Wai Phyo

CS-301

Project-1

Prof. A. Raheja

**NOTE-1**: The project states that the stopping point for the algorithms is when the approximate or true error is less than 1% or 0.01. For more detailed analysis, the error is reduced to **0.0001**. All algorithms are run for **100** iterations and delta for Modified Secant is **0.01** as stated in the project.

**NOTE-2**: The actual condition to stop is when the true or approximate error is less than 0.0001. In order to generate graphs, the program overwrites this condition with the minimum number of iterations which is 10. That means the program will keep on computing at least 10 times. Some exceptions are when the error becomes 0, infinity, or NaN (Not a Number). For those cases, the program will stop.   
**NOTE-3**: Only relevant data is selected for this report. Full output for each program is submitted via Blackboard in the file result.txt

NOTE-4: All calculations are done using **Double** data type. Hence, the precision is around 10-17.

*================================*

First function is a polynomial, and the problem states that all roots lies between 0 and 4. In order to verify whether Bisection & False Position methods are validating correctly, the results are printed. First, the functional values are printed.

*================================*

*PART - 1*

*================================*

*at x = 0, f(x) = -5.0*

*at x = 1, f(x) = 3.0*

*at x = 2, f(x) = -0.3999999999999986*

*at x = 3, f(x) = -3.1999999999999957*

*at x = 4, f(x) = 6.6000000000000085*

*================================*

From the table, we can verify that those two methods should have some validation when the values are 2 and 3.

The graph is generated from the values to analyze the nature of the graph.

Hypothesis for **Bisections** and **False Position** is that (0,1) will point to root around 0.4. (1, 2) will point to root around 2. (2, 3) will return error, and (3,4) will point to root around 3.6. **False Position** will arrive to the root much faster than **Bisection** as it includes how far f(x) from the origin in the calculation.

From the graph, hypothesis for Newton and **Modified** **Secant** are the followings. The root for 0 & 1 is 0.4, for 2 is 2, for 3 & 4 is 3. 6. Iterations for 1 and 3 will be bigger than others for **both** methods. For **Modified** **Secant**, x cannot start as 0 as it will result in NaN. Hence, it is started from 0.1.

For **Secant**, both positives and negatives are chosen as choosing positive and negative will yield similar results to **False Position**. The root for (-1, 0) is 0.4, (0.5, 1.5) is 0.4, (1, 1.5) is 2, and (2.5 and 3.5) is 3.6. Iterations for (0.5,1.5) will be the largest.

*================================*

*Bisection*

*-----------------------------*

*Total Iterations: 15*

*c: [0.5, 0.25, 0.375, 0.3125, 0.34375, 0.359375, 0.3671875, 0.36328125, 0.365234375, 0.3642578125, 0.36474609375, 0.364990234375, 0.3651123046875, 0.36505126953125, 0.365081787109375]*

*Error\_a: [1.0, 0.3333333333333333, 0.2, 0.09090909090909091, 0.043478260869565216, 0.02127659574468085, 0.010752688172043012, 0.0053475935828877, 0.002680965147453083, 0.0013386880856760374, 6.688963210702341E-4, 3.3433634236041456E-4, 1.6719612104999163E-4, 8.359107247345983E-5]*

*================================*

Result can confirm the hypothesis. It needed around **14** **iterations** to get around 0.0001.

*================================*

*Total Iterations: 13*

*c: [1.5, 1.75, 1.875, 1.9375, 1.90625, 1.921875, 1.9140625, 1.91796875, 1.919921875, 1.9208984375, 1.92138671875, 1.921630859375, 1.9217529296875]*

*Error\_a: [0.14285714285714285, 0.06666666666666667, 0.03225806451612903, 0.01639344262295082, 0.008130081300813009, 0.004081632653061225, 0.002036659877800407, 0.001017293997965412, 5.083884087442806E-4, 2.5412960609911054E-4, 1.2704865963664082E-4, 6.352029473416756E-5]*

*================================*

Result can confirm the hypothesis. It needed around **13** **iterations** to get around 0.0001.

