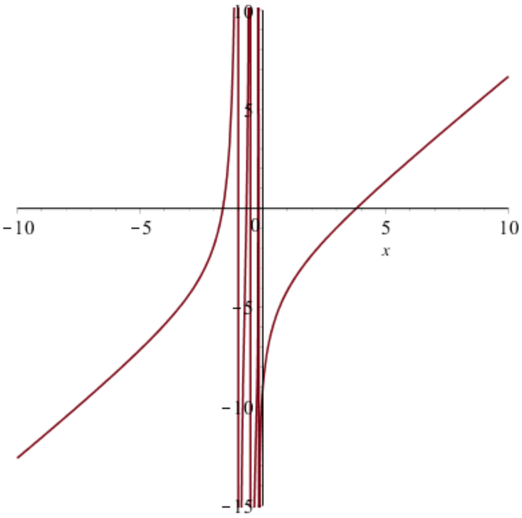
Numerical Analysis Project 1

Margaret Dorsey

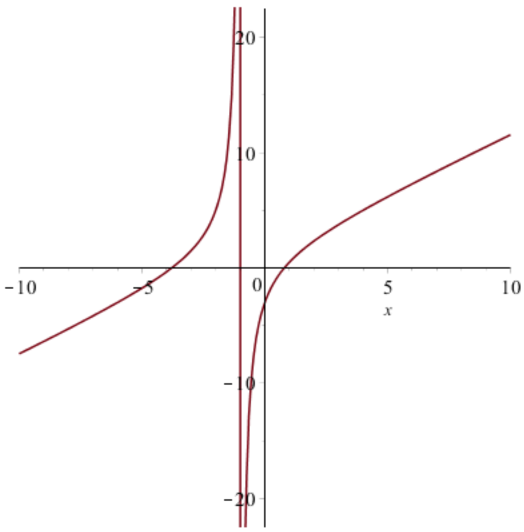
September 21, 2016

Graphical Illustration of the Function



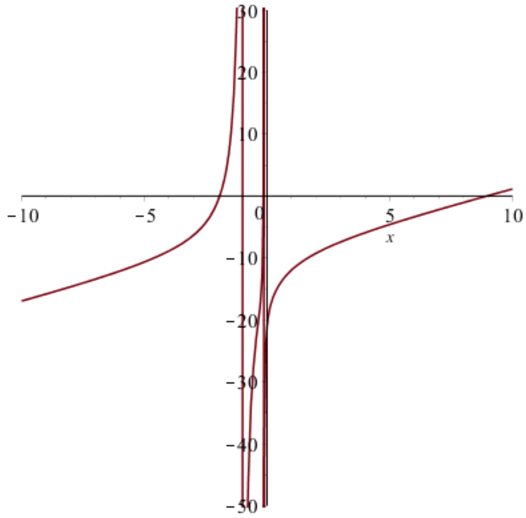
$\xi = 9$, $M = 3$

i	1	2	3
k	1	2	6
η	2	3	1



$\xi = 3$, $M = 1$

i	1
k	1
η	5



$\xi = 21$, $M = 2$

i	1	2
k	1	8
η	9	4

Positive Solutions of X

Claim: There is exactly one positive solution for x.
Proof:

We first note $f(0) < 0$ for all values of k ,η, ξ - when $x = 0$, the rest of the function $f(x)$ disappears, leaving us with $f(0) = -\xi$, a negative value, because ξ represents ligand concentration, a physical quantity, and thus must be positive (because 0 is a trivial case and negative values are non-physical).

Additionally, for $x = \xi$, $f(x) > 0$, because $f(x) = \xi - \xi + \sum_{j=1}^{M} \frac{k_j\eta_j}{1+k_j\eta}$ in this case, where all of the sum terms are non-negative, and at least one is non-zero (otherwise it is modelling a trivial case where there is no amount of any binding molecules).

Thus, by the intermediate value theorem, we know that f has at least one positive root between 0 and ξ. Calculating the derivative with respect to x of f, we get

$$f'(x) = 1 + \sum_{j=1}^{M} = \frac{k_j\eta_j}{(1 + k_jx)^2}$$

which by reasoning analogous to the above argument, is always positive for positive x. Thus $f(x)$ is strictly increasing for positive x, and due to Rolle's theorem, we know that $f(x)$ restricted to positive x has at most one root, which completes the proof.

