HOMEWORK #1

Ann Guilinger   Raphael Leu   Brian Roberts

September 22, 2014

1 Break Math

See Ruby file hw1-1.rb

2 Matrix inversion

See Ruby file hw1-2.rb

We used the Ruby matrix library. This inverter uses Gauss-Jordan with pivoting.

3 Binary and Floating point

Decimal	Binary	Floating point
10.5	1010.1	0 10000000010 10101000000000000000000000000000000000000000000000000
$\frac{1}{3}$	$.\overline{01}$	0 01111111101 10101010101010101010101010101010101010101010101010101
$\frac{22}{7}$	$11.\overline{001}$	0 10000000000 11001001001001001001001001001001001001001000100000111

4 Head-to-head

1. Bisection for $f(x) = x^4 - 2$

```
xmid: 1.0, y value: -1.0
xmid: 1.0, y value: 3.0625
xmid: 1.0, y value: 0.44140625
xmid: 1.125, y value: -0.398193359375
xmid: 1.1875, y value: -0.0114593505859375
xmid: 1.1875, y value: 0.2062692642211914
xmid: 1.1875, y value: 0.09528452157974243
xmid: 1.1875, y value: 0.041389349848032
xmid: 1.1875, y value: 0.014835044974461198
xmid: 1.1875, y value: 0.0016554650064790621
xmid: 1.1884765625, y value: -0.004910025046228839
xmid: 1.18896484375, y value: -0.0016293022444529015
xmid: 1.18896484375, y value: 1.2575617237331471e-05
xmid: 1.1890869140625, y value: -0.0008084897285944859
xmid: 1.18914794921875, y value: -0.0003979866266971267
xmid: 1.189178466796875, y value: -0.00019271442486945567
xmid: 1.1891937255859375, y value: -9.007137940519883e-05
xmid: 1.1892013549804688, y value: -3.8748374987518375e-05
xmid: 1.1892051696777344, y value: -1.3086502351766782e-05
xmid: 1.1892070770263672, y value: -2.554734264137437e-07
xmid: 1.1892070770263672, y value: 6.160064188076575e-06
xmid: 1.1892070770263672, y value: 2.9522934514858434e-06
xmid: 1.1892070770263672, y value: 1.348409530255168e-06
xmid: 1.1892070770263672, y value: 5.464679313504917e-07
xmid: 1.1892070770263672, y value: 1.454972222703077e-07
xmid: 1.1892071068286896, y value: -5.498810962123457e-08
xmid: 1.1892071068286896, y value: 4.525455432613512e-08
xmid: 1.1892071142792702, y value: -4.866778091638935e-09
xmid: 1.1892071142792702, y value: 2.019388789520349e-08
xmid: 1.1892071142792702, y value: 7.663555123826882e-09
xmid: 1.1892071142792702, y value: 1.3983885160939735e-09
xmid: 1.1892071147449315, y value: -1.7341947877724806e-09
xmid: 1.189207114977762, y value: -1.67903246861556e-10
Final y value: -1.67903246861556e-10
Final x value: 1.1892071152105927
Number of iterations: 33
```

The bisection produced results within 8 digits of accuracy in 33 iterations. The convergence in the y-axis seemed to waiver between positive and negative y-values at random.

