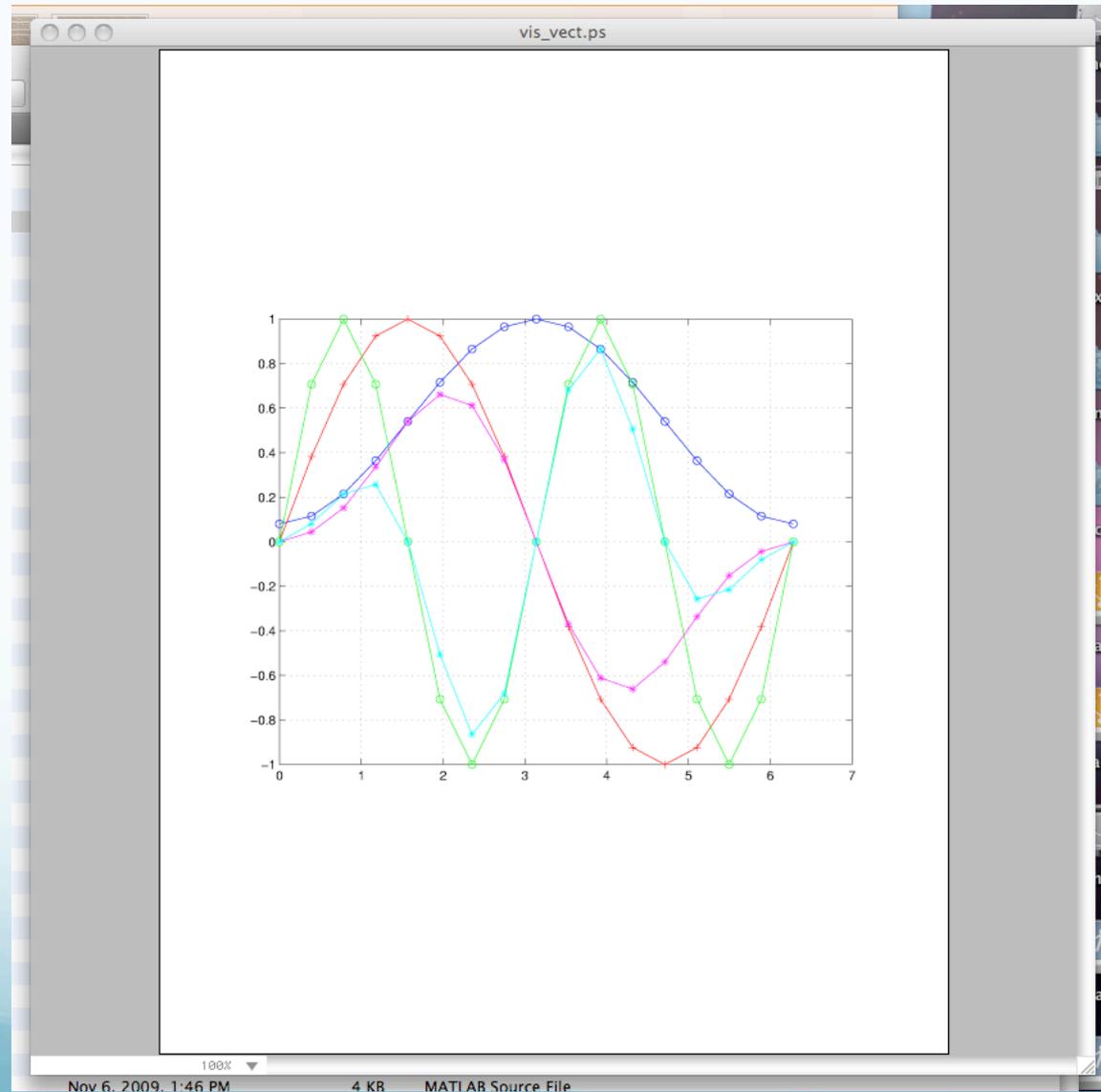


Store figure in ai, ps or eps formats.