■

Fixed Point Iteration

Finding $g(x)$

Case Testing

```
Enter the initial guess for the first test: 1

--------------------------
 Fixed Point Iteration
 -----------------------



i: 0 x: 1.000000000 value: 2.500000000
i: 1 x: 2.500000000 value: 2.285714286
i: 2 x: 2.285714286 value: 2.304347826
i: 3 x: 2.304347826 value: 2.302631579
i: 4 x: 2.302631579 value: 2.302788845
i: 5 x: 2.302788845 value: 2.302774427
i: 6 x: 2.302774427 value: 2.302775749
i: 7 x: 2.302775749 value: 2.302775628
i: 8 x: 2.302775628 value: 2.302775639
i: 9 x: 2.302775639 value: 2.302775638

Enter the initial guess for the second test:  1

--------------------------
 Fixed Point Iteration
 -----------------------



i: 0 x: 1.000000000 value: 0.500000000
i: 1 x: 0.500000000 value: 1.333333333
i: 2 x: 1.333333333 value: 0.142857143
i: 3 x: 0.142857143 value: 2.375000000
i: 4 x: 2.375000000 value: -0.518518519
i: 5 x: -0.518518519 value: 8.384615385
i: 6 x: 8.384615385 value: -1.467213115
⋮

i: 54 x: -3.791287852 value: -3.791287845
i: 55 x: -3.791287845 value: -3.791287849
i: 56 x: -3.791287849 value: -3.791287846
i: 57 x: -3.791287846 value: -3.791287848
```

```
i: 58 x: -3.791287848 value: -3.791287847
```

Even though the proposed $g(x)$ converged in both cases, it converged to the negative root in the second case, even though the initial guess was closer to the positive root (which is approximately 0.79). This is because surrounding the positive root, the absolute value of $g'(x)$ is greater than 1, meaning that by the contraction theorem, it cannot converge to that root (because as it approaches it, the sequence will begin to diverge rather than converge).

Choosing α

<u>Claim:</u> If $\alpha = \sum_{j=1}^{M} k_j n_j$, the FPI always converges.

<u>Proof:</u> First, we note that because k_j, η_j are positive for all j and we are only discussing the positive root, $\frac{1}{(k_j * x+1)^2}$ is positive and less than 1, and

$$\sum_{j=1}^{M} k_j \eta_j (1 - \frac{1}{(k_j x + 1)^2} < \sum_{j=1}^{M} k_j \eta_j < 1 + \sum_{j=1}^{M} k_j \eta_j$$

All terms are positive, so we can say that

$$|\sum_{j=1}^{M} k_j \eta_j (1 - \frac{1}{(k_j x + 1)^2}| < 1 + \sum_{j=1}^{M} k_j \eta_j$$

as well. We then perform a few elementary manipulations to get

$$|\sum_{j=1}^{M} k_j \eta_j - \frac{k_j \eta_j}{(k_j x + 1)^2}| < 1 + \sum_{j=1}^{M} k_j \eta_j$$

$$|\sum_{j=1}^{M} k_j \eta_j - \sum_{j=1}^{M} \frac{k_j \eta_j}{(k_j x + 1)^2}| < 1 + \sum_{j=1}^{M} k_j \eta_j$$

with the second step being valid in a finite sum. Substituting in α, we have

$$|\alpha - \sum_{j=1}^{M} \frac{k_j \eta_j}{(k_j x + 1)^2}| < 1 + \alpha$$

and then dividing by $1 + \alpha$ (which can be brought in to the absolute value because it is positive), we have

$$|\frac{1}{1+\alpha}(\alpha - \sum_{j=1}^{M} \frac{k_j \eta_j}{(k_j x + 1)^2})| < 1$$

where the left hand side is equal to $|g'_\alpha(x)|$. Thus, by the contraction theorem, the FPI must always converge because $g'(x)$ is always less than 1 in absolute value for this choice of α. ■

Newton's Method

Test One Output

```
Enter the intial guess: 1.6

--------------------------
 Newton
 -----------------------



i: 0 x: -1.124105012 value: 88.452818065
i: 1 x: -1.260131012 value: 46.182036757
i: 2 x: -1.570535828 value: 24.956847208
i: 3 x: -2.357298970 value: 14.010274589
i: 4 x: -4.536830098 value: 7.290560454
i: 5 x: -8.588461093 value: 1.729329152
i: 6 x: -10.061914568 value: 0.041604994
i: 7 x: -10.099003088 value: 0.000018410
i: 8 x: -10.099019514 value: 0.000000000
```