*================================*

*java.lang.IllegalArgumentException: Invalid initial points*

*================================*

As expected, it threw exception.

*================================*

*Total Iterations: 12*

*c: [3.5, 3.75, 3.625, 3.5625, 3.59375, 3.578125, 3.5703125, 3.56640625, 3.564453125, 3.5634765625, 3.56298828125, 3.563232421875]*

*Error\_a: [0.06666666666666667, 0.034482758620689655, 0.017543859649122806, 0.008695652173913044, 0.004366812227074236, 0.002188183807439825, 0.001095290251916758, 5.47945205479452E-4, 2.740476842970677E-4, 1.3704262025489926E-4, 6.851661527920521E-5]*

*================================*

As expected, it points to root at 3.56, around 3.6. It needed around **11** **iterations** for error of 0.0001.

*================================*

*False Position*

*-----------------------------*

*Total Iterations: 11*

*c: [0.625, 0.4476776720761052, 0.3887117536768692, 0.3716293366827822, 0.36688756141399365, 0.3655871767763859, 0.3652317490577878, 0.36513469059780485, 0.36510819297341723, 0.36510095943417314, 0.3650989847999825]*

*Error\_a: [0.39609375, 0.1516957432891354, 0.04596627689990025, 0.012924328234278732, 0.0035569755183265687, 9.73156686172572E-4, 2.658154989986669E-4, 7.257471866576443E-5, 1.9812435593978205E-5, 5.408489951682342E-6]*

*================================*

As expected, it points to 0.365. It needed around **8** **iterations** for error of 0.0001 which is few than Bisection.

*================================*

*Total Iterations: 7*

*c: [1.882352941176471, 1.921685522661067, 1.9217409206219749, 1.9217409317734833, 1.9217409317757086, 1.9217409317757097, 1.9217409317757097]*

*Error\_a: [0.020467751367627466, 2.8826966379001664E-5, 5.802815665213924E-9, 1.1579766001562109E-12, 5.777173219697716E-16, 0.0]*

*================================*

As expected, it points to 1.922. It needed around **3** **iterations** for error of 0.0001 which is few than Bisection.

*================================*

*java.lang.IllegalArgumentException: Invalid initial points*

*================================*

As expected, it threw exception.

*================================*

*Total Iterations: 11*

*c: [3.3265306122448974, 3.4812727094176568, 3.537095819952593, 3.555101343844147, 3.560691651684571, 3.5624065006842396, 3.562930582487523, 3.563090566725527, 3.5631393874361987, 3.563154283956879, 3.5631588291412783]*

*Error\_a: [0.044449863624342265, 0.015782187810700656, 0.005064700595028533, 0.0015700061637685583, 4.8137375657134756E-4, 1.470929032014231E-4, 4.490041300045665E-5, 1.370160001151214E-5, 4.1807116652703694E-6, 1.275605331523633E-6]*

*================================*

As expected, it points to 3.562 It needed around **7** **iterations** for error of 0.0001 which is few than Bisection.

*================================*

*Newton*

*-----------------------------*

*Total Iterations: 7*

*a: [0.0, 0.2824858757062147, 0.3592933065396092, 0.3650663350424796, 0.36509824238985855, 0.3650982433622361, 0.3650982433622361]*

*Error\_a: [1.0, 0.21377362014654488, 0.015813642477328436, 8.739386738785056E-5, 2.663331267769176E-9, 0.0]*

*================================*

As expected, it points to 0.365. It needed only **5 iterations** for error to be less than 0.0001.

*================================*

*Total Iterations: 12*

*a: [1.0, -8.999999999999977, -5.402071958560814, -3.0294698046987008, -1.4868398675796246, -0.5172574659581148, 0.04189935119596255, 0.29765259287102497, 0.3611498406093906, 0.36508344055813463, 0.3650982431529569, 0.3650982433622362]*

*Error\_a: [-1.1111111111111114, -0.6660274185606554, -0.7831740557975567, -1.0375225811171382, -1.874467678925725, 13.345238081107997, 0.8592340460003389, 0.1758196753769094, 0.010774523058976335, 4.054414147394694E-5, 5.732135511269841E-10]*

*================================*

As expected, it also points to 0.35. It needed **11 iterations** for error of around 0.0001 as predicted.