(getting this figure into PowerPoint was a bit more complicated – since I wanted to get the dialog box. I used Grab again, but had to use the “self timer” mode. It grabs the whole screen after 10 seconds. So I started the timer, then selected SaveAs from the file menu and waited till it “took the picture”. I then used the MacOS GraphicConverter program to crop it to the part I wanted, copied that to the clipboard and pasted it into PowerPoint.)



Here is the postscript version of the figure displayed with MacGhostView.



Read ai, ps or eps format file into Illustrator.

"Select all" and "ungroup".

Get rid of bounding boxes, etc., use the "open arrow" (the solid arrow usually selects too much).

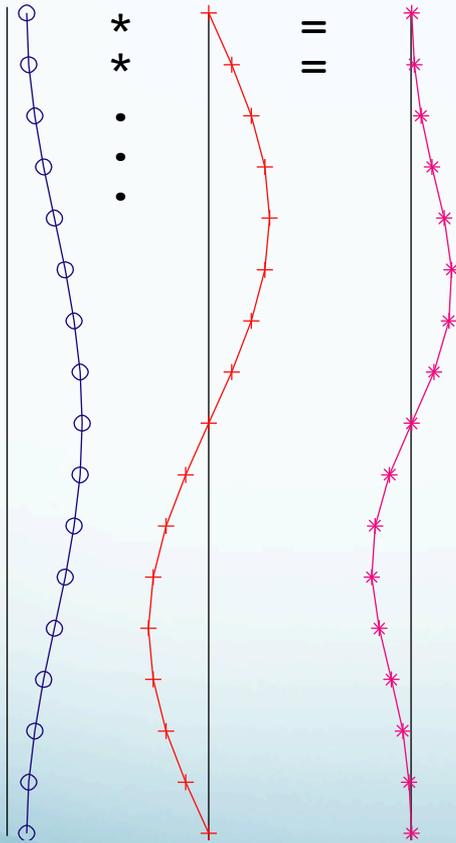
Use "open arrow" to select what you want to remove, use "select same fill and stroke" under the "select menu" to get all the stuff like it (be careful for stuff with same format you DON'T want to remove)

To modify figure – generally have to use the "open arrow" to select something, then use the "select same stroke", "select same color", "select same fill and stroke", etc. to select everything like it (careful that you don't get stuff you don't want to edit).

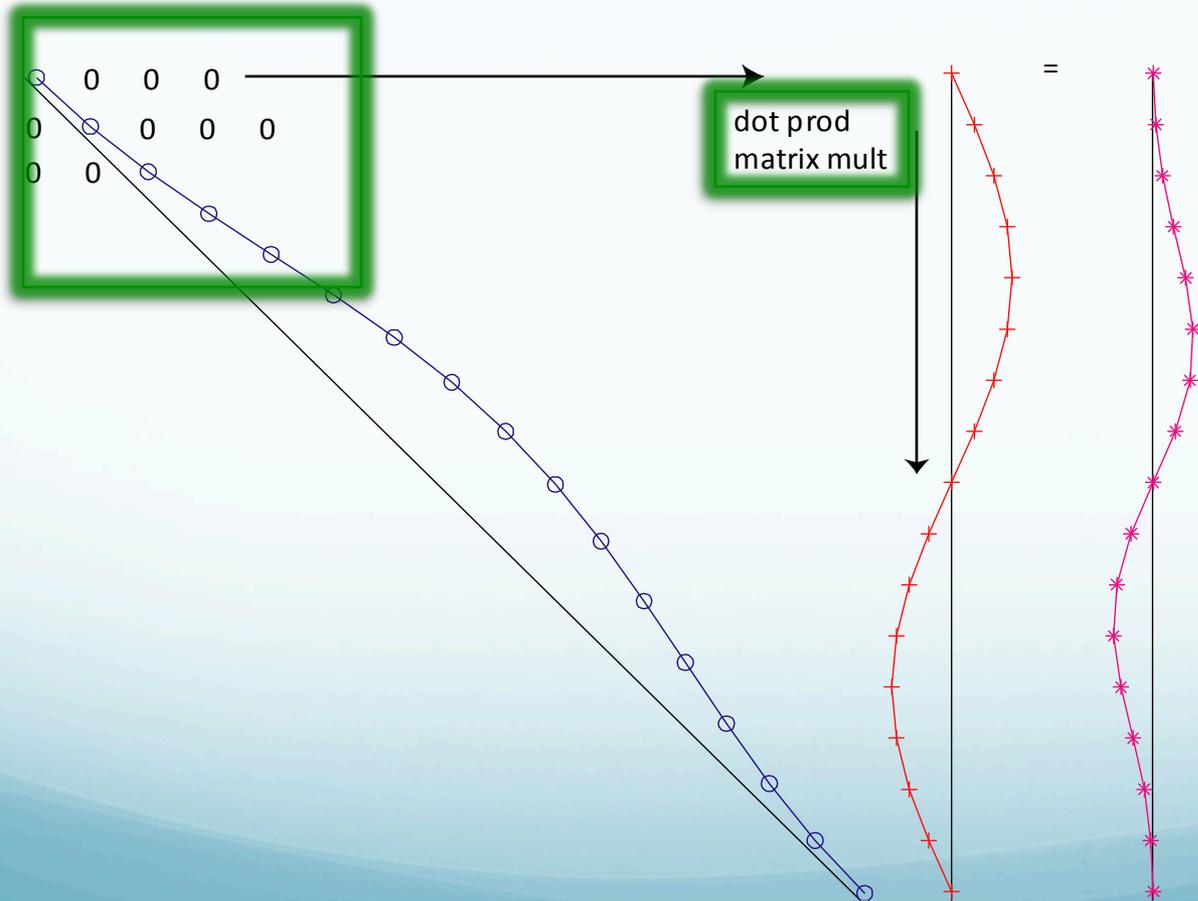
Then click the solid arrow to "group" the stuff you have selected.