2. Secant for $f(x) = x^4 - 2$

```
xold: 2.0, xnew: 0.25, y value: -1.99609375
xold: 0.25, xnew: 0.4683760683760684, y value: -1.9518741085724152
xold: 0.4683760683760684, xnew: 10.107590228289883, y value: 10435.356258593585
xold: 10.107590228289883, xnew: 0.47017869145727964, y value: -1.9511289385357296
xold: 0.47017869145727964, xnew: 0.4719802894982221, y value: -1.9503755793809
xold: 0.4719802894982221, xnew: 5.1361471750910574, y value: 693.904811749434
xold: 5.1361471750910574, xnew: 0.48505323536033096, y value: -1.9446449021369268
xold: 0.48505323536033096, xnew: 0.4980513429722569, y value: -1.9384686474022659
xold: 0.4980513429722569, xnew: 4.577614956156458, y value: 437.0935297596889
xold: 4.577614956156458, xnew: 0.5160639392931019, y value: -1.9290725979414054
xold: 0.5160639392931019, xnew: 0.5339104618404529, y value: -1.9187405790343723
xold: 0.5339104618404529, xnew: 3.848155946132483, y value: 217.28587285784266
xold: 3.848155946132483, xnew: 0.5629206966155693, y value: -1.899587248960527
xold: 0.5629206966155693, xnew: 0.5913924345984497, y value: -1.8776784267907698
xold: 0.5913924345984497, xnew: 3.031540415601771, y value: 82.4604614727347
xold: 3.031540415601771, xnew: 0.6457191393083876, y value: -1.8261500232815193
xold: 0.6457191393083876, xnew: 0.6974102383087862, y value: -1.7634334834441627
xold: 0.6974102383087862, xnew: 2.150835682533229, y value: 19.400746929765756
xold: 2.150835682533229, xnew: 0.8185119792689675, y value: -1.5511511058682408
xold: 0.8185119792689675, xnew: 0.9171491233681796, y value: -1.292445624208415
xold: 0.9171491233681796, xnew: 1.4099223899318556, y value: 1.951671449155835
xold: 1.4099223899318556, xnew: 1.113468379946871, y value: -0.46286661840229004
xold: 1.113468379946871, xnew: 1.1702985732485325, y value: -0.12419926249858793
xold: 1.1702985732485325, xnew: 1.191139873635705, y value: 0.01303373004931263
xold: 1.191139873635705, xnew: 1.1891604670944667, y value: -0.00031379000692033365
xold: 1.1891604670944667, xnew: 1.1892070014311567, y value: -7.640152617227614e-
07
xold: 1.1892070014311567, xnew: 1.1892071150094037, y value: 4.495515071312184e-
11
Final y value: 4.495515071312184e-11
Final x value: 1.1892071150094037
Number of iterations: 27
```

The secant method produced results in 27 iterations. The convergence has a pattern of very large positive values followed by two small negative values. This makes sense considering the root was bracketed, so each large guess will lead to a btter, smaller guess in the next iteration.