Test Two Output

Output

```
Enter the intial guess: 1.4999

--------------------------
 Newton
 -----------------------



i: 0 x: -0.999876924 value: -81242.749956933
i: 1 x: -0.999753861 value: -40619.374953876
i: 2 x: -0.999507770 value: -20307.687427792
i: 3 x: -0.999015733 value: -10151.843615770
i: 4 x: -0.998032241 value: -5073.921612311
i: 5 x: -0.996067582 value: -2534.960417718
i: 6 x: -0.992147546 value: -1265.479442812
i: 7 x: -0.984344518 value: -630.738232506
i: 8 x: -0.968885874 value: -313.366310246
i: 9 x: -0.938552168 value: -154.678222468
i: 10 x: -0.880170250 value: -75.331900493
i: 11 x: -0.772155001 value: -35.661641481
```

```
i: 12 x: -0.587979613 value: -15.858623262
i: 13 x: -0.323256320 value: -6.099899665
i: 14 x: -0.056125978 value: -1.650760190
i: 15 x: 0.078909646 value: -0.189707093
i: 16 x: 0.098689910 value: -0.003059278
i: 17 x: 0.099019425 value: -0.000000818
i: 18 x: 0.099019514 value: -0.000000000
```

Test Three Output

```
Enter the intial guess: 1.5

--------------------------
 Newton
 -----------------------


i: 0 x: -1.000000000 value: -inf
i: 1 x: -nan value: -nan
```

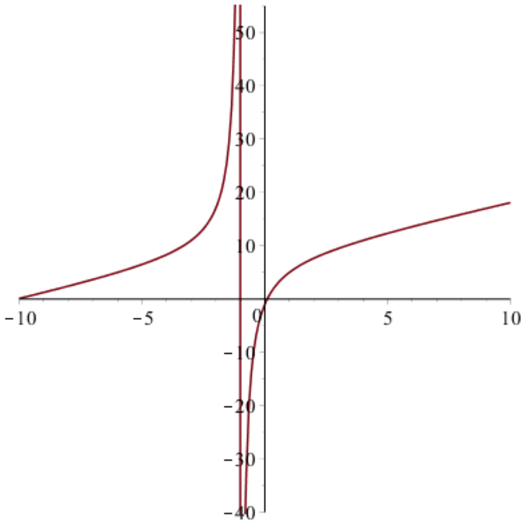
Test Four Output

```
Enter the intial guess: 1.0

--------------------------
 Newton
 -----------------------


i: 0 x: -0.428571429 value: -8.928571429
i: 1 x: -0.146245059 value: -2.859208022
i: 2 x: 0.048003192 value: -0.493952479
i: 3 x: 0.096885702 value: -0.019834462
i: 4 x: 0.099015816 value: -0.000034315
i: 5 x: 0.099019514 value: -0.000000000
```

Analysis



For an initial guess of 1.6, the method successfully converged, however unfortunately it was to the negative root. For 1.499, the method converged (somewhat slowly, for newton's method) to the correct root, however 1.5 leads to division by zero. Examining the graph of f(x), the cause of this behavior becomes fairly apparent - as the positive side of the graph flattens and the slope of the tangent line approaches 0, Newton's method tends to send x_{n+1} through and across the singularity at $x = -1$, into the negative values of x. For x before this jump across the singularity of $f(x)$, such as 1.4999, the method, although displaced into negative x, can self-correct back towards the positive root. As the slope of $f(x)$ gets more and more positive approaching the positive root, Newton's method manages to converge more and more efficiently, as evidenced by the results of initial guess 1.0.

