**1.4 CP #4.**

Carry out the steps of Computer Problem 3 for

**(a) f (x) = 2e^(x−1) − x2 − 1**

f’(x) = 2e^(x−1) −2\*x

Input the function and its derivative into the code:

double f(double x) { return 2\*exp(x-1)-x\*x-1; }

double df(double x) { return 2\*exp(x-1)-2\*x; }

Then the output with each step is:

Enter guess at root: 0.5

Enter tolerance, max iteration, and debug flag: 1e-6 100 1

Guess: x = 0.5

Iter 1: x = 0.673371, dx = 0.173371

Iter 2: x = 0.785145, dx = 0.111774

Iter 3: x = 0.858027, dx = 0.072882

Iter 4: x = 0.905906, dx = 0.0478789

Iter 5: x = 0.937515, dx = 0.0316091

Iter 6: x = 0.958451, dx = 0.0209364

Iter 7: x = 0.972349, dx = 0.0138974

Iter 8: x = 0.981587, dx = 0.00923831

Iter 9: x = 0.987734, dx = 0.00614708

Iter 10: x = 0.991827, dx = 0.00409283

Iter 11: x = 0.994553, dx = 0.00272623

Iter 12: x = 0.99637, dx = 0.00181646

Iter 13: x = 0.99758, dx = 0.00121051

Iter 14: x = 0.998387, dx = 0.000806805

Iter 15: x = 0.998925, dx = 0.00053778

Iter 16: x = 0.999283, dx = 0.000358479

Iter 17: x = 0.999522, dx = 0.000238969

Iter 18: x = 0.999681, dx = 0.000159305

Iter 19: x = 0.999788, dx = 0.000106199

Iter 20: x = 0.999858, dx = 7.0798e-05

Iter 21: x = 0.999906, dx = 4.71947e-05

Iter 22: x = 0.999937, dx = 3.14603e-05

Iter 23: x = 0.999958, dx = 2.09655e-05

Iter 24: x = 0.999972, dx = 1.39912e-05

Iter 25: x = 0.999981, dx = 9.35933e-06

Iter 26: x = 0.999988, dx = 6.40362e-06

Iter 27: x = 0.999992, dx = 4.46263e-06

Iter 28: x = 0.999996, dx = 3.69215e-06

Iter 29: x = 1, dx = 6.72594e-06

Iter 30: x = 1, dx = -0

The root is 1.000003

**Find the multiplicity of the root:**

f’(r) = f’(1) = 2e^(1−1) −2\*1 = 2-2 = 0.

find the second derivative:

f’’(x) = 2e^(x−1) −2

f’’(r) = f’’(1) = 2e^(1−1) −2 = 2 - 2 = 0

find the third derivative:

f’’’(x) = 2e^(x−1)

f’’’(r) = f’’’(1) = 2e^(1−1) = 2

Thus, the multiplicity of the root r = 1 is 3.

Applying x(i+1) = x(i) - m\*f(x)/f’(x) = x(i) - 3\*f(x)/f’(x)

In coding, I changed the dx:

double dx = -3\*f(x)/dfx;

x += dx;

**Observe Converge to Root Quadratically:**

Enter guess at root: 0.5

Enter tolerance, max iteration, and debug flag: 1e-6 100 1

Guess: x = 0.5

Iter 1: x = 1.02011, dx = 0.520113

Iter 2: x = 1.00003, dx = -0.0200796

Iter 3: x = 1, dx = -3.3319e-05

Iter 4: x = 1, dx = -0

The root is 1

**Report Backward and Forward Error**

Since we set accuracy tolerence was 1e-6, the forward error is 1e-6.