3. False position for $f(x) = x^4 - 2$

lower bound: 0.25, upper bound: 2.0, y value: -1.99609375
lower bound: 0.4683760683760684, upper bound: 2.0, y value: -1.9518741085724152
lower bound: 0.6557858407864514, upper bound: 2.0, y value: -1.815052637648054
lower bound: 0.8100578189546664, upper bound: 2.0, y value: -1.5694098673868484
lower bound: 0.9300050113311887, upper bound: 2.0, y value: -1.251931866272392
lower bound: 1.0178339391556377, upper bound: 2.0, y value: -0.9267331576136155
lower bound: 1.0788121749997586, upper bound: 2.0, y value: -0.6454864426247353
lower bound: 1.1194126872789572, upper bound: 2.0, y value: -0.4297785731256021
lower bound: 1.1456402247879947, upper bound: 2.0, y value: -0.2773659977159153
lower bound: 1.1622378487123188, upper bound: 2.0, y value: -0.1753479592382161
lower bound: 1.1726009018082801, upper bound: 2.0, y value: -0.10939459391485573
lower bound: 1.1790159884195255, upper bound: 2.0, y value: -0.06768120516830023
lower bound: 1.1829658353429302, upper bound: 2.0, y value: -0.04165678048632615
lower bound: 1.1853896955311558, upper bound: 2.0, y value: -0.02555704581654239
lower bound: 1.1868740595964067, upper bound: 2.0, y value: -0.015648737161840565
lower bound: 1.1877819301031147, upper bound: 2.0, y value: -0.009570241854789385
lower bound: 1.188336773915848, upper bound: 2.0, y value: -0.005848509189096074
lower bound: 1.188675705172536, upper bound: 2.0, y value: -0.003572489470656581 3
lower bound: 1.1888826843203741, upper bound: 2.0, y value: -0.002181607824133902
lower bound: 1.1890090603331798, upper bound: 2.0, y value: -0.0013320148893902761
lower bound: 1.18908621385012, upper bound: 2.0, y value: -0.0008131987448745104
lower bound: 1.1891333135480973, upper bound: 2.0, y value: -0.0004964288145836715
lower bound: 1.1891620652134391, upper bound: 2.0, y value: -0.00030304043021711635
lower bound: 1.189179616024724, upper bound: 2.0, y value: -0.00018498391974808293
lower bound: 1.1891903293640844, upper bound: 2.0, y value: -0.00011291747608788 15
lower bound: 1.1891968689243104, upper bound: 2.0, y value: -6.892623404963771e-05
lower bound: 1.1892008607336848, upper bound: 2.0, y value: -4.207320739291198e-05
lower bound: 1.1892032973635294, upper bound: 2.0, y value: -2.568178922102149e-05
lower bound: 1.189204784697231, upper bound: 2.0, y value: -1.567631818732984e-05
lower bound: 1.1892056925736272, upper bound: 2.0, y value: -9.568907039625 43e-06
lower bound: 1.1892062467457738, upper bound: 2.0, y value: -5.840906839083004e-06
lower bound: 1.189206585014974, upper bound: 2.0, y value: -3.56531598955101e-06
lower bound: 1.1892067914959734, upper bound: 2.0, y value: -2.1762844273354176e-06
lower bound: 1.1892069175328561, upper bound: 2.0, y value: -1.328413281598273e-06
lower bound: 1.1892069944662989, upper bound: 2.0, y value: -8.10869039469253e-07
lower bound: 1.1892070414267923, upper bound: 2.0, y value: -4.949578318313996e-07
lower bound: 1.1892070700916717, upper bound: 2.0, y value: -3.0212430623954845e-07
lower bound: 1.189207087588832, upper bound: 2.0, y value: -1.8441792137835478e-07
lower bound: 1.1892070982691707, upper bound: 2.0, y value: -1.1256945775528493e-07
lower bound: 1.1892071047884933, upper bound: 2.0, y value: -6.871285895826418e-08
lower bound: 1.1892071087679146, upper bound: 2.0, y value: -4.19426111619714e-08
lower bound: 1.1892071111969695, upper bound: 2.0, y value: -2.5601942343911333e-08
lower bound: 1.1892071126796748, upper bound: 2.0, y value: -1.5627529936779183e-08
lower bound: 1.1892071135847242, upper bound: 2.0, y value: -9.539107947986736e-09
lower bound: 1.18920711413717, upper bound: 2.0, y value: -5.8227098698182544e-09
lower bound: 1.1892071144743852, upper bound: 2.0, y value: -3.554205818545597e-09
lower bound: 1.1892071146802228, upper bound: 2.0, y value: -2.1695012453193385e-09
lower bound: 1.1892071148058667, upper bound: 2.0, y value: -1.324272913549862e-09

```
lower bound: 1.1892071148825603, upper bound: 2.0, y value: -8.083422819993302e-10
Final y value: -8.083422819993302e-10
Final x value: 1.1892071148825603
Number of iterations: 49
```

This method acheives 8 digits of accuracy in 49 iterations. This method converges with each step getting closer at a constant rate (although it is consistently negative which is logical considering the method keeps the intial guess of 2 throughout the algorithm).