Algorithm Comparison

Output Evaluation

```
--------------------------
 Bisection
 -----------------------



i: 0 a: 2.000000000 b: 4.000000000 value: -2.343589744
i: 1 a: 3.000000000 b: 4.000000000 value: -0.981203008
i: 2 a: 3.500000000 b: 4.000000000 value: -0.364898990
i: 3 a: 3.750000000 b: 4.000000000 value: -0.066547000
i: 4 a: 3.750000000 b: 3.875000000 value: 0.080649333
i: 5 a: 3.750000000 b: 3.812500000 value: 0.007204681
i: 6 a: 3.781250000 b: 3.812500000 value: -0.029631976
i: 7 a: 3.796875000 b: 3.812500000 value: -0.011203954
⋮

i: 26 a: 3.806382805 b: 3.806382835 value: -0.000000012
i: 27 a: 3.806382805 b: 3.806382820 value: 0.000000005
i: 28 a: 3.806382813 b: 3.806382820 value: -0.000000003
i: 29 a: 3.806382813 b: 3.806382816 value: 0.000000001

--------------------------
 Fixed Point Iteration
 -----------------------



i: 0 x: 10.000000000 value: 9.556071251
i: 1 x: 9.556071251 value: 9.142717834
i: 2 x: 9.142717834 value: 8.757913091
i: 3 x: 8.757913091 value: 8.399762914
i: 4 x: 8.399762914 value: 8.066496782
i: 5 x: 8.066496782 value: 7.756459426
i: 6 x: 7.756459426 value: 7.468103079
i: 7 x: 7.468103079 value: 7.199980270
i: 8 x: 7.199980270 value: 6.950737133
i: 9 x: 6.950737133 value: 6.719107200
i: 10 x: 6.719107200 value: 6.503905636
i: 11 x: 6.503905636 value: 6.304023891
i: 12 x: 6.304023891 value: 6.118424739
i: 13 x: 6.118424739 value: 5.946137680
i: 14 x: 5.946137680 value: 5.786254664
```

```
i: 15 x: 5.786254664 value: 5.637926132
i: 16 x: 5.637926132 value: 5.500357326
⋮

i: 234 x: 3.806382850 value: 3.806382848
i: 235 x: 3.806382848 value: 3.806382845
i: 236 x: 3.806382845 value: 3.806382843
i: 237 x: 3.806382843 value: 3.806382841
i: 238 x: 3.806382841 value: 3.806382839
i: 239 x: 3.806382839 value: 3.806382837
i: 240 x: 3.806382837 value: 3.806382835
i: 241 x: 3.806382835 value: 3.806382834
i: 242 x: 3.806382834 value: 3.806382832
i: 243 x: 3.806382832 value: 3.806382831
i: 244 x: 3.806382831 value: 3.806382830
i: 245 x: 3.806382830 value: 3.806382829
i: 246 x: 3.806382829 value: 3.806382828
```

```
Enter the initial guess: 1
--------------------------
 Newton
 -----------------------


i: 0 x: 3.318702187 value: -0.585022456
i: 1 x: 3.796734040 value: -0.011370114
i: 2 x: 3.806379686 value: -0.000003686
i: 3 x: 3.806382815 value: -0.000000000
```

Asymptotic Error Constant Calculation

First, we verify that the methods are converging at the expected rates (linearly, linearly, and quadratically respectively), and estimate their asymptotic error constants.

Bisection

Simply by examining the data, it can be seen that the bisection method is converging linearly, with error constant $\frac{1}{2}$, as expected. After all, the distance between the brackets is halved at each interval simply by the construction of the algorithm.

FPI

Testing a few iterations ($i = 2, 27, 69, 124$), we find the data set $\{.92811, .920973, .9214266, .9230767\}$ for $\frac{x_{k+1}-r}{x_k-r}$, with r being the root. This, along with the consistency (although slow) of the convergence rate by inspection of the entire output implies linear convergence with an error constant $\approx .92$.

Newton's Method

Using all of the small number of iterations we have, we obtain a data set of $\{.0619, .0406, .0338\}$ for $|\frac{x_{k+1}-r}{(x_k-r)^2}|$. Because the data is so small and the variation in these values so big, it is hard to decide what the error constant will actually be - although it seems apparent that it is probably less than .0338. Additionally, it is fairly obvious with these values that Newton's method did indeed converge quadratically.