Now you can move the selected items, change the properties (line properties (color, dashes, thickness, etc.) fill properties (be careful with filling lines – draws a new line from the beginning to the end and fills between it and the line), rescale them (as a group or individually), rotate them (as a group or individually), change their vertical position with respect to other items (things on “top”, overdraw things “underneath”).

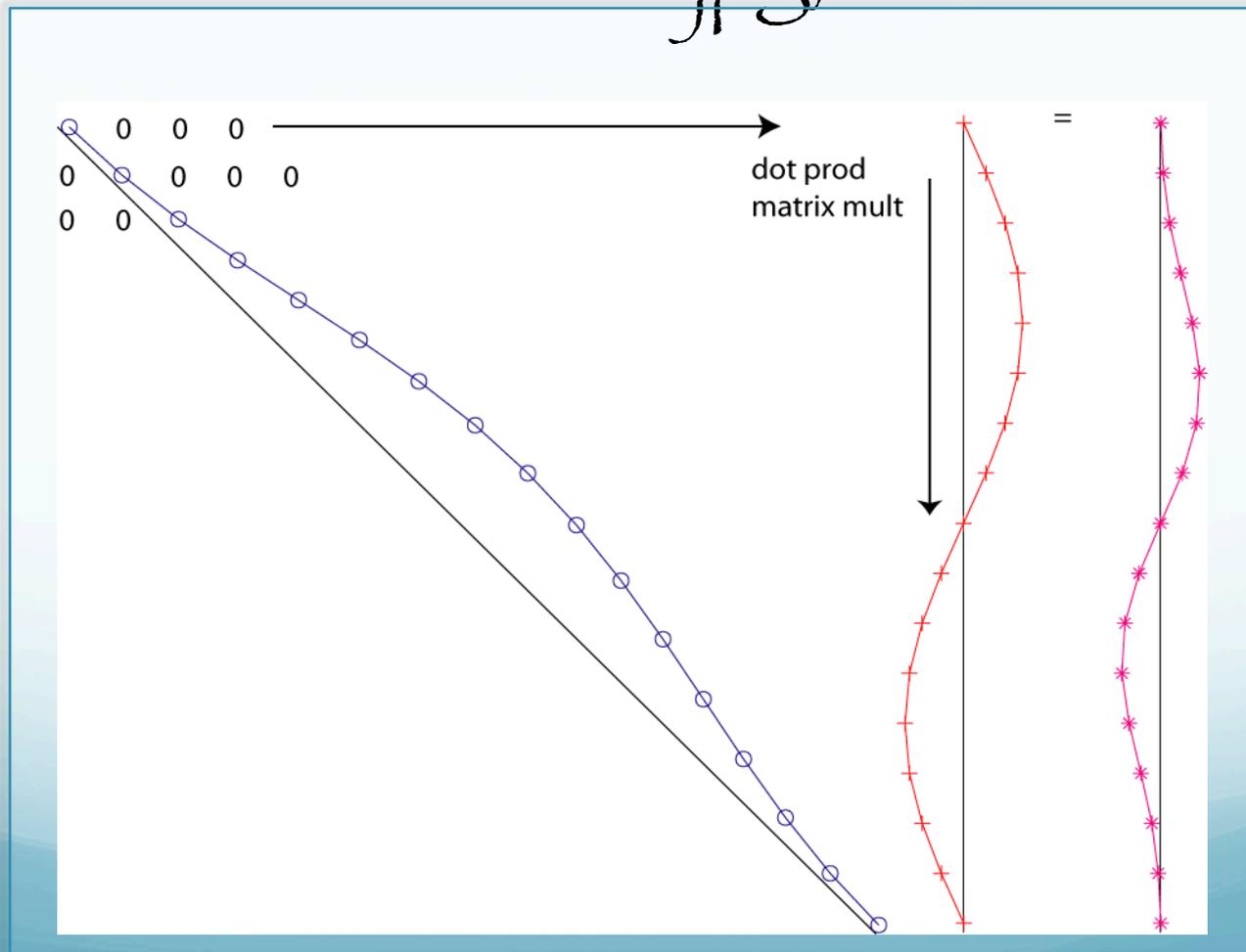
Pt by pt multiply to apply window to a vector,
 $w \cdot * v = wv$