4. FPI for $g(x) = \frac{x}{2} + \frac{1}{x^3}$

old x: 1.0, new x: 1.5, difference: 0.5
old x: 1.5, new x: 1.0462962962962963, difference: -0.4537037037037037
old x: 1.0462962962962963, new x: 1.3961917541713085, difference: 0.3498954578750122
old x: 1.3961917541713085, new x: 1.065517569872507, difference: -0.33067418429880147
old x: 1.065517569872507, new x: 1.3594020918672058, difference: 0.29388452199469883
old x: 1.3594020918672058, new x: 1.077768066601146, difference: -0.28163402526605985
old x: 1.077768066601146, new x: 1.3376582996965998, difference: 0.2598902330954538
old x: 1.3376582996965998, new x: 1.0866253005767548, difference: -0.251032999119845
old x: 1.0866253005767548, new x: 1.3227129378725515, difference: 0.23608763729579674
old x: 1.3227129378725515, new x: 1.0934753027729127, difference: -0.2292376350996388
old x: 1.0934753027729127, new x: 1.3115820042343793, difference: 0.2181067014614666
old x: 1.3115820042343793, new x: 1.0990051879662324, difference: -0.21257681626814695
old x: 1.0990051879662324, new x: 1.3028594974358345, difference: 0.20385430946960215
old x: 1.3028594974358345, new x: 1.1036054808456455, difference: -0.19925401659018904
old x: 1.1036054808456455, new x: 1.2957779322327534, difference: 0.19217245138710792
old x: 1.2957779322327534, new x: 1.1075188544259187, difference: -0.1882590778068347
old x: 1.1075188544259187, new x: 1.2898760327884606, difference: 0.18235717836254195
old x: 1.2898760327884606, new x: 1.1109060030282172, difference: -0.17897002976024345
old x: 1.1109060030282172, new x: 1.284856864848635, difference: 0.17395086182041775
old x: 1.284856864848635, new x: 1.113878554204667, difference: -0.17097831064396796
old x: 1.113878554204667, new x: 1.2805191411018542, difference: 0.1666405868971872
old x: 1.2805191411018542, new x: 1.116517013665047, difference: -0.1640021274368071
old x: 1.116517013665047, new x: 1.2767207734536554, difference: 0.16020375978860835
old x: 1.2767207734536554, new x: 1.1188812244487978, difference: -0.15783954900485764
old x: 1.1188812244487978, new x: 1.2733581347570162, difference: 0.15447691030821842
old x: 1.2733581347570162, new x: 1.121016793695299, difference: -0.15234134106171715
old x: 1.121016793695299, new x: 1.2703535868879654, difference: 0.14933679319266635
old x: 1.2703535868879654, new x: 1.1229592151180952, difference: -0.14739437176987025
old x: 1.1229592151180952, new x: 1.2676476340732572, difference: 0.14468841895516205
old x: 1.2676476340732572, new x: 1.1247366095948568, difference: -0.14291102447840043
old x: 1.1247366095948568, new x: 1.2651937974661023, difference: 0.1404571878712455
old x: 1.2651937974661023, new x: 1.126371602994749, difference: -0.13882219447135324
old x: 1.126371602994749, new x: 1.2629551593016917, difference: 0.13658355630694263
old x: 1.2629551593016917, new x: 1.1278826466348442, difference: -0.1350725126668475
old x: 1.1278826466348442, new x: 1.2609019684937854, difference: 0.13301932185894128
old x: 1.2609019684937854, new x: 1.1292849669246898, difference: -0.13161700156909562
old x: 1.1292849669246898, new x: 1.2590099420143555, difference: 0.12972497508966563
old x: 1.2590099420143555, new x: 1.1305912619299407, difference: -0.1284186800844147
old x: 1.1305912619299407, new x: 1.2572590346696955, difference: 0.1266677727397547
old x: 1.2572590346696955, new x: 1.13181222128628, difference: -0.12544681338341546
old x: 1.13181222128628, new x: 1.255632531659297, difference: 0.12382031037301688
old x: 1.255632531659297, new x: 1.1329569203361833, difference: -0.12267561132311355
old x: 1.1329569203361833, new x: 1.2541163682283913, difference: 0.121159447892208
...

The error in the y-axis switches off between being positive and negative for this method, although the backward error got consistently smaller, the method never converged to be within 8 digits of accuracy. This makes sense, as linear convergence is not predicted since $g'(x) = \frac{1}{2} - 3\frac{1}{x^4}$, so $g'(\sqrt[4]{2}) \approx -2.02269$, therefore $|g'(r)| \geq 1$, so linear convergence will not exist in this case.

