Prob 4

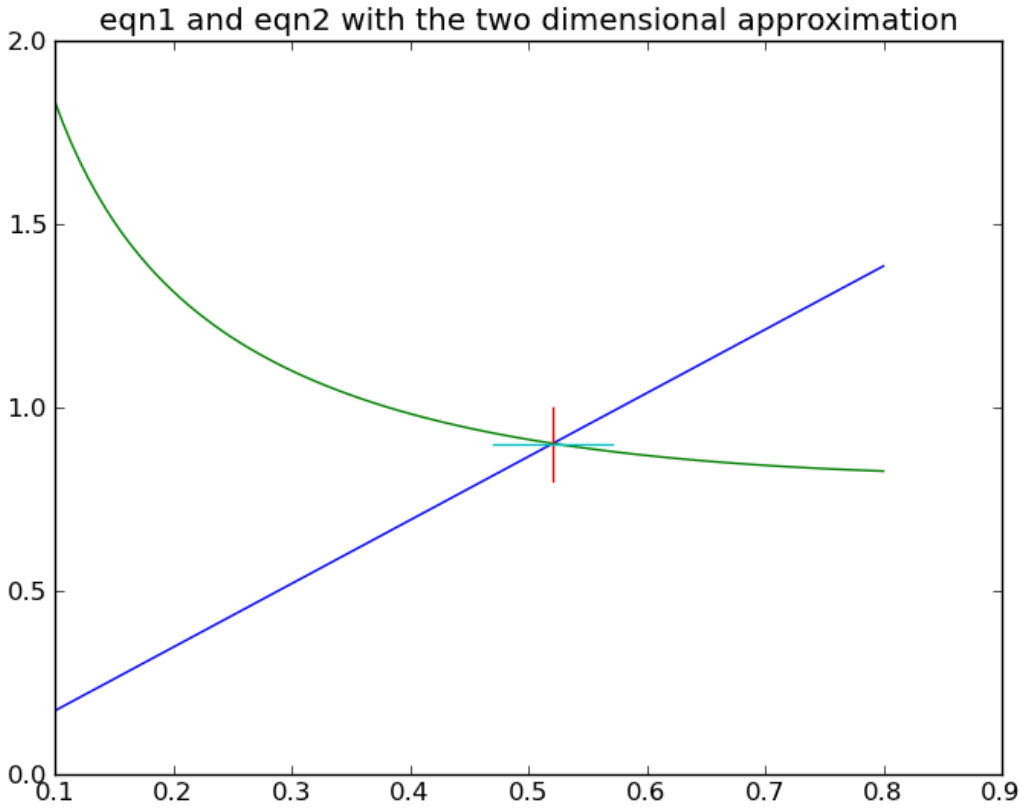
```
***Solving using the two dimensional version of Newtons
method***
        Our equations are:
3x^2 -1y^2  = 0
-1x^2 3xy^2 -1  = 0

The first iteration of Newtons methdo using a guess of [1,1]
gives us:
[0.631578947368421, 0.8947368421052632]

using the infinity norm our error is:
0.368421052632
...
Iteration 2:
[0.5147475360559554, 0.8686913766633854]
Error:  0.130576283232
...
Iteration 3:
[0.5223831442795319, 0.9054436885714463]
Error:  0.0423076743886
...
Iteration 4:
[0.5204292071132371, 0.9014124892667191]
Error:  0.00445218113021
...
Iteration 5:
[0.5208678548198985, 0.9021695863387289]
Error:  0.00083990080127
...
Iteration 6:
[0.5207670846235098, 0.901995049477007]
Error:  0.000193463473348
...
Iteration 7:
[0.5207901245087303, 0.9020349557292424]
Error:  4.42422076025e-05
...
Iteration 8:
[0.5207848509316706, 0.9020258216258374]
Error:  1.0126108026e-05
...
Iteration 9:
[0.5207860576928399, 0.9020279117974953]
Error:  2.3171971442e-06

***Solving by using the one dimensional method***
```