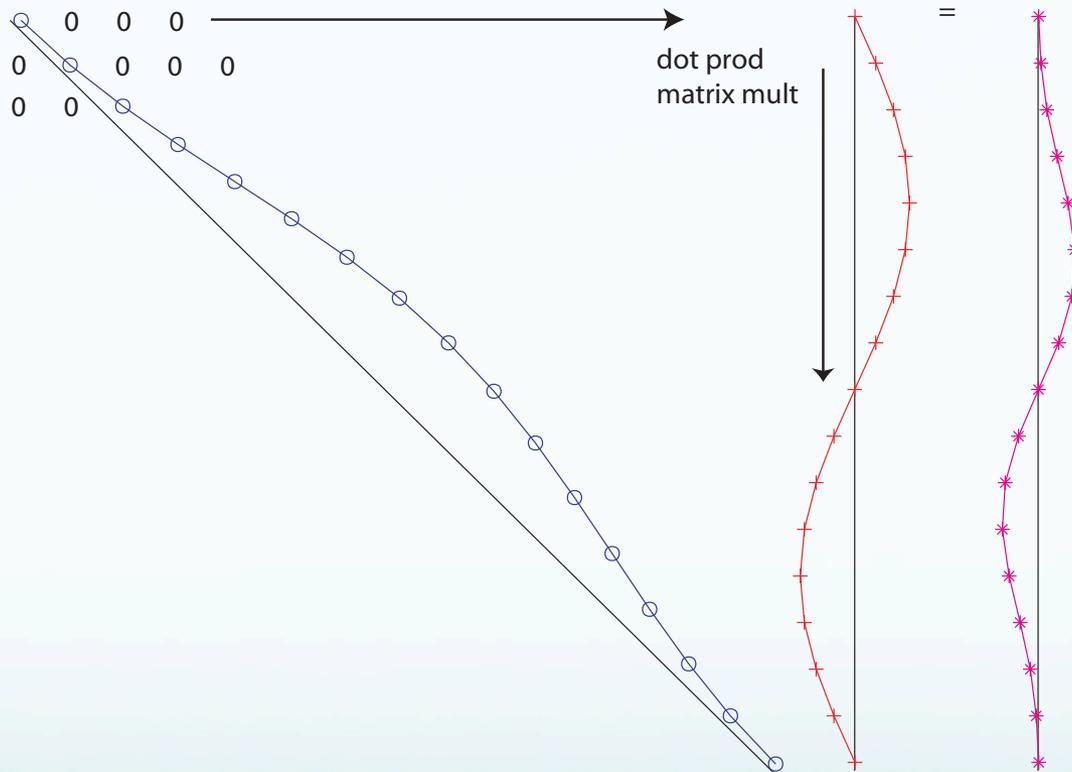
Sometimes it does not work – problem between Microsoft and rest of world. MWF format messes up. Can't edit it further in Illustrator.