*================================*

*Total Iterations: 9*

*a: [2.0, 1.9215686274509807, 1.9217409327647854, 1.9217409317757088, 1.9217409317757093, 1.9217409317757095, 1.92174093177571, 1.921740931775709, 1.921740931775709]*

*Error\_a: [0.040816326530612096, 8.966105205286444E-5, 5.146773855137523E-10, 2.310869287879087E-16, 1.1554346439395432E-16, 2.310869287879086E-16, 4.621738575758174E-16, 0.0]*

*================================*

As expected, it points to 1.922 which is around 2, and only **3 iterations** are needed.

*================================*

*Total Iterations: 12*

*a: [3.0, 5.133333333333326, 4.269750056533243, 3.792934480643228, 3.599819288398082, 3.5643380328473415, 3.5631621003251914, 3.5631608248635547, 3.5631608248620545, 3.563160824862055, 3.563160824862054, 3.5631608248620545]*

*Error\_a: [0.4155844155844147, 0.20225616613756908, 0.1257115244999312, 0.053645801851093015, 0.009954514758073196, 3.300249859647896E-4, 3.5795791979175423E-7, 4.2101196791519737E-13, 1.2463350145506139E-16, 2.492670029101228E-16, 1.2463350145506139E-16]*

*================================*

As expected, it points to 3.563. It needed **7 iterations**.

*================================*

*Total Iterations: 12*

*a: [4.0, 3.6716417910447756, 3.5725543717519477, 3.563241095182271, 3.563160830801589, 3.5631608248620545, 3.563160824862055, 3.563160824862054, 3.5631608248620545, 3.563160824862055, 3.563160824862054, 3.5631608248620545]*

*Error\_a: [0.08943089430894324, 0.027735734430330415, 0.0026137093508123163, 2.2526173948711356E-5, 1.6669285878014266E-9, 1.2463350145506139E-16, 2.492670029101228E-16, 1.2463350145506139E-16, 1.2463350145506139E-16, 2.492670029101228E-16, 1.2463350145506139E-16]*

*================================*

As expected, it also points to 3.56. It needed **5 iteration**s.

From the experiments, hypothesis is confirmed as iterations for x = 1 and 3 are higher than others.

================================

Modified Secant

-----------------------------

Total Iterations: 10

a: [0.1, 0.3170823073571384, 0.36317459592446955, 0.3651013810960689, 0.36509823238141836, 0.365098243400631, 0.36509824336210195, 0.3650982433622367, 0.3650982433622361, 0.3650982433622361]

Error\_a: [0.6846244723223636, 0.12691495794192914, 0.0052773976527148185, 8.624294426121303E-6, 3.018150005547893E-8, 1.0553068804024922E-10, 3.690118111704885E-13, 1.5204442157828146E-15, 0.0]

================================

As expected, it points to 0.365. It needed **5** **iterations**.

================================

Total Iterations: 12

a: [1.0, -11.335526315795157, -6.987253025943172, -4.095056493774239, -2.188875957502407, -0.9591347343593222, -0.20626709889441164, 0.19561027900202876, 0.3434089630544954, 0.3647310022938069, 0.36509939635038313, 0.3650982393294628]

Error\_a: [-1.0882182240278142, -0.6223151320994326, -0.7062653559397709, -0.8708490445693681, -1.282136053559275, -3.6499647277741776, 2.054479856307921, 0.4303867981133985, 0.058459629439823795, 0.001009024008965199, 3.1690673788933994E-6]

================================

As expected, it also points to 0.365. It needed **11** **iterations**.