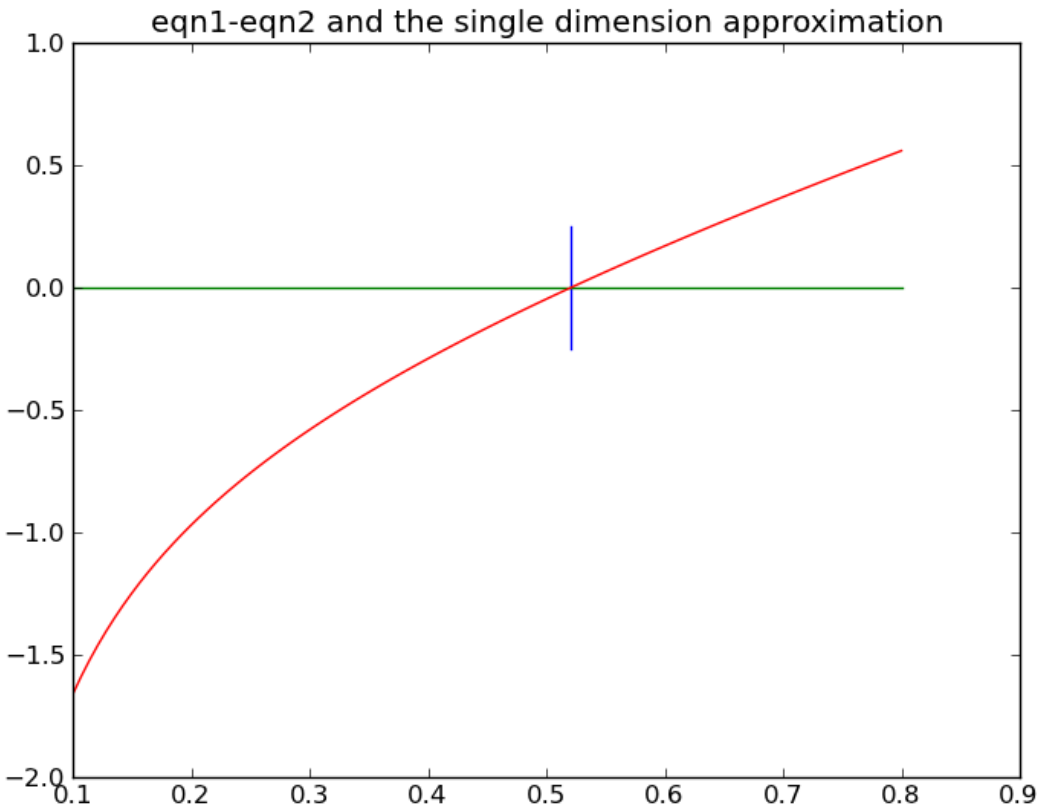
```
For this case f(x) is first equation minus the second equation
for the first iterations we will guess the x value is 1, which
makes the f(1)=  0.915554226641
which gives us the x1:  0.664883894507

using the infinity norm our error is:
0.335116105493
...
Iteration 2:
f(x1) =  0.301337066928
x2 =  0.531894517046
Error:  0.200018948511
...
Iteration 3:
f(x2) =  0.0246140765581
x3 =  0.520863934851
Error:  0.25002960775
...
Iteration 4:
f(x3) =  0.000174045913882
x4 =  0.52078586871
Error:  0.000149900651395
...
Iteration 5:
f(x4) =  7.96710600826e-08
x5 =  0.520785832975
Error:  6.86187734287e-08
...
Iteration 6:
f(x5) =  3.2717273335e-11
x6 =  0.52078583296
Error:  2.81786447192e-11
...
Iteration 7:
f(x6) =  1.3433698598e-14
x7 =  0.52078583296
Error:  1.15118422076e-14
...
Iteration 8:
f(x7) =  2.22044604925e-16
x8 =  0.52078583296
Error:  2.13182263103e-16
...
Iteration 9:
f(x8) =  -1.11022302463e-16
x9 =  0.52078583296
Error:  0.0
```

For this problem we first expanded the polynomial.py file from our first practice homework to include polynomials with 2 variables and an evaluate function to solve the polynomial for a given x and a derivative function for polynomials with 2 variables.

Then we created the newtons2d function that when given 2 of the above polynomials and a guess vector, produces the next iteration of newtons method. We also created the newton function that does the same thing for single variable functions. Finally we made functions to check the tolerance of our guesses.

I was really surprised at how quickly the single variable versions error shrunk. After 9 iterations it is 10^-16 away from 0. The two variable version, while still very fast, was much slower. After 9 iterations the error is still 2*10^-6.




eqn1-eqn2 and the single dimension approximation
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9