Using "Save for Microsoft Office gets ugly white background (in box). PNG (another Microsoft) format. Can't edit further with Illustrator. Same with jpg, tif.

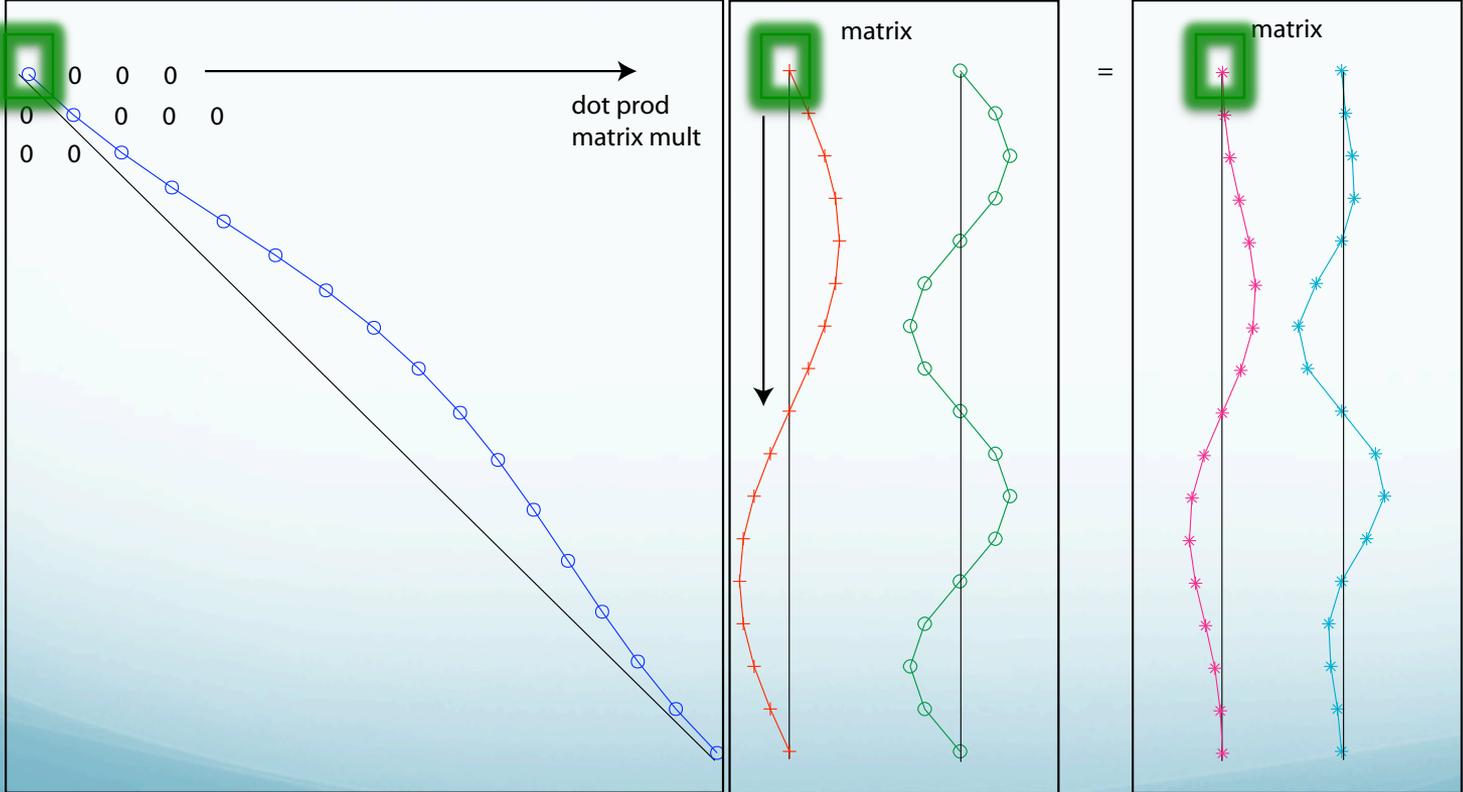


ai, ps and eps (encapsulated post script) works, can import to PowerPoint. Can edit in Illustrator.



If we put the window values on the diagonal of a matrix, we see that we get the same windowed vector when we multiply $w * v = wv$.

If have a number of traces to window, make matrix with new traces in new columns.



Use the sparse matrix function to save space and time with such matrices.

Edit map from gmt.

- 1) Transfer file to PC or Mac that has Illustrator
- 2) Drag and drop the ps or eps file or open it with Illustrator.
- 3) Select all with solid arrow and ungroup.
- 4) Select things you want to change (usually with open arrow), select “same fill and stroke” to get all such objects and can then modify parameters for the objects
(Illustrator works with “objects”, as opposed to pixels).

(eg. Make WBZ lines and great circle path lines thicker)

A little tricky keeping data sets (two or more sets of contour lines) separate – use slightly different colors when you make the gmt figure (eg. [255 0 0] and [255 1 1] for red – you can't see the difference, but they are different to the computer.)

Make altogether different color and dash it, etc.

Say I wanted to remove the focal mechanism labels.