================================

Total Iterations: 11

a: [2.0, 1.9214639126678712, 1.9217407960918251, 1.9217409317088383, 1.9217409317756766, 1.9217409317757097, 1.9217409317757093, 1.9217409317757095, 1.92174093177571, 1.921740931775709, 1.921740931775709]

Error\_a: [0.04087304831194294, 1.4407948486966655E-4, 7.056987283629376E-8, 3.478008484761797E-11, 1.7215976194699192E-14, 2.310869287879087E-16, 1.1554346439395432E-16, 2.310869287879086E-16, 4.621738575758174E-16, 0.0]

================================

As expected, it points to 1.922. It needed **3** **iterations**.

================================

Total Iterations: 12

a: [3.0, 4.892595221197089, 4.14294941812408, 3.742315956197904, 3.5910547395828694, 3.564700590048562, 3.5632121873916636, 3.5631624723484676, 3.5631608776355645, 3.563160826552462, 3.5631608249162, 3.563160824863789]

Error\_a: [0.38682848991828533, 0.18094495670006208, 0.10705495383484648, 0.04212166830756941, 0.007393089228273201, 4.177137309322531E-4, 1.3952505276355118E-5, 4.475556837955021E-7, 1.433645718602625E-8, 4.5921637364324494E-10, 1.470912120821773E-11]

================================

As expected, it points to 3.563. It needed **8 iterations**.

================================

Total Iterations: 12

a: [4.0, 3.6795369794903667, 3.5769413811564412, 3.5637630305606276, 3.5631804270344953, 3.5631614530771687, 3.5631608449849557, 3.5631608255066176, 3.5631608248827007, 3.5631608248627153, 3.563160824862076, 3.5631608248620545]

Error\_a: [0.08709330067774422, 0.028682493617146136, 0.0036978751063985573, 1.6350660261600712E-4, 5.325034404560608E-6, 1.7066089337053536E-7, 5.4665896397850235E-9, 1.7510209299925353E-10, 5.608881465981088E-12, 1.7947224209528733E-13, 5.982408069842947E-15]

================================

As expected, it also points to 3.65. It needed **5** **iterations**.

================================

Secant

-----------------------------

Total Iterations: 10

x\_i+1 [0.15923566878980894, 0.3147097607138443, 0.35678414501480216, 0.36471490535769685, 0.365095218960824, 0.36509824225524523, 0.36509824336223295, 0.36509824336223606, 0.3650982433622362, 0.3650982433622362]

Error\_a: [0.4940237365735951, 0.11792672092873484, 0.021745095213798683, 0.001041683329104164, 8.280769588428785E-6, 3.03202697858723E-9, 8.514487608383763E-15, 3.0408884315656286E-16, 0.0]

================================

As expected, it points to 0.365. It needed **5** **iterations**.

================================

Total Iterations: 9

x\_i+1 [-0.9687499999999984, 0.45222135743443936, 0.351645905743149, 0.36628200628784907, 0.36511331483434395, 0.36509822630706706, 0.36509824336248164, 0.3650982433622362, 0.3650982433622362]

f(x\_i+1): [-34.94534912109368, 0.7965816941503707, -0.13566369319876959, 0.011772791464098376, 1.5005667270528988E-4, -1.698094775903769E-7, 2.4442670110147446E-12, 0.0, 0.0]

Error\_a: [3.142203113749316, 0.28601343012579583, 0.0399585573231736, 0.0032009006684277468, 4.132730917239516E-5, 4.671458954049927E-8, 6.723404322191605E-13, 0.0]

================================

As expected, it also points to 0.365. Surprisingly it only needed **6** **iterations**. The prediction was wrong as I predicted it will need many iterations from observing the graph visually.

================================

Total Iterations: 9

x\_i+1 [2.4634146341463423, 1.9251989933466107, 1.9214278307903532, 1.921740897187601, 1.92174093177607, 1.9217409317757095, 1.92174093177571, 1.9217409317757097, 1.9217409317757097]

f(x\_i+1): [-2.499949217219715, -0.017673360666855764, 0.0015999946493154837, 1.767524464568737E-7, -1.84297022087776E-12, 1.7763568394002505E-15, -4.440892098500626E-15, -1.7763568394002505E-15, -1.7763568394002505E-15]

Error\_a: [0.2795636412961867, 0.001962687588795005, 1.6290770400218424E-4, 1.7998507710345745E-8, 1.8764258617578184E-13, 2.310869287879086E-16, 1.1554346439395432E-16, 0.0]

================================

As predicted, it points to 1.922. It only needed **4** **iterations**.

