MATH 448 - Project 1

Sam Snarr

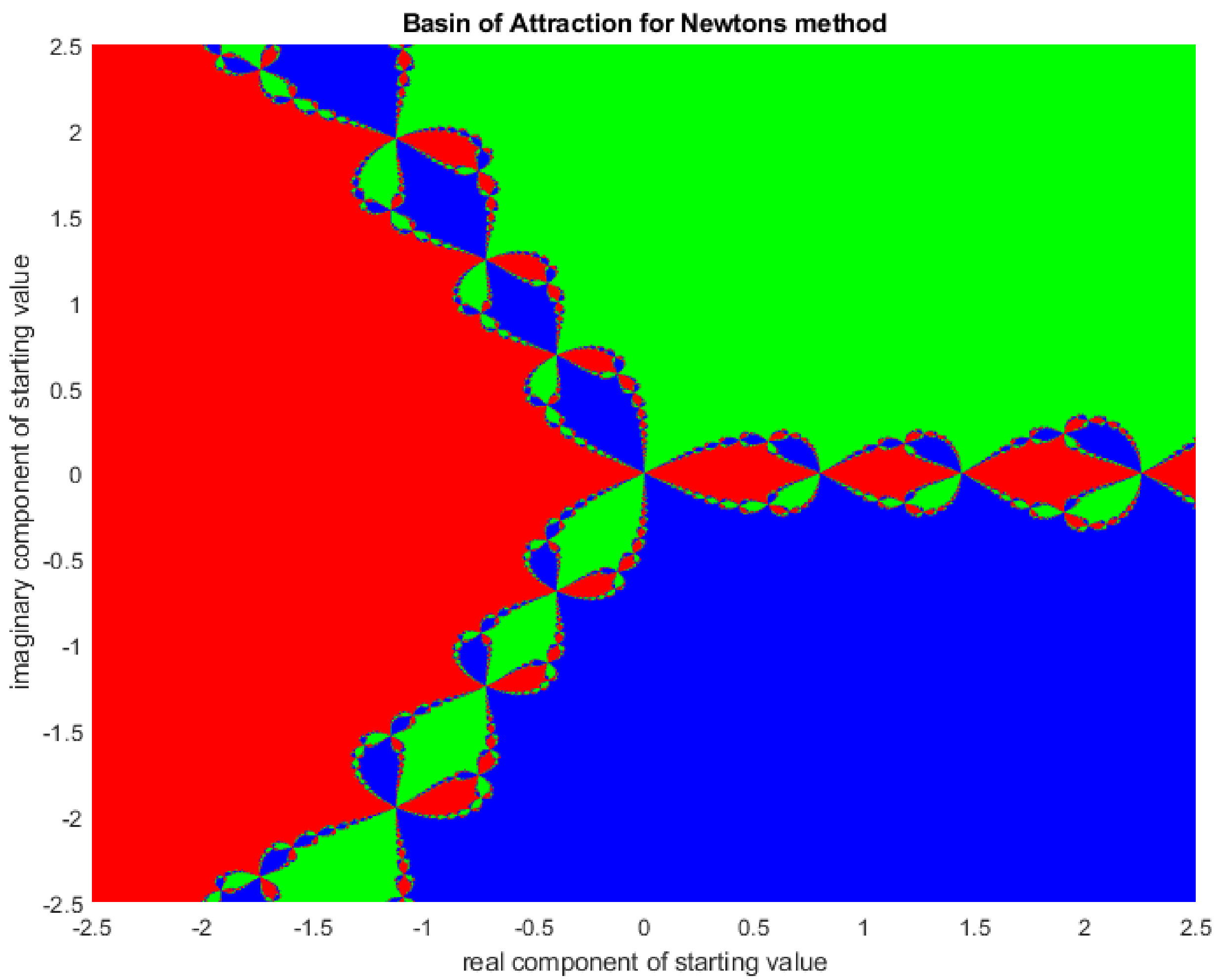
March 7, 2019

Please see attached code that is in `cfnewt.m`
I programmed this all by myself though I used the MATLAB documentation heavily. Function runs with the following arguments

`cfnewt(a, b, c, d, x1, y1, x2, y2, n)`
a, b, c, d - coefficients of polynomial $ax^3 + bx^2 + cx + d$
x1, y1 - bottom left coordinate on plot
x2, y2 - top right coordinate on plot
n - number of columns of pixels (total number of pixels is n^2)