The backward error is:

2e^(1e-6) − (1e-6+1)^2 − 1 = 3.33333 × 10^-22

**(b) f (x) = ln(3 − x) + x − 2.**

f’(x) = -1/(3-x) + 1

Input the function and its derivative:

double f(double x) { return log(3-x)+x-2; }

double df(double x) { return 1/(3-x)+1; }

The output result is:

Enter guess at root: 1

Enter tolerance, max iteration, and debug flag: 1e-6 100 1

Guess: x = 1

Iter 1: x = 1.61371, dx = 0.613706

Iter 2: x = 1.82781, dx = 0.214102

Iter 3: x = 1.91846, dx = 0.0906518

Iter 4: x = 1.96029, dx = 0.0418353

Iter 5: x = 1.98041, dx = 0.0201102

Iter 6: x = 1.99027, dx = 0.00986083

Iter 7: x = 1.99515, dx = 0.00488276

Iter 8: x = 1.99758, dx = 0.00242958

Iter 9: x = 1.99879, dx = 0.00121185

Iter 10: x = 1.9994, dx = 0.000605193

Iter 11: x = 1.9997, dx = 0.000302414

Iter 12: x = 1.99985, dx = 0.000151161

Iter 13: x = 1.99992, dx = 7.55692e-05

Iter 14: x = 1.99996, dx = 3.77817e-05

Iter 15: x = 1.99998, dx = 1.88902e-05

Iter 16: x = 1.99999, dx = 9.44489e-06

Iter 17: x = 2, dx = 4.7224e-06

Iter 18: x = 2, dx = 2.3612e-06

Iter 19: x = 2, dx = 1.18058e-06

Iter 20: x = 2, dx = 5.90376e-07

The root is 1.999999

**Find the multiplicity of the root:** at root = 2:

f’(r) = f’(2) = -1/(3-2) + 1 = -1 + 1 = 0

Then find th second derivative:

f’’(x) = -1/(-3 + x)^2

f’’(r) = f’’(2) = -1/(-3 + 2)^2 = -1

Thus the multiplicity is 2.

Applying x(i+1) = x(i) - m\*f(x)/f’(x) = x(i) - 2\*f(x)/f’(x)

In coding, I changed the dx:

double dx = -2\*f(x)/dfx;

x += dx;

**Observe Converge to Root Quadratically:**

Then the result output with debug is:

Enter guess at root: 1

Enter tolerance, max iteration, and debug flag: 1e-6 100 1

Guess: x = 1

Iter 1: x = 2.22741, dx = 1.22741

Iter 2: x = 2.01951, dx = -0.207897

Iter 3: x = 2.00013, dx = -0.0193865

Iter 4: x = 2, dx = -0.000128189

Iter 5: x = 2, dx = 0

The root is 2

**Report Backward and Forward Error**

Since we set accuracy tolerence was 1e-6, the forward error is 1e-6.

The backward error is:

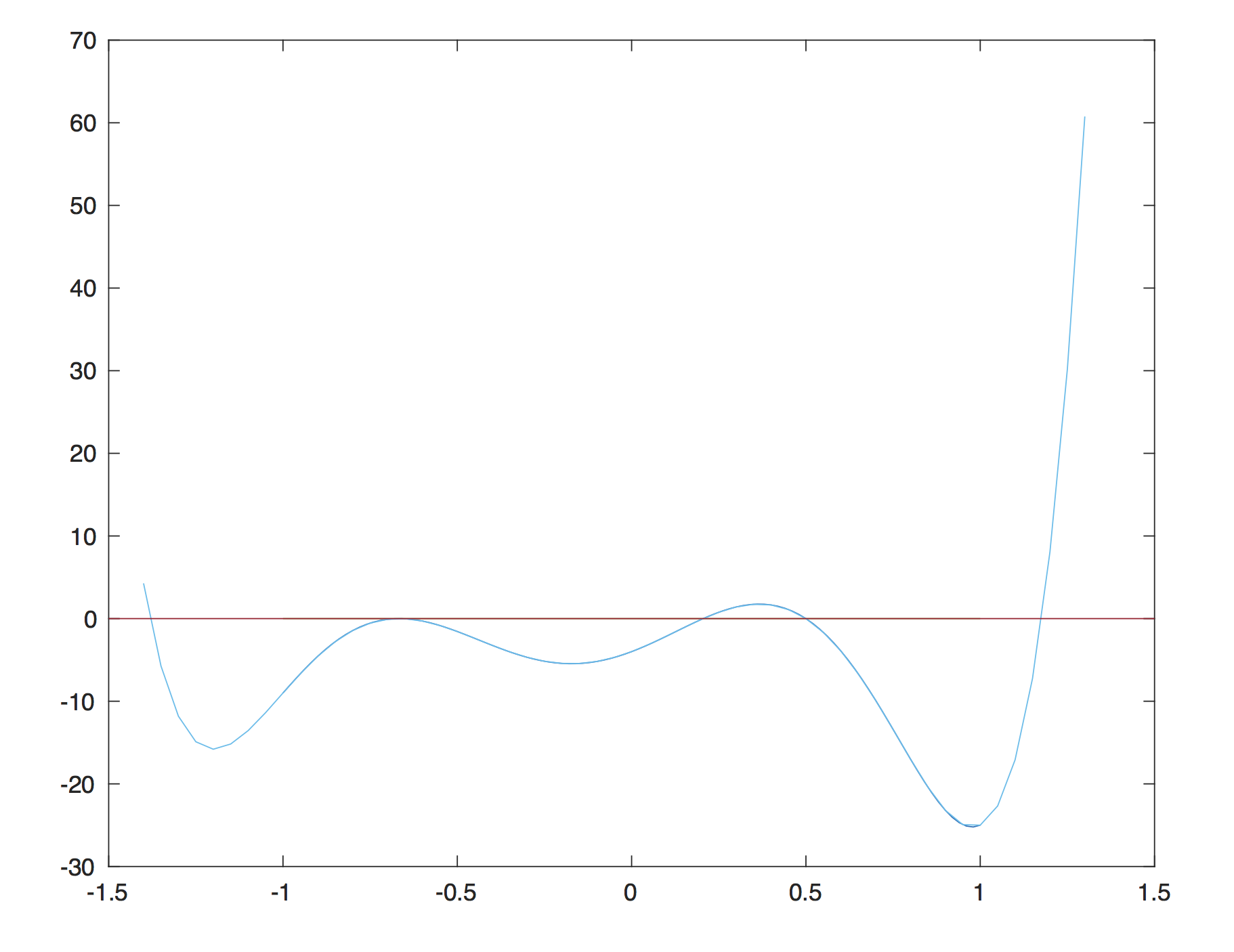
ln(3 − 2 + 1e-6) + (2 + 1e-6) − 2 = -5.00000 × 10^-15

**1.5 CP # 4**

4. Set f (x) = 54x6 + 45x5 − 102x4 − 69x3 + 35x2 + 16x − 4. Plot the function on the interval [−2,2], and use the Secant Method to find all five roots in the interval. To which of the roots is the convergence linear, and to which is it superlinear?

Hints: The roots are near -1.4, -0.7, 0.2, 0.5, 1.2.

Plot from hints:



**Initial guesses for -1.4: -1.5, -1.3**

Enter 2 guesses at root: -1.5 -1.3

Enter tolerance, max iteration, and debug flag: 1e-6 100 1

Guess 0: x = -1.5

Guess 1: x = -1.3

Iter 1: x = -1.34506, dx = -0.0450562

Iter 2: x = -1.40008, dx = -0.0550242

Iter 3: x = -1.3782, dx = 0.0218773

Iter 4: x = -1.38106, dx = -0.00285291

Iter 5: x = -1.3813, dx = -0.000245765

Iter 6: x = -1.3813, dx = 3.29936e-06

Iter 7: x = -1.3813, dx = -3.47974e-09

The root is -1.381298

Converge superlinear to root -1.381298.