5. FPI for $g(x) = \frac{2x}{3} + \frac{2}{3x^3}$

old x: 1.0, new x: 1.3333333333333333, difference: 0.33333333333333326
old x: 1.3333333333333333, new x: 1.1701388888888888, difference: -0.16319444444444442
old x: 1.1701388888888888, new x: 1.1961914292767475, difference: 0.026052540387858647
old x: 1.1961914292767475, new x: 1.1869602531658547, difference: -0.009231176110892747
old x: 1.1869602531658547, new x: 1.1899645861030304, difference: 0.0030043329371756133
old x: 1.1899645861030304, new x: 1.1889555885621519, difference: -0.0010089975408784646
old x: 1.1889555885621519, new x: 1.1892910635866454, difference: 0.0003354750244934923
old x: 1.1892910635866454, new x: 1.1891791439922261, difference: -0.00011191959441925192
old x: 1.1891791439922261, new x: 1.1892164399887344, difference: 3.7295996508257545e-05
old x: 1.1892164399887344, new x: 1.189204006820289, difference: -1.2433168445458165e-05
old x: 1.189204006820289, new x: 1.1892081510797794, difference: 4.144259490468372e-06
old x: 1.1892081510797794, new x: 1.1892067696455069, difference: -1.3814342725293471e-06
old x: 1.1892067696455069, new x: 1.1892072301219931, difference: 4.6047648627478566e-07
old x: 1.1892072301219931, new x: 1.1892070766296527, difference: -1.5349234039341297e-07
old x: 1.1892070766296527, new x: 1.1892071277937464, difference: 5.1164093628486285e-08
old x: 1.1892071277937464, new x: 1.1892071107390463, difference: -1.7054700096608144e-08
old x: 1.1892071107390463, new x: 1.1892071164239462, difference: 5.684899884172978e-09
old x: 1.1892071164239462, new x: 1.1892071145289793, difference: -1.8949668501022643e-09
old x: 1.1892071145289793, new x: 1.189207115160635, difference: 6.316556167007548e-10
Final y value: 6.316556167007548e-10
Final x value: 1.189207115160635
Number of iterations: 19

This method converges to 8 digits of accuracy within 19 iterations. The method produced y-values switching off from positive to negative. The quick convergence implies that the equation was well built with α, since for $\frac{3}{2}g_\alpha(x) = x + \frac{1}{x^3}$, so $f(x) = \frac{1}{x^3}$, so $f'(r) = \frac{-3}{2}$, therefore $\alpha = \frac{2}{3}$.

6. FPI for $g(x) = x - \frac{2}{5}(x^4 - 2)$

old x: 1.0, new x: 1.15, difference: 0.1499999999999999
old x: 1.15, new x: 1.1876490625, difference: 0.03764906250000011
old x: 1.1876490625, new x: 1.1892181683767857, difference: 0.0015691058767857147
old x: 1.1892181683767857, new x: 1.1892070145301257, difference: -1.1153846660061362e-05
old x: 1.1892070145301257, new x: 1.1892071159145672, difference: 1.0138444150875614e-07
old x: 1.1892071159145672, new x: 1.1892071149944454, difference: -9.201217565646402e-10
old x: 1.1892071149944454, new x: 1.189207115002796, difference: 8.350653502020577e-12
Final y value: 8.350653502020577e-12
Final x value: 1.189207115002796
Number of iterations: 7

This method yielded the quickest convergence, with the method getting to 8 digits of accuracy within 7 iterations. The second guess got within nearly 0.15 of the origin, and quickly got closer. This quick convergence follows logically from picking our β to be close to $-g'(r)$ since $-g'(r) = 8/5 = 1.6$ while $-g'(r) \approx 1.6908685$.

5 More FPI

1. $g(x) = \frac{2x^3-1}{6}$

   First test:

   old x: 1.0, new x: 0.16666666666666666, difference: -0.8333333333333334
   old x: 0.16666666666666666, new x: -0.16512345679012344, difference: -0.3317901234567901
   old x: -0.16512345679012344, new x: -0.16816740529326554, difference: -0.003043948503142102
   old x: -0.16816740529326554, new x: -0.16825194022334902, difference: -8.453493008347968e-05
   old x: -0.16825194022334902, new x: -0.1682543320964722, difference: -2.3918731231753476e-06
   old x: -0.1682543320964722, new x: -0.16825439980829027, difference: -6.77118180691938e-08
   old x: -0.16825439980829027, new x: -0.16825440172518014, difference: -1.916889869058025e-09
   old x: -0.16825440172518014, new x: -0.1682544017794464, difference: -5.426625815374564e-11
   Final y value: -5.426625815374564e-11
   Final x value: -0.1682544017794464
   Number of iterations: 8