They do not have anything unique about them for selecting them (I could have written them in a different color, but forgot).

So I'm stuck removing them "by hand".

If can't get all of 'em (they are under the FM balloons for ex.) you can select the "same fill and stroke" to get all the text, paste it into a new layer, edit it there, and then past it back into the original layer or leave it in the new layer. You also have to erase the original text in the first layer.

You can turn on/off the visibility of layers.
You can lock layers so they will not be edited.

To change the station labels, lock layer 1 (and turn it off). Then use the select tool (solid or open arrow) to select the station labels by drawing a box around them.

Then modify the properties.

Improve the display of the focal mechanism balloons by making the lines thicker (have to do the red and blue ones separately, the white circles are all done together).

Same for station symbols and gray earthquake circles.

Notice that you have control of both fill and boundary properties.

Scaling

Scale whole selection – select something, use “object”, “transform”, “scale” drop down menus.

Scale individual items – select them, use “object”, “transform”, “transform each” drop down menus. Changes size of item only, does not move around by scaling relative positions/whole figure.

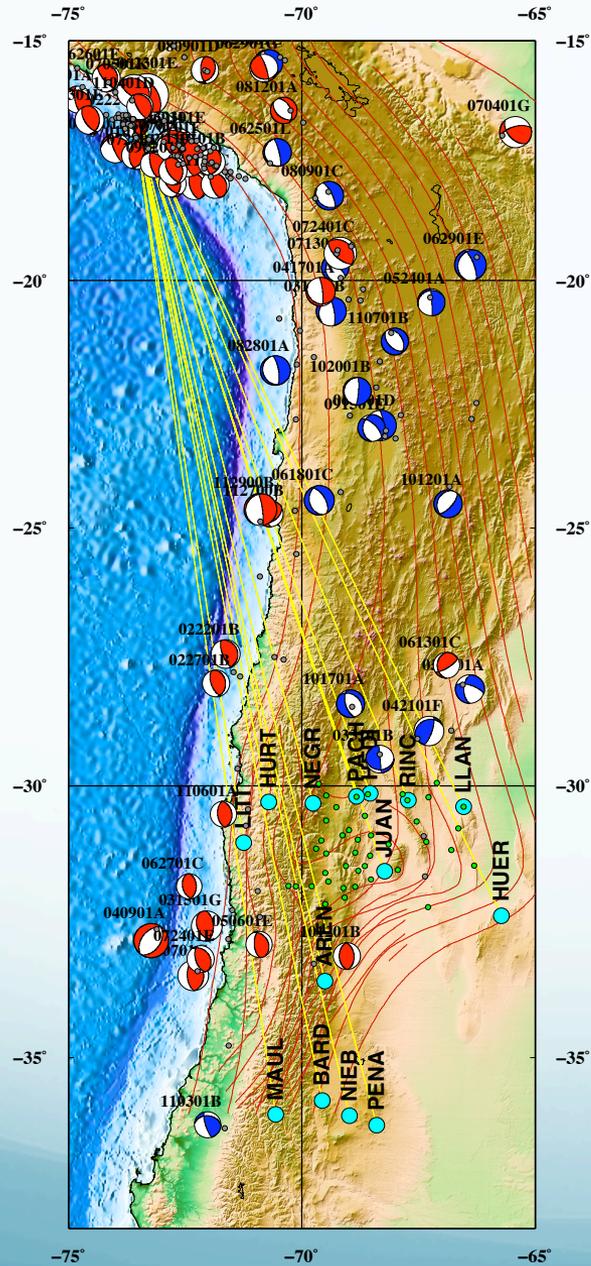
Transparency – select items, use “transparency” in formatting palette.