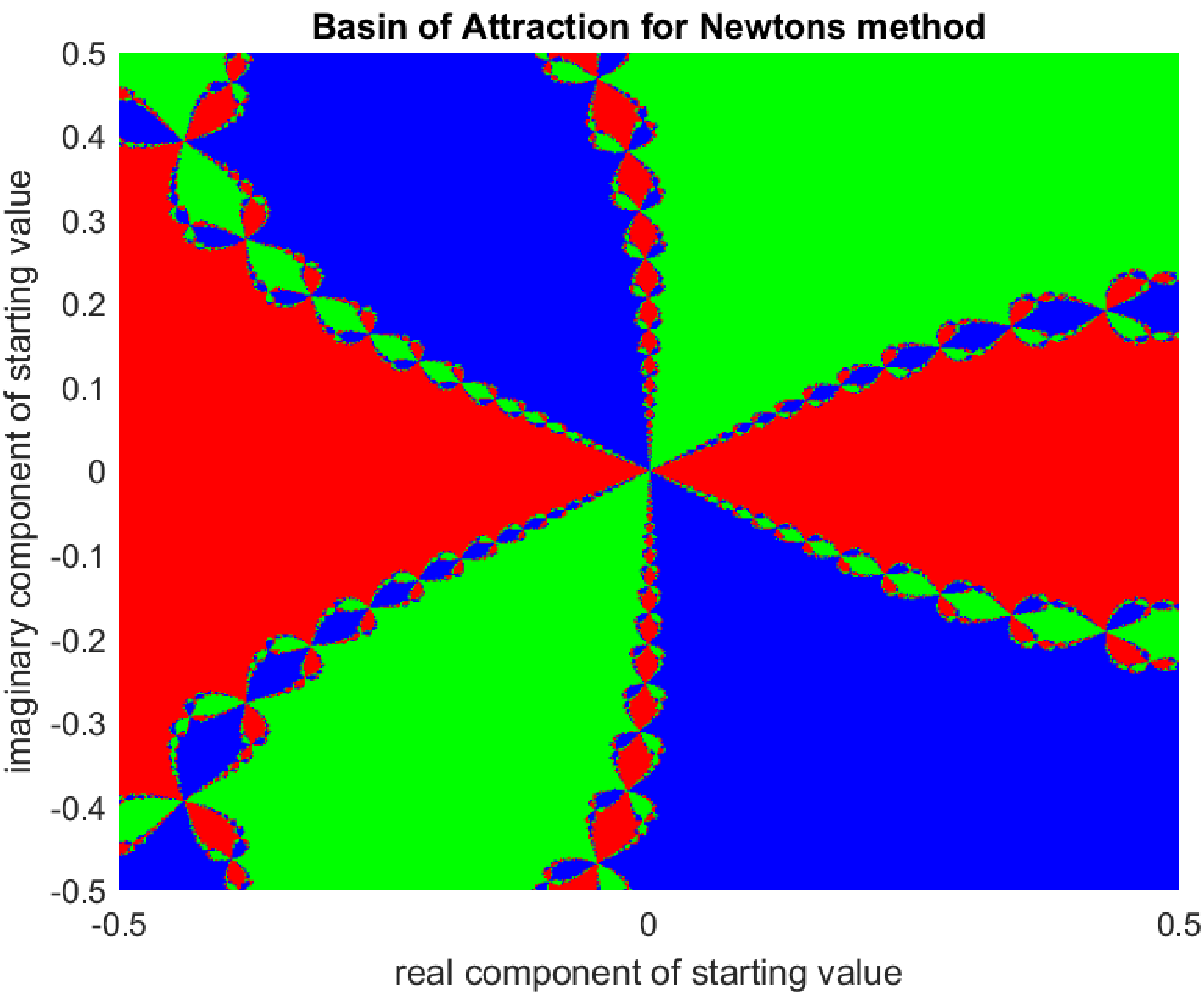
```
>> cfnewt(1, 0, 0, 1 ,-5, -5, 5, 5, 1500)
```