   Second test:

   old x: -1.0, new x: -0.5, difference: 0.5
   old x: -0.5, new x: -0.20833333333333334, difference: 0.29166666666666663
   old x: -0.20833333333333334, new x: -0.16968074845679013, difference: 0.03865258487654322
   old x: -0.16968074845679013, new x: -0.1682951242795515, difference: 0.0013856241772386146
   old x: -0.1682951242795515, new x: -0.16825555489542218, difference: 3.9569384129328755e-05
   old x: -0.16825555489542218, new x: -0.16825443442539353, difference: 1.120470028648235e-06
   old x: -0.16825443442539353, new x: -0.1682544027051747, difference: 3.1720218840458614e-08
   old x: -0.1682544027051747, new x: -0.16825440180718962, difference: 8.979850751877905e-10
   Final y value: 8.979850751877905e-10
   Final x value: -0.16825440180718962
   Number of iterations: 8

   Notice that this function can only find the root at -0.16825440180718962, even though our intial guess were near the other two roots. This implies that the function we built for FPI only can capture the root for which the function is going from positive to negative. Near this root, though, the function is very stable, since it found the root within 8 iterations.

2. $g(x) = x - \frac{2x^3-6x-1}{6x^2-6}$

   old x: -2.0, new x: -1.7222222222222223, difference: 0.2777777777777777
   old x: -1.7222222222222223, new x: -1.6473632187917904, difference: 0.07485900343043195
   old x: -1.6473632187917904, new x: -1.6418134196310956, difference: 0.005549799160694757
   old x: -1.6418134196310956, new x: -1.6417835283181434, difference: 2.9891312952168292e-05
   old x: -1.6417835283181434, new x: -1.6417835274529258, difference: 8.652176752832474e-10
   Final y value: 8.652176752832474e-10
   Final x value: -1.6417835274529258

Number of iterations: 5

old x: -0.5, new x: -0.1111111111111111, difference: 0.3888888888888889
old x: -0.1111111111111111, new x: -0.16782407407407407, difference: -0.056712962962962965
old x: -0.16782407407407407, new x: -0.1682543697750313, difference: -0.0004302957009572339
old x: -0.1682543697750313, new x: -0.16825440178102724, difference: -3.200599593689013e-08
Final y value: -3.200599593689013e-08
Final x value: -0.16825440178102724
Number of iterations: 4

old x: 2.0, new x: 1.8333333333333333, difference: -0.16666666666666674
old x: 1.8333333333333333, new x: 1.8104575163398693, difference: -0.02287581699346397
old x: 1.8104575163398693, new x: 1.8100380691578124, difference: -0.0004194471820568868
old x: 1.8100380691578124, new x: 1.8100379292339688, difference: -1.399238436228245e-07
old x: 1.8100379292339688, new x: 1.810037929233953, difference: -1.5765166949677223e-14
Final y value: -1.5765166949677223e-14
Final x value: 1.810037929233953
Number of iterations: 5

Yay it found all the roots!
This equation found all the roots with an initial guess near the actual value of the root. Being near the root is important, as our function was built assuming that there is a neighborhood of r such that this function converges. This yeilds very quick results for all three roots, though, so as long as our initial guesses can be close enough, this method is very effective.

3. $g(x) = x^4 - 3x^2 + \frac{x}{2}$

old x: -1.0, new x: -2.5, difference: -1.5
old x: -2.5, new x: 19.0625, difference: 21.5625
old x: 19.0625, new x: 130963.62403869629, difference: 130944.56153869629
old x: 130963.62403869629, new x: 2.941729512841314e+20, difference: 2.9417295128413127e+20
old x: 2.941729512841314e+20, new x: 7.488777894424145e+81, difference: 7.488777894424145e+81
old x: 7.488777894424145e+81, new x: Infinity, difference: Infinity
old x: Infinity, new x: NaN, difference: NaN
Final y value: NaN Final x value: NaN

old x: 2.0, new x: 5.0, difference: 3.0
old x: 5.0, new x: 552.5, difference: 547.5
old x: 552.5, new x: 93180462671.5625, difference: 93180462119.0625
old x: 93180462671.5625, new x: 7.538751886004188e+43, difference: 7.538751886004188e+43
old x: 7.538751886004188e+43, new x: 3.2299648823842375e+175, difference: 3.2299648823842375e+175
old x: 3.2299648823842375e+175, new x: NaN, difference: NaN
Final y value: NaN Final x value: NaN Number of iterations: 6