================================

Total Iterations: 9

x\_i+1 [3.8124999999999916, 3.550396875393725, 3.5639466612814696, 3.563151497841562, 3.5631608181085968, 3.5631608248621123, 3.563160824862054, 3.5631608248620545, Infinity]

f(x\_i+1): [3.250488281249874, -0.13243249936950452, 0.008256260205653732, -9.792100062799847E-5, -7.090270415233135E-8, 6.10178574333986E-13, -2.6645352591003757E-15, -2.6645352591003757E-15, Infinity]

Error\_a: [0.07382361291009197, 0.0038019047913787924, 2.2316296132487499E-4, 2.6157301088485213E-6, 1.89537207455541E-9, 1.6326988690613045E-14, 1.2463350145506139E-16, NaN]

================================

As expected, it points to 3.563. It needed **5 iterations**. Without the validations, the vulnerability of the algorithm is exposed, and the result will return infinity or NaN.

Approximate Error Graph for root at 0.3650982433622362. **Bisection** and **False** **Position** reduces slowly at a constant rate. **Secant** reduces very rapidly, **Modified** **Secant** and **Newton** did not reduces significantly at first, but the reducing rate is getting faster as they iterate more.

Approximate Error Graph for root at 1.9217409317757097. Apart from **Bisection, all other methods** reduce rapidly to 0. This is because the true root is so close to 2 which all of the method originate from. One minor exception is the secant where it does not start from 2. Hence, it took a few more iterations to reduces rapidly.

Approximate Error Graph for root at 3.5631608248620545. The reducing patterns are similar to the first graph for **Bisection** and **False** **Position**. **Secant** and **Newton** has similar patterns, while **Modified** **Secant** is reducing close to constant rate. This may be because both first and third roots are near the center of the two edges while second root is very close to one edge.

*================================*

*PART - 2*

*================================*

From the graph, we can confirm that this function has only one true root. Based on the nature of the graph, all points will always converge to this root with some exceptions for Newton method.

*Based on the graph,* ***Bisection*** *and* ***False******Position*** *between (0, 1) should have similar iterations to get to the true root within error of 0.0001.* ***Newton*** *and* ***Modified******Secant*** *starting from 1 will need very few iterations. For* ***Secant****, (-1, 0) is chosen as starting point to experience different approach to the true root compared to* ***False******Position****. It will also need only a few iterations.*

*================================*

*Bisection*

*-----------------------------*

*Total Iterations: 12*

*c: [0.5, 0.75, 0.625, 0.5625, 0.59375, 0.578125, 0.5703125, 0.56640625, 0.568359375, 0.5673828125, 0.56689453125, 0.567138671875]*

*Error\_a: [0.3333333333333333, 0.2, 0.1111111111111111, 0.05263157894736842, 0.02702702702702703, 0.0136986301369863, 0.006896551724137931, 0.003436426116838488, 0.0017211703958691911, 8.613264427217916E-4, 4.3047783039173483E-4]*

*Error\_t: [0.11838858218705182, 0.32241712671942224, 0.10201427226618522, 0.00818715496043331, 0.04691355865287595, 0.019363201846221322, 0.0055880234428940054, 0.0012995657587696518, 0.002144228842062177, 4.2233154164626264E-4, 4.3861710856169455E-4, 8.14278345771597E-6]*

*================================*

It needed **12** **iterations** for both approximate and true error to be smaller than 0.0001 which can be claimed as high.