**Initial guesses for -0.7: -1, -0.5**

Enter 2 guesses at root: -1 -0.5

Enter tolerance, max iteration, and debug flag: 1e-6 100 1

Guess 0: x = -1

Guess 1: x = -0.5

Iter 1: x = -0.394958, dx = 0.105042

Iter 2: x = -0.593492, dx = -0.198534

Iter 3: x = -0.617057, dx = -0.0235647

Iter 4: x = -0.638393, dx = -0.021336

Iter 5: x = -0.649095, dx = -0.0107026

Iter 6: x = -0.655979, dx = -0.00688313

Iter 7: x = -0.660075, dx = -0.00409623

Iter 8: x = -0.662609, dx = -0.00253442

Iter 9: x = -0.664163, dx = -0.00155331

Iter 10: x = -0.665121, dx = -0.000958451

Iter 11: x = -0.665712, dx = -0.000590986

Iter 12: x = -0.666077, dx = -0.000364917

Iter 13: x = -0.666302, dx = -0.000225361

Iter 14: x = -0.666441, dx = -0.000139226

Iter 15: x = -0.666528, dx = -8.60233e-05

Iter 16: x = -0.666581, dx = -5.3157e-05

Iter 17: x = -0.666614, dx = -3.28495e-05

Iter 18: x = -0.666634, dx = -2.03009e-05

Iter 19: x = -0.666646, dx = -1.25462e-05

Iter 20: x = -0.666654, dx = -7.75378e-06

Iter 21: x = -0.666659, dx = -4.79203e-06

Iter 22: x = -0.666662, dx = -2.96161e-06

Iter 23: x = -0.666664, dx = -1.83038e-06

Iter 24: x = -0.666665, dx = -1.13122e-06

Iter 25: x = -0.666666, dx = -6.99143e-07

The root is -0.666666

Converge linear to root -0.666666.

**Initial guesses for 0.2: 0, 0.4**

Enter 2 guesses at root: 0 0.4

Enter tolerance, max iteration, and debug flag: 1e-6 100 1

Guess 0: x = 0

Guess 1: x = 0.4

Iter 1: x = 0.282946, dx = -0.117054

Iter 2: x = -0.0472995, dx = -0.330246

Iter 3: x = 0.214485, dx = 0.261784

Iter 4: x = 0.205248, dx = -0.00923703

Iter 5: x = 0.205182, dx = -6.55545e-05

Iter 6: x = 0.205183, dx = 9.51536e-07

The root is 0.205183

Converge superlinear to root 0.205183.

**Initial guesses for 0.5: 0.4, 0.6**

Enter 2 guesses at root: 0.4 0.6

Enter tolerance, max iteration, and debug flag: 1e-6 100 1

Guess 0: x = 0.4

Guess 1: x = 0.6

Iter 1: x = 0.459531, dx = -0.140469

Iter 2: x = 0.486551, dx = 0.0270195

Iter 3: x = 0.50286, dx = 0.0163086

Iter 4: x = 0.499833, dx = -0.00302647

Iter 5: x = 0.499998, dx = 0.000164878

Iter 6: x = 0.5, dx = 1.96467e-06

Iter 7: x = 0.5, dx = -1.36316e-09

The root is 0.5

Converge superlinear to root 0.5.

**Initial guesses for 1.2: 1, 1.3**

Enter 2 guesses at root: 1 1.3

Enter tolerance, max iteration, and debug flag: 1e-6 100 1

Guess 0: x = 1

Guess 1: x = 1.3

Iter 1: x = 1.08745, dx = -0.212551

Iter 2: x = 1.13777, dx = 0.0503254

Iter 3: x = 1.19587, dx = 0.0580949

Iter 4: x = 1.17296, dx = -0.0229051

Iter 5: x = 1.17588, dx = 0.00291327

Iter 6: x = 1.17612, dx = 0.000241101

Iter 7: x = 1.17612, dx = -3.01846e-06

Iter 8: x = 1.17612, dx = 2.86298e-09

The root is 1.176116

Converge superlinear to root 1.176116.