Test Case Results

Test Case 1

$\xi = 12$, $M = 7$

i	1	2	3	4	5	6	7
k	2	4	6	8	10	12	14
η	1	3	5	7	9	11	13

```
Enter the brackets separated by a space:  0 12

--------------------------
 Bisection
 -----------------------



i: 0 a: 0.000000000 b: 6.000000000 value: 42.073917554
i: 1 a: 0.000000000 b: 3.000000000 value: 38.193270272
i: 2 a: 0.000000000 b: 1.500000000 value: 35.050610574
i: 3 a: 0.000000000 b: 0.750000000 value: 31.401650779
i: 4 a: 0.000000000 b: 0.375000000 value: 26.340373048
i: 5 a: 0.000000000 b: 0.187500000 value: 19.491743318
i: 6 a: 0.000000000 b: 0.093750000 value: 11.426573055
i: 7 a: 0.000000000 b: 0.046875000 value: 3.596209203
i: 8 a: 0.023437500 b: 0.046875000 value: -2.617682398
i: 9 a: 0.023437500 b: 0.035156250 value: 0.771000643
i: 10 a: 0.029296875 b: 0.035156250 value: -0.842334991
i: 11 a: 0.032226562 b: 0.035156250 value: -0.016903399
i: 12 a: 0.032226562 b: 0.033691406 value: 0.381569812
```

```
i: 13 a: 0.032226562 b: 0.032958984 value: 0.183484060
i: 14 a: 0.032226562 b: 0.032592773 value: 0.083580670
i: 15 a: 0.032226562 b: 0.032409668 value: 0.033411552
i: 16 a: 0.032226562 b: 0.032318115 value: 0.008272347
i: 17 a: 0.032272339 b: 0.032318115 value: -0.004310953
i: 18 a: 0.032272339 b: 0.032295227 value: 0.001981840
⋮

i: 45 a: 0.032288018 b: 0.032288018 value: -0.000000000


Enter the initial guess: 1

--------------------------
 Fixed Point Iteration
 -----------------------



i: 0 x: 1.000000000 value: 0.934567127
i: 1 x: 0.934567127 value: 0.869860516
i: 2 x: 0.869860516 value: 0.805948615
i: 3 x: 0.805948615 value: 0.742911886
i: 4 x: 0.742911886 value: 0.680845556
i: 5 x: 0.680845556 value: 0.619863138
i: 6 x: 0.619863138 value: 0.560100946
i: 7 x: 0.560100946 value: 0.501723926
i: 8 x: 0.501723926 value: 0.444933154
i: 9 x: 0.444933154 value: 0.389975471
i: 10 x: 0.389975471 value: 0.337155624
i: 11 x: 0.337155624 value: 0.286851060
i: 12 x: 0.286851060 value: 0.239528568
⋮

i: 45 x: 0.032288018 value: 0.032288018
i: 46 x: 0.032288018 value: 0.032288018

Enter the intial guess: 1

--------------------------
 Newton
 -----------------------
```

```
i: 0 x: -5.188426064 value: 32.946741439
i: 1 x: -32.060273608 value: 5.118528517
i: 2 x: -37.150270786 value: 0.003926264
i: 3 x: -37.154180750 value: 0.000000002
i: 4 x: -37.154180752 value: 0.000000000
```

Test Case 2

$\xi = 1876$, $M = 4$

i	1	2	3	4
k	31	73	5	78
η	62	81	102	56

```
Enter the brackets separated by a space: 0 1876

--------------------------
 Bisection
 -----------------------



i: 0 a: 938.000000000 b: 1876.000000000 value: -637.025824194
i: 1 a: 1407.000000000 b: 1876.000000000 value: -168.017217182
i: 2 a: 1407.000000000 b: 1641.500000000 value: 66.485242158
i: 3 a: 1524.250000000 b: 1641.500000000 value: -50.765892933
i: 4 a: 1524.250000000 b: 1582.875000000 value: 7.859695630
i: 5 a: 1553.562500000 b: 1582.875000000 value: -21.453093099
```

$\vdots$

```
i: 39 a: 1575.015380733 b: 1575.015380735 value: 0.000000002
i: 40 a: 1575.015380733 b: 1575.015380734 value: 0.000000001
i: 41 a: 1575.015380733 b: 1575.015380733 value: 0.000000000
i: 42 a: 1575.015380733 b: 1575.015380733 value: 0.000000000
i: 43 a: 1575.015380733 b: 1575.015380733 value: 0.000000000
i: 44 a: 1575.015380733 b: 1575.015380733 value: -0.000000000
i: 45 a: 1575.015380733 b: 1575.015380733 value: 0.000000000

Enter the initial guess: 1000

--------------------------
 Fixed Point Iteration
 -----------------------
```

```
i: 0 x: 1000.000000000 value: 1000.045227641
i: 1 x: 1000.045227641 value: 1000.090451724
i: 2 x: 1000.090451724 value: 1000.135672250
i: 3 x: 1000.135672250 value: 1000.180889219
i: 4 x: 1000.180889219 value: 1000.226102632
i: 5 x: 1000.226102632 value: 1000.271312489
i: 6 x: 1000.271312489 value: 1000.316518789
i: 7 x: 1000.316518789 value: 1000.361721534
i: 8 x: 1000.361721534 value: 1000.406920723
i: 9 x: 1000.406920723 value: 1000.452116357
i: 10 x: 1000.452116357 value: 1000.497308437
i: 11 x: 1000.497308437 value: 1000.542496961
i: 12 x: 1000.542496961 value: 1000.587681932
⋮

i: 282685 x: 1575.015380606 value: 1575.015380606
i: 282686 x: 1575.015380606 value: 1575.015380606


Enter the intial guess: 1000

--------------------------
 Newton
 -----------------------



i: 0 x: 1575.010297079 value: -0.005083704
i: 1 x: 1575.015380733 value: -0.000000000
```