Topo is bitmap, can't change single color, range of colors, etc.

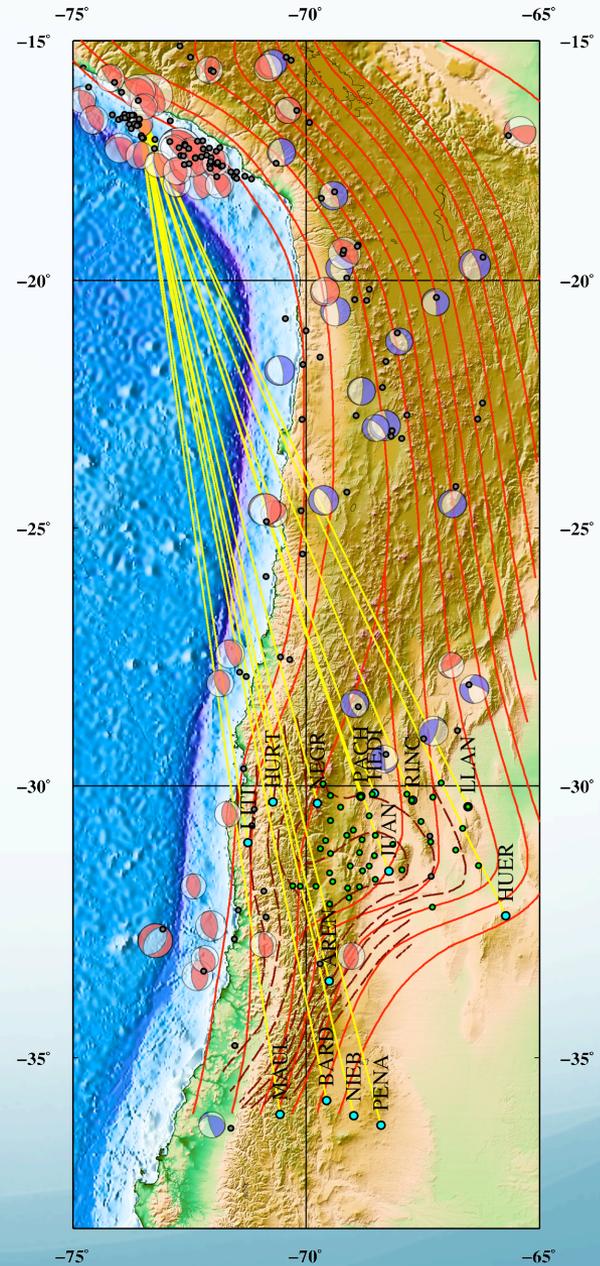
Can move topo up and down in display order (bring to front, send to back, etc.) and make transparent.

Can also move layers up and down (by dragging to where you want them in layer box.

Before



After



Can share multiple figures in one file using layers
(share common stuff in some layers, put stuff for
individual display in others).