old x: -0.5, new x: -0.9375, difference: -0.4375
old x: -0.9375, new x: -2.3329925537109375, difference: -1.3954925537109375
old x: -2.3329925537109375, new x: 12.129603404604936, difference: 14.462595958315873
old x: 12.129603404604936, new x: 21211.11911870733, difference: 21198.989515302725
old x: 21211.11911870733, new x: 2.0242042331057843e+17, difference: 2.0242042331055722e+17

old x: 2.0242042331057843e+17, new x: 1.6788709519617318e+69, difference: 1.6788709519617318e+69
old x: 1.6788709519617318e+69, new x: 7.944549216216155e+276, difference: 7.944549216216155e+276
old x: 7.944549216216155e+276, new x: NaN, difference: NaN
Final y value: NaN Final x value: NaN Number of iterations: 8

This method found none of the roots. In fact, for each initial guess, the function diverged to values too large for the computer to handle within 6-8 iterations. This was completely unstable.

6 Secants behaving badly

The slope of the function that $\Omega_1 - \Omega_2$ produces is very steep on the domain $-1 \leq x \leq 1$. $\Omega_1 - \Omega_2 = 3x^2 - 7.9x + 1.75$. Secant method fails with large slopes around the root because the method uses the slope between points on the function, which will not differ that much if the slope is already near vertical, so the method will hardly gain effeciency at each step. Newton's method or Bisection method would better solve a function like this.

7 Convergence, or the lack thereof

1. $x_0 = 2$, first criteria

   Iterations: 30, calculated value: 403.4287934925369, actual value: 403.428793492735
2. $x_0 = -2$, first criteria

   Iterations: 29, calculated value: 0.002478751478959605, actual value: 0.0024787521766663594
3. $x_0 = -12$, first criteria

   Iterations: 100, calculated value: 0.05229714454636593, actual value: 2.319522830243574e-16
4. $x_0 = 2$, second criteria

   Iterations: 36, calculated value: 403.4287934927351, actual value: 403.428793492735

   Note that the answer is accurate to all decimal places that were outputted.
5. $x_0 = -2$, second criteria

   Iterations: 36, calculated value: 0.0024787521766719985, actual value: 0.0024787521766663594

   This was accurate to 13 decimal places of accuracy.
6. $x_0 = -12$, second criteria

   Iterations: 122, calculated value: 0.05109305371814203, actual value: 2.319522830243574e-16

   This was accurate to 1 decimal place of accuracy.

7. $x_0 = 20$, second criteria

   Iterations: 188, calculated value: 1.1420073898156829e+26, actual value: 1.1420073898156806e+26

8. $x_0 = -20$, second criteria

   Iterations: 188, calculated value: -1315533380.7076263, actual value: 8.756510762696549e-27

   The summation was accurate for positive x values, but inaccurate for negative x values (and very innacurate for large negative x values). This is because the series diverges due to the negative beign within the power, so the terms being added switch off being positive and negative, so the series gets further from the true answer (although for small x-values, it does not throw off the series as much, so we can still get results to a certain degree of accuracy.

8 Newton's method

2D method - initial guess of $.5$ for v and 1 for w.

1: v: -1.0935968922950163, w: 2.216943825417973
2: v: -0.02641888966774464, w: 2.172591985348798
3: v: 0.017665840703981778, w: 2.190791850802902
4: v: 0.016933807603772656, w: 2.1918846825617706
5: v: 0.016934795162778068, w: 2.191866669273077
final v: 0.016934795162778068, final w: 2.191866669273077, number of iterations: 5

The 1D method does not work at this moment.

9 Blackbody radiation

1.

2.

3.

4. xold: 4.0, xnew: 3.145551586016073, y value: -8.919768895944435e-20
6.982509981073105e-19 xold: 3.145551586016073, xnew: 2.5801531082601983, y value: 7.938790872291784e-20
6.982509981073105e-19 xold: 2.5801531082601983, xnew: 2.8464024401202717, y value: -7.610525634677855e-21
6.982509981073105e-19 xold: 2.8464024401202717, xnew: 2.8231112466603054, y value: -5.134205172949907e-22
6.982509981073105e-19 xold: 2.8231112466603054, xnew: 2.8214263094363456, y value: 4.01354624142499e-24
6.982509981073105e-19 xold: 2.8214263094363456, xnew: 2.8214393788763745, y value: -2.075267602873835e-27
6.982509981073105e-19 xold: 2.8214393788763745, xnew: 2.8214393721221063, y value: -8.42350406614243e-33
6.982509981073105e-19 xold: 2.8214393721221063, xnew: 2.8214393721220787, y value: 3.9546967446679624e-35
Final y value: 3.9546967446679624e-35
Final x value: 2.8214393721220787
Number of iterations: 8

   We used the secant method to find the root, which was found within 8 iterations. Note that the tolerance had to be set very small so that any approximation of the root could be found since the slope around the root is very shallow.