Test Case 3

$\xi = 21$, $M = 2$

i	1	2
k	1	8
η	9	4

```
Enter the brackets separated by a space: 0 21

--------------------------
 Bisection
 -----------------------
```

```
i: 0 a: 0.000000000 b: 10.500000000 value: 1.670332481
i: 1 a: 5.250000000 b: 10.500000000 value: -4.283023256
i: 2 a: 7.875000000 b: 10.500000000 value: -1.201584507
i: 3 a: 7.875000000 b: 9.187500000 value: 0.250373142
i: 4 a: 8.531250000 b: 9.187500000 value: -0.470774028
i: 5 a: 8.859375000 b: 9.187500000 value: -0.109113941
i: 6 a: 8.859375000 b: 9.023437500 value: 0.070887801
i: 7 a: 8.941406250 b: 9.023437500 value: -0.019046911
i: 8 a: 8.941406250 b: 8.982421875 value: 0.025936779
i: 9 a: 8.941406250 b: 8.961914062 value: 0.003449043
i: 10 a: 8.951660156 b: 8.961914062 value: -0.007797904
i: 11 a: 8.956787109 b: 8.961914062 value: -0.002174173
i: 12 a: 8.956787109 b: 8.959350586 value: 0.000637499
i: 13 a: 8.958068848 b: 8.959350586 value: -0.000768321
⋮

i: 39 a: 8.958769351 b: 8.958769351 value: -0.000000000

Enter the initial guess: 1

--------------------------
 Fixed Point Iteration
 -----------------------



i: 0 x: 1.000000000 value: 1.284391534
i: 1 x: 1.284391534 value: 1.546537998
i: 2 x: 1.546537998 value: 1.791461752
i: 3 x: 1.791461752 value: 2.022260632
i: 4 x: 2.022260632 value: 2.241034383
i: 5 x: 2.241034383 value: 2.449300595
i: 6 x: 2.449300595 value: 2.648208965
i: 7 x: 2.648208965 value: 2.838662573
i: 8 x: 2.838662573 value: 3.021391482
i: 9 x: 3.021391482 value: 3.196999863
i: 10 x: 3.196999863 value: 3.365997501
i: 11 x: 3.365997501 value: 3.528821612
i: 12 x: 3.528821612 value: 3.685852405
i: 13 x: 3.685852405 value: 3.837424475
i: 14 x: 3.837424475 value: 3.983835310
i: 15 x: 3.983835310 value: 4.125351782
i: 16 x: 4.125351782 value: 4.262215177
```

```
i: 17 x: 4.262215177 value: 4.394645145
i: 18 x: 4.394645145 value: 4.522842861
⋮

i: 888 x: 8.958769350 value: 8.958769350
i: 889 x: 8.958769350 value: 8.958769350
i: 890 x: 8.958769350 value: 8.958769350
i: 891 x: 8.958769350 value: 8.958769350
i: 892 x: 8.958769350 value: 8.958769350

Enter the intial guess: 1

--------------------------
 Newton
 -----------------------


i: 0 x: 4.276883997 value: -5.542255721
i: 1 x: 8.385252241 value: -0.632451741
i: 2 x: 8.955501144 value: -0.003584694
i: 3 x: 8.958769256 value: -0.000000105
i: 4 x: 8.958769351 value: 0.000000000
```

Conclusions and Preferred Algorithm

Given that we need no guesswork in choosing brackets, and the function is guaranteed to be continuous between 0 and ξ, the bisection method is a very tempting, stable method for use, and does not converge horribly slowly even for large ξ. Newton's method is much, much faster, as anticipated, but as noted in the previous section can fail very badly due to the singularity in the functions, despite a reasonable guess. Although we determined α such that FPI always converges, the rate of convergence depends on the value of g at root r, and so as seen by the second test, can be very, very slow to work even though we have improved its stability.

With all of these things in mind, my recommendation would be to try Newton's method to see if educated guesses yield a quick and easy solution, and if a good value proves hard to find, to switch to bisection method where no guess work is necessary.