Task 1

For this task I modified the line `p = p0 - f(p0)/df(p0)` in `newton.m`.

I changed it to these 3 lines:

```
q = p0 - f(p0)/df(p0);
term = (f(p0)-f(q))/(f(p0)-2*f(q));
p = p0 - (f(p0)/df(p0) * term);
```

I also changed the convergence check for `ostrowski.m`.

```
if abs(q - p) < TOL
```

Task 2

For this task I modified the `newton.` and `ostrwoski.m` to return whether they converged or not.

I got a range of 1000 x values between [-10 10] and applied the root finding algorithms where x = p0.

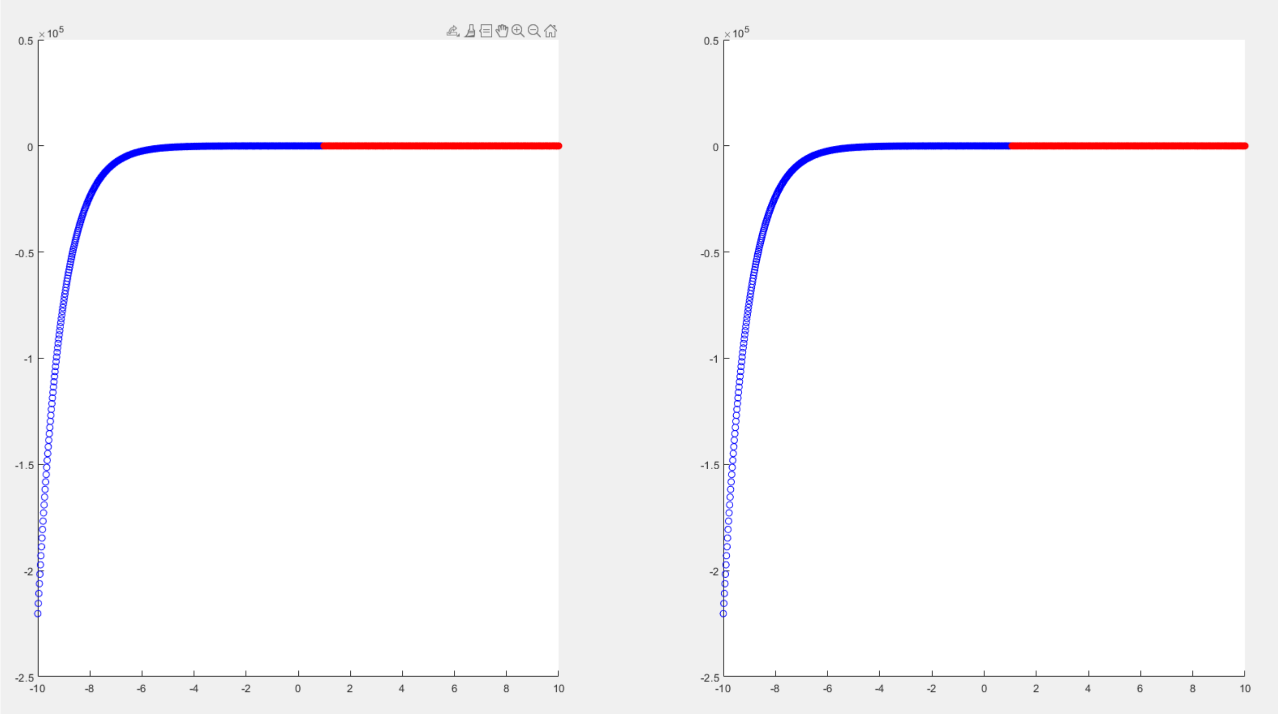
```
x = linspace(-10,10,1000);
```

```
[~, temp_c1] = Newton(f, df, x(i), TOL, N0);
[~, temp_c2] = Ostrowski(f, df, x(i), TOL, N0);
```

I plugged in the x values into the original function to sample it.

```
func(i) = f(x(i));
```