Figure 1: Basins of attraction for $y = x^3 + 1$



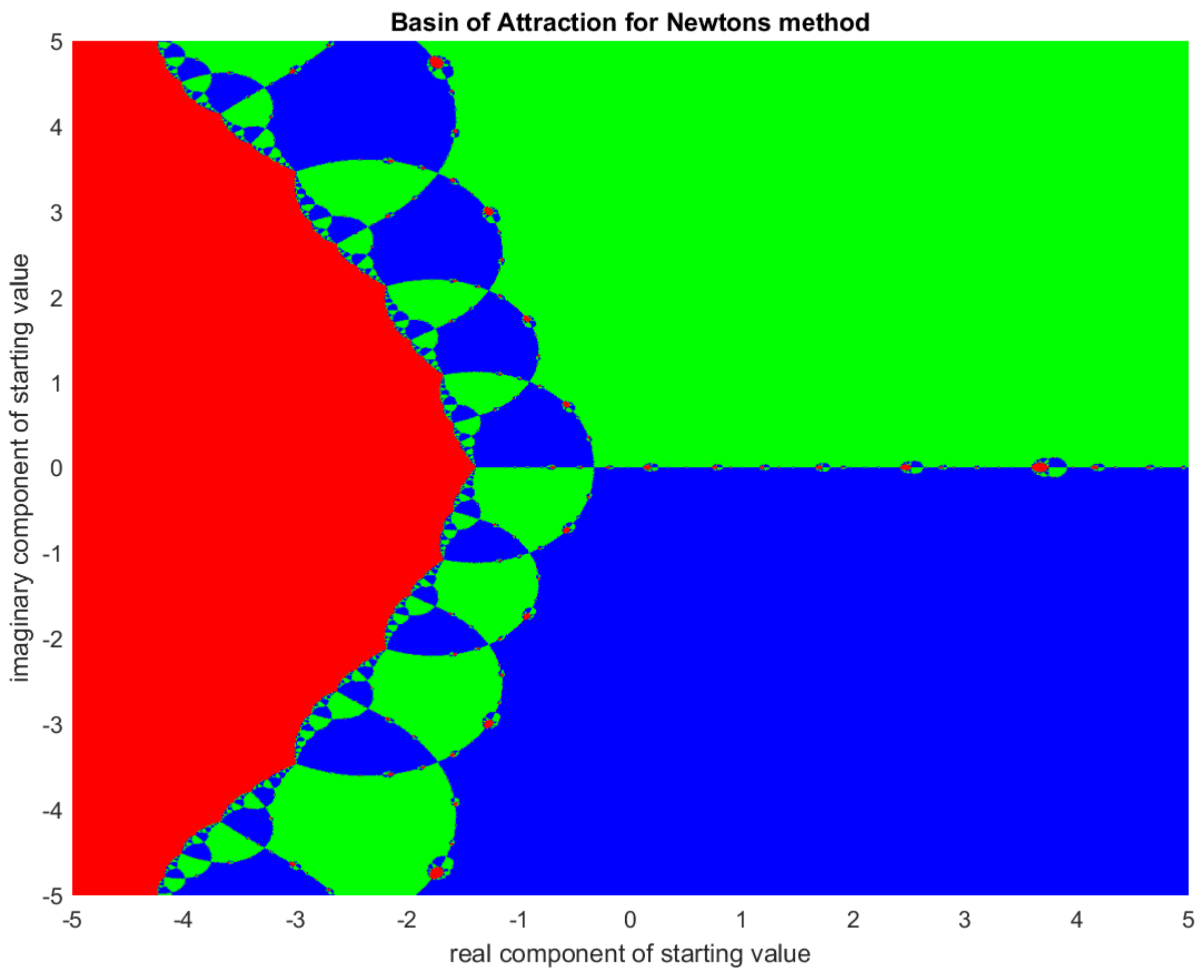
```
>> cfnewt(1, 0, 0, 1 ,-0.5, -0.5, 0.5, 0.5, 600)
```

Figure 2: Basins of attraction for $y = x^3 + 1$ zoomed



```
>> cfnewt(1, 1, -3, 2 ,-5, -5, 5, 5, 1000)
```

Figure 3: Basins of attraction for $y = x^3 + x^2 - 3x + 2$



For other cubics the fractal pattern is there but it usually of a different form. Some fractals are more interesting than others.