*================================*

*False Position*

*-----------------------------*

*Total Iterations: 11*

*c: [0.6126998367802821, 0.5721814120905075, 0.567703214235785, 0.5672055526330225, 0.567150214240495, 0.5671440603751026, 0.5671433760339215, 0.5671432999316328, 0.5671432914686629, 0.5671432905275366, 0.5671432904228786]*

*Error\_a: [0.07081394787317088, 0.007888272855299796, 8.773919797721316E-4, 9.757272612790568E-5, 1.0850621248538754E-5, 1.2066458147533642E-6, 1.3418529091714586E-7, 1.4922101711765043E-8, 1.6594153606872895E-9, 1.845354811799295E-10]*

*Error\_t: [0.08032634359525279, 0.008883331918654242, 9.872711987564766E-4, 1.0978289635157002E-4, 1.2208978960303935E-5, 1.358342972900057E-6, 1.5169697510217277E-7, 1.751168183521678E-8, 2.5895800848097612E-9, 9.301647225789419E-10, 7.456292412614173E-10]*

*================================*

It needed **5** **iterations** for true error and **4** **iterations** for approximate error. This contradicts the prediction as it converges much faster than **Bisection** even though the root is around the middle of two starting points.

*================================*

*Newton*

*-----------------------------*

*Total Iterations: 6*

*a: [1.0, 0.5378828427399902, 0.5669869914054133, 0.567143285989123, 0.5671432904097838, 0.5671432904097838]*

*Error\_a: [0.8591409142295228, 0.05133124587793714, 2.7558218103051026E-4, 7.794610131410855E-9, 0.0]*

*Error\_t: [0.0515926887894764, 2.755892511514275E-4, 7.072069853615181E-9, 7.22540283427593E-10, 7.22540283427593E-10]*

*================================*

It needed **4 iterations** for true error and **3 iterations** for approximate error. This confirmed the prediction of needing very few iterations.

*================================*

*Modified Secant*

*-----------------------------*

*Total Iterations: 9*

*a: [1.0, 0.5372626655366407, 0.5670096853649093, 0.567143424147485, 0.5671432902726529, 0.5671432904099245, 0.5671432904097837, 0.5671432904097838, 0.5671432904097838]*

*Error\_a: [0.8612869721761843, 0.052462983606927846, 2.3581121966943973E-4, 2.3605116097654187E-7, 2.4204048751315964E-10, 2.4821995058912425E-13, 1.9575705882423042E-16, 0.0]*

*Error\_t: [0.052686199396556846, 2.355747435374077E-4, 2.3653190890995617E-7, 4.807478199331989E-10, 7.227883076213026E-10, 7.225400876705341E-10, 7.22540283427593E-10, 7.22540283427593E-10]*

*================================*

It needed **3** **iterations** for both approximate and true error. This also confirmed the prediction of needing very few iterations.

*================================*

*Secant*

*-----------------------------*

*Total Iterations: 8*

*x\_i+1 [0.36787944117144233, 0.5444590443832716, 0.5662980367205954, 0.5671398084913074, 0.5671432898771658, 0.5671432904097835, 0.5671432904097838, 0.5671432904097838]*

*Error\_a: [0.324321186383904, 0.03856448534378175, 0.0014842403197744974, 6.1384590464372186E-6, 9.391237394610106E-10, 5.872711764726913E-16, 0.0]*

*Error\_t: [0.3513465685692193, 0.03999737988036203, 0.0014903698841337704, 6.138675629151709E-6, 2.1658404398314865E-10, 7.225396961564161E-10, 7.22540283427593E-10, 7.22540283427593E-10]*

*================================*

It needed only **4 iterations** for both approximate and true error to be less than 0.0001. This also confirmed the prediction of needing very few iterations.

Approximate Error Graph. Similar to polynomial problem, Bisection and False Position reduces at constant rate, Newton and Secant has very similar pattern, and Modified Secant reduces rapidly at a constant rate.

True Error Graph. Every method follows similar trend. One exception is the **Bisection** method. In all other graphs, **Bisection** method reduces at a constant rate. However, they are all approximate graphs. For true error, it is not a straight line, as it will fluctuate by switching between positives and negatives.