CRISTIAN B. JETOMO
MATH 175 - H4L

Exercise 2.2: Fixed Point Iteration

In this exercise, we wish to derive a fixed point iteration formula $j(x_n)$ to solve the equation

$$x^2 - 5 = 0 \tag{2.2.1}$$

We will perform an analysis to the obtained equation to show that the fixed point involving j converges to a solution of the equation. We implement the iteration formula and verify that for any initial guess x_0, j produces a sequence that will converge to its fixed point, and thus, converge to the solution of Equation 2.2.1. From Equation 2.2.1, we have

$$\begin{aligned} x^2 - 5 &= 0 \\ x^2 &= 5 \\ x &= \frac{5}{x} \\ x + x &= \frac{5}{x} + x \\ \frac{1}{2}(2x) &= \frac{1}{2}\left(\frac{5}{x} + x\right) \\ x &= \frac{1}{2}\left(x + \frac{5}{x}\right), \quad x \neq 0 \end{aligned}$$

Therefore, we have the fixed point iteration formula

$$x_{n+1} = j(x_n) = \frac{1}{2}\left(x_n + \frac{5}{x_n}\right) \tag{2.2.2}$$

Now, we use the Contraction Mapping Theorem to determine whether the obtained formula in Equation 2.2.2 converges to a fixed point, particularly, to the roots of Equation 2.2.1 which are $p = \pm\sqrt{5}$. To do this, we compute for $j'(x)$.

$$\begin{aligned} j'(x) &= \frac{1}{2} \cdot \frac{d}{dx}\left(x + \frac{5}{x}\right) \\ &= \frac{1}{2}\left(1 - \frac{5}{x^2}\right) \end{aligned} \tag{2.2.3}$$

Note that

$$\left|j'(p)\right| = \left|j'\left(\pm\sqrt{5}\right)\right| = \frac{1}{2}\left(1 - \frac{5}{(\pm\sqrt{5})^2}\right) = \frac{1}{2} \cdot 0 = 0 < 1$$

Hence, by Corollary 3.1, the fixed point iteration $x_{n+1} = j(x_n)$ will converge to both the roots $p = \pm\sqrt{5}$, regardless of how far our initial guess to p is. The root (between $\sqrt{5}$ and $-\sqrt{5}$) to converge to will depend on our initial guess. We implement this iteration using MATLAB to verify. For our iteration, we will use the jump-based stopping criterion with maximum iteration $N_{\max} = 100$ and error tolerance $\epsilon = 10^{-6}$. Using five different initial guess which are $x_0 = 100, 10, 1, -1, -10$, we obtain the results in Figure 2.2.1.

$x_0 = 100$

n	x_n	x_(n+1)	rel jump	rel_error
1	50.025000	25.062475	4.990010E-01	4.776505E-01
2	25.062475	12.630988	4.960199E-01	4.553901E-01
3	12.630988	6.513420	4.843301E-01	4.114848E-01
4	6.513420	3.640533	4.410720E-01	3.283492E-01
5	3.640533	2.506979	3.113703E-01	1.928928E-01
6	2.506979	2.250706	1.022240E-01	5.403139E-02
7	2.250706	2.236116	6.482459E-03	3.251804E-03
8	2.236116	2.236068	2.128622E-05	1.064322E-05
9	2.236068	2.236068	2.265610E-10	0.000000E+00

Root is 2.2360679775 after 9 iterations.
Note that the absolute function value at the estimate is 0.000000E+00

$x_0 = 10$

n	x_n	x_(n+1)	rel jump	rel_error
1	5.250000	3.101190	4.092971E-01	2.870411E-01
2	3.101190	2.356737	2.400540E-01	1.394823E-01
3	2.356737	2.239157	4.989103E-02	2.560092E-02
4	2.239157	2.236070	1.378695E-03	6.898232E-04
5	2.236070	2.236068	9.530256E-07	4.763830E-07

Root is 2.2360679775 after 5 iterations.
Note that the absolute function value at the estimate is 4.540544E-12.

$x_0 = 1$

n	x_n	x_(n+1)	rel jump	rel_error
1	3.000000	2.333333	2.222222E-01	1.273220E-01
2	2.333333	2.238095	4.081633E-02	2.084258E-02
3	2.238095	2.236069	9.053871E-04	4.528986E-04
4	2.236069	2.236068	4.106060E-07	2.050810E-07

Root is 2.2360679775 after 4 iterations.
Note that the absolute function value at the estimate is 8.425614E-1

$x_0 = -1$

n	x_n	x_(n+1)	rel jump	rel_error
1	-3.000000	-2.333333	2.222222E-01	1.273220E-01
2	-2.333333	-2.238095	4.081633E-02	2.084258E-02
3	-2.238095	-2.236069	9.053871E-04	4.528986E-04
4	-2.236069	-2.236068	4.106060E-07	2.050810E-07

Root is -2.2360679775 after 4 iterations.
Note that the absolute function value at the estimate is 8.425614E-13.

$x_0 = -10$

n	x_n	x_(n+1)	rel jump	rel_error
1	-5.250000	-3.101190	4.092971E-01	2.870411E-01
2	-3.101190	-2.356737	2.400540E-01	1.394823E-01
3	-2.356737	-2.239157	4.989103E-02	2.560092E-02
4	-2.239157	-2.236070	1.378695E-03	6.898232E-04
5	-2.236070	-2.236068	9.530256E-07	4.763830E-07

Root is -2.2360679775 after 5 iterations.
Note that the absolute function value at the estimate is 4.540544E-12.

Figure 2.2.1: Screenshot of MATLAB results implementing the fixed point iteration $x_{n+1} = j(x_n)$ with five different initial points x_0.

Note from Figure 2.2.1 that using the values $100, 10,$ and 1 as initial guess, the iteration converges to $p = \sqrt{5}$. On the other hand, for values -1 and -10 used as initial guess, the iteration converges to $p = -\sqrt{5}$. It can also be noted that the farther the initial guess is to the root, the more iterations it requires to find a good approximate. However, it is still evident especially for $x_0 = 100$ that despite a large value for $|x_0 - p|$, the iteration still converges to its closest root (or fixed point). This verifies the consequence of the Contraction Mapping Theorem which guarantees the convergence to the fixed point since the condition $|j'(p)| < 1$ is satisfied.
We now perform error analysis on the results we obtained. In the 5^{th} columns of the results, the relative errors are calculated for each iteration using the equations below.

$$\epsilon_n = \frac{|\sqrt{5} - x_{n+1}|}{|\sqrt{5} - x_n|} \tag{2.2.4}$$

$$\epsilon_n = \frac{|-\sqrt{5} - x_{n+1}|}{|-\sqrt{5} - x_n|} \tag{2.2.5}$$

We use Equation 2.2.4 for $x_0 > 0$ and Equation 2.2.5 for $x_0 < 0$. Note that for each initial guess x_0 used, the relative errors decreases for every iteration, hence approaching to 0. This verifies another result of the Contraction Mapping Theorem that the relative errors eventually decreases by a factor of $j'(p) = 0$ as obtained. That is, $\epsilon_n \approx 0$ for sufficiently large n values.

Remark: *The program used to solve this problem is made in MATLAB and is made solely by the author of this paper.*