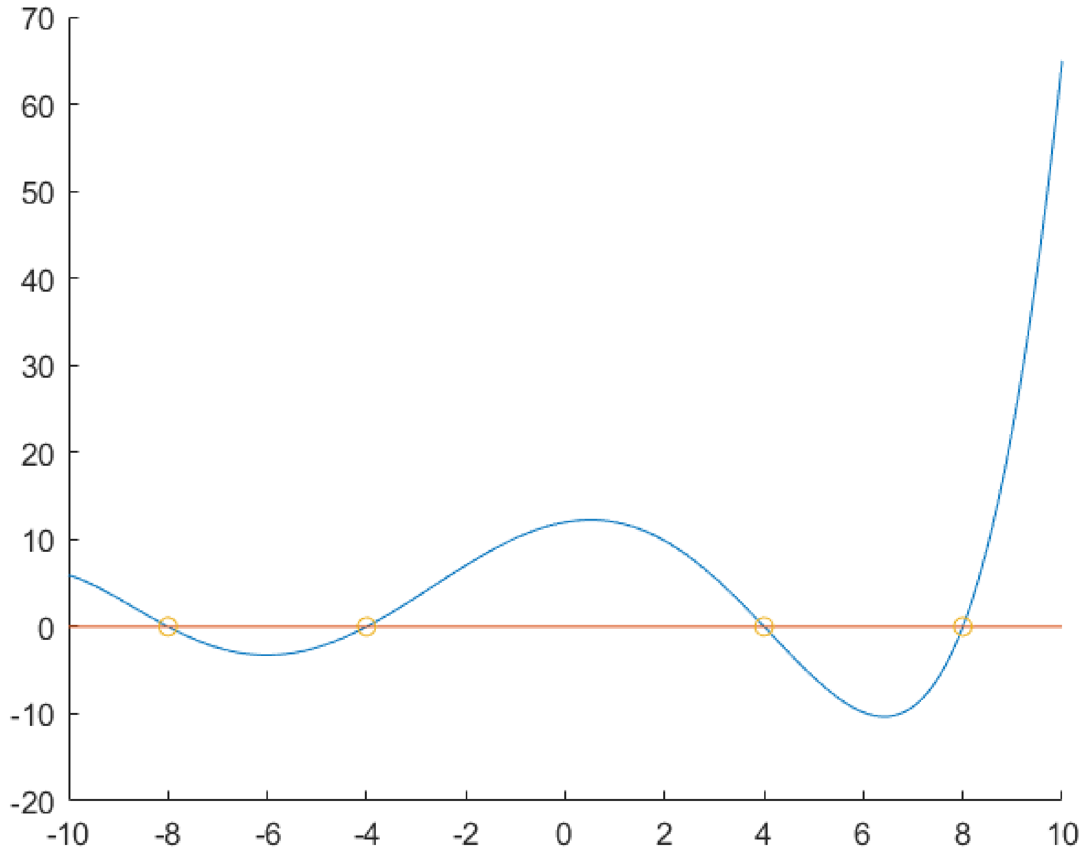
Using the sampled function and the modified methods (to check for convergence), I plotted the graphs.



Task 3

---

1. Until i = 10
    a. Run `newton.m` on the polynomial and store the root
    b. Divide the polynomial by (1 - root)
    c. If the degree of the new polynomial equals 0, then exit the for loop



Task 4

1. Check if xMin and xMax have different signs.
    a. If they do, then appropriate roots have been found and I exit.
    b. If not then set xMin to (xMin + xMax)/2.
2. Check if |xMax - xMin| < range/2^20, where range is the absolute difference between xMax and xMin a the very beginning.
    a. If this is true, we exit.
3. We then repeat this until we exit.

Task 5

```
[xMin, xMax] = BisectionInitialise(f, xMin, xMax);

% run bisection 5 times
p = Bisection(f, xMin, xMax, TOL, 5);

% refine using Ostrowski
p = Ostrowski(f, df, p, TOL, N0);
```

This is very simple, I run `BisectionInitalise`, run the bisection algorithm 5 times, and refine the answer using `ostrowski.m`.

Task 6

For this task I modified `newton.m` and `ostrowski.m` to return the number of iterations.

I then generated 400 values between -1 and 1 to create 400x400 pixels. Using these values I generate 1600 versions of the equation $x = a + bi$.

1. For every possible value of x:
    a. Find root, iterations and if it converged with `Newton.m` with x as p0
    b. Find root, iterations and if it converged with `Ostrowski.m` with x as p0
    c. If the roots are unique store them in an array roots
    d. Map the roots found in step 1.b and 1.c to unique roots
2. For every unique root assign a distinct colour.
3. For every root in the 400x400 grid, replace the root with the assigned colour to create an image.
4. For every iteration value in the 400x400 grid, replace the iteration value with log(num of iterations) to create a greyscale image.
5. Add the two images together and divide by 2.