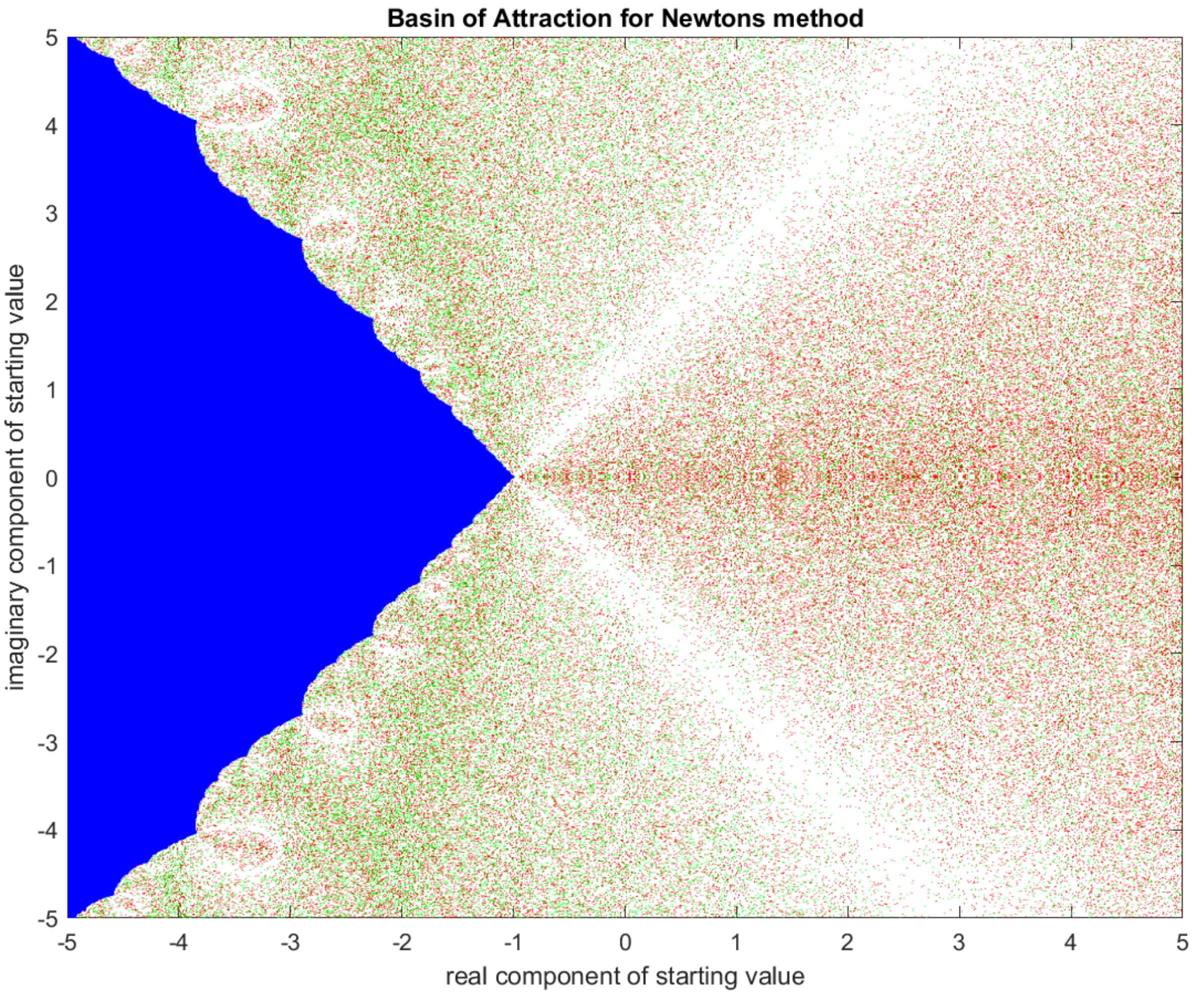
Figure 4: Sample output that comes out along with a plot

```
>> cfnewt(1, 1, -3, 2 ,-5, -5, 5,5, 1000)

Starting rendering process...
Working on pixel column 100 out of 1000
Working on pixel column 200 out of 1000
Working on pixel column 300 out of 1000
Working on pixel column 400 out of 1000
Working on pixel column 500 out of 1000
Working on pixel column 600 out of 1000
Working on pixel column 700 out of 1000
Working on pixel column 800 out of 1000
Working on pixel column 900 out of 1000
Working on pixel column 1000 out of 1000

Plotting...

Roots
Red:   -2.511547+0.000000i
Green: 0.755774+0.474477i
Blue:  0.755774-0.474477i

Elapsed time is 15.818267 seconds.
```

```
>> cfnewt(1, 3.0011, 3.0022, 1.0011 ,-5, -5, 5,5, 1000)
```

$$y = (x + 1)(x + 1.0001)(x + 1.001)$$

Figure 5: Output when all 3 roots are close to each other.



When a few of the roots are close to each other there is not much of a visible pattern in the basins. The majority of the points are scattered and many are not colored. This could just be the property of the basins when all the roots are close but it is most likely because of a problem with the tolerance. The starting points may be assigned the wrong colors because the tolerance is greater than the difference between the roots. Decreasing the tolerance of the newton's method as well as for the color assignment did not really help much with this scattering/color error. Newton's method does not behave well for a double root anyways; it is clear that when the roots are close together the time it takes to run this program is increased dramatically. Many of the approximated roots from `newton.m` converge so slowly that they hit the n=50 maximum on the loop rather than the tolerance. This causes many of the points to not be close to any of the actual roots an therefore the point is not assigned a color. This is the reason for so many white spots on the plot.