5. xold: 9.0, xnew: 6.188913538984333, y value: -1.5590138013365619e-19

   xold: 6.188913538984333, xnew: 4.609805618150875, y value: 1.9142759422095137e-19

   xold: 4.609805618150875, xnew: 5.48011835512334, y value: -1.5186879843665448e-20

   xold: 5.48011835512334, xnew: 5.416147352982033, y value: -1.0207588908259355e-21

   xold: 5.416147352982033, xnew: 5.411537836417254, y value: 8.3482215676381e-24

   xold: 5.411537836417254, xnew: 5.411575229284468, y value: -4.444986030038481e-27

   xold: 5.411575229284468, xnew: 5.411575209385341, y value: -1.935559594109172e-32

   Final y value: -1.935559594109172e-32

   Final x value: 5.411575209385341

   Number of iterations: 7
xmid: 1.0, y value: 3.780494521708601e-19

   xmid: 1.0, y value: 1.8059520852131914e-19 xmid: 1.0, y value: 5.1364613435260135e-20

xmid: 1.125, y value: -1.833349821603271e-20

xmid: 1.125, y value: 1.682899640026377e-20

xmid: 1.15625, y value: -6.833824506200851e-22

xmid: 1.15625, y value: 8.091246981443213e-21

xmid: 1.15625, y value: 3.708391710152226e-21

xmid: 1.15625, y value: 1.5136004705432396e-21

xmid: 1.15625, y value: 4.1538058001023157e-22

xmid: 1.1572265625, y value: -1.339333423495234e-22

xmid: 1.1572265625, y value: 1.407405545490535e-22

xmid: 1.1572265625, y value: 3.407835346781976e-24

xmid: 1.1573486328125, y value: -6.526169677519548e-23

xmid: 1.15740966796875, y value: -3.092666645947761e-23

xmid: 1.157440185546875, y value: -1.375934948356551e-23

xmid: 1.1574554443359375, y value: -5.1757405490647066e-24

xmid: 1.1574630737304688, y value: -8.839484711771925e-25

xmid: 1.1574630737304688, y value: 1.2619444702452868e-24

xmid: 1.1574630737304688, y value: 1.8899825765680798e-25

xmid: 1.1574640274047852, y value: -3.4747504228968735e-25

xmid: 1.1574645042419434, y value: -7.923837613862815e-26

xmid: 1.1574645042419434, y value: 5.48799448035428e-26

xmid: 1.157464623451233, y value: -1.2179214656429457e-26

xmid: 1.157464623451233, y value: 2.1350365266149664e-26

xmid: 1.157464623451233, y value: 4.585575449304849e-27

xmid: 1.157464638352394, y value: -3.796819603562304e-27

xmid: 1.157464638352394, y value: 3.9437787472302413e-28

xmid: 1.1574646420776844, y value: -1.7012209125678885e-27

xmid: 1.1574646439403296, y value: -6.534215189224322e-28

xmid: 1.1574646448716521, y value: -1.2952177395145541e-28

xmid: 1.1574646448716521, y value: 1.3242809853403297e-28

xmid: 1.1574646448716521, y value: 1.4531141430401695e-30

xmid: 1.1574646449880674, y value: -6.403437805245623e-29

xmid: 1.157464645046275, y value: -3.129058380645942e-29

xmid: 1.157464645075379, y value: -1.4918734831709626e-29

xmid: 1.1574646450899309, y value: -6.732858492583338e-30
xmid: 1.1574646450972068, y value: -2.6399684712688035e-30
xmid: 1.1574646451008448, y value: -5.933790158657074e-31
Final y value: -5.933790158657074e-31
Final x value: 1.1574646451044828
Number of iterations: 39