CRISTIAN B. JETOMO
MATH 175 - H4L

Exercise 1.2: The Regula Falsi Method

In this exercise, we will reconsider again the function solved from Exercise 1.1 which is given by

$$\frac{\rho_f}{3}h^3 - R\rho_f h^2 + \frac{4}{3}R^3\rho_0 = 0 \tag{1.2.1}$$

where $R = 5\text{cm}$, $\rho_0 = 0.12\ \text{g/cm}^3$, and $\rho_f = 0.89\ \text{g/cm}^3$. In this case, we will apply the Regula Falsi Method in estimating the root of the function

$$f(h) = \frac{\rho_f}{3}h^3 - R\rho_f h^2 + \frac{4}{3}R^3\rho_0 \tag{1.2.2}$$

Moreover, we shall compare our results with the one we obtained from Exercise 1.1. For our initial bracket, we use the interval $[0, 2.4]$ just as obtained from the previous exercise. Similarly, we will use the *interval-based halting criterion* in which we stop the iterations whenever the length of our interval is less than the error tolerance $\epsilon = 10^{-6}$ or whenever the maximum iterations $N_{\max} = 100$ is reached.

```matlab
%% Regula Falsi Method

% initial endpoints
a = 0;
b = 2.4;

% iteration counter
iter_count_reg = 0;

while iter_count_reg <= max_iteration

    % function values at current endpoints
    f_a = double(subs(func, a));
    f_b = double(subs(func, b));

    % current estimate
    c = (a*f_b - b*f_a)/(f_b - f_a);
%     c = a + ((b-a)*f_a)/(f_a-f_b);
    f_c = double(subs(func, c));

    % saves values in table
    table_regulafalsi(iter_count_reg+1,:) = [iter_count_reg, a, b, c, b-a];


    % interval-based stopping criterion
    if f_c == 0 || (b-a) < ErrorTol  % if root is found or error tol is met
        iter_count_reg = iter_count_reg + 1;
        break
    elseif f_a*f_c < 0
        b = c;
    else
        a = c;
    end

    iter_count_reg = iter_count_reg + 1;
end
```

Figure 1.2.1: Program used in MATLAB to perform Regula Falsi Method on the function f in Equation 1.2.2.

Figure 1.2.1 shows the program used in MATLAB to find the root using Regula Falsi Method. The initial endpoints are defined as with the endpoints used in the Bisection Method. *While* the iteration counter still do not reach the maximum iteration $N_{\max}$, the function values at the current endpoints a and b are evaluated. From these values, the current estimate c_n is determined using the equation

$$c_n = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)} \tag{1.2.3}$$

Note that c_n is defined as the x-intercept of the line passing through the current endpoints $(a_n, f(a_n))$ and $(b_n, f(b_n))$, hence the formula in Equation 1.2.3. The values of the iteration count, endpoints a_n, b_n, the current estimate c_n, and the length of the current interval $b_n - a_n$

a table are recorded for analysis reference. After this, the condition process is performed. If the current estimate is the actual root being determined or if the length of the current interval is smaller than the defined error tolerance $\epsilon = 10^{-6}$, the iteration process is halted. Otherwise, the new intervals are obtained as follows: if $f(a_n)f(c_n) < 0$, we let $a_{n+1} = a_n$ and $b_{n+1} = c_n$, otherwise, we let $a_{n+1} = c_n$ and $b_{n+1} = b_n$. The result of the program is summarized in Table 1.2.1. For comparison, the result of applying Bisection Method to the same problem is summarized in Table 1.2.2.

Table 1.2.1: Summarized table by applying the Regula Falsi Method on the given function.

n	a_n	b_n	c_n	$b_n - a_n$
0	0	2.4	2.22936	2.4
1	2.22936	2.4	2.30329	0.170644
2	2.30329	2.4	2.30436	0.096707
3	2.30436	2.4	2.30438	0.0956379
4	2.30438	2.4	2.30438	0.0956227
5	2.30438	2.4	2.30438	0.0956225
6	2.30438	2.4	2.30438	0.0956225
7	2.30438	2.4	2.30438	0.0956225
8	2.30438	2.4	2.30438	0.0956225
9	2.30438	2.4	2.30438	0.0956225
10	2.30438	2.4	2.30438	0.0956225
11	2.30438	2.30438	2.30438	4.44089×10^{-16}

Table 1.2.2: Summarized table by applying the Bisection Method on the given function.

n	a_n	b_n	c_n	$b_n - a_n$
0	0	2.4	1.2	2.4
1	1.2	2.4	1.8	1.2
2	1.8	2.4	2.1	0.6
3	2.1	2.4	2.25	0.3
4	2.25	2.4	2.325	0.15
5	2.25	2.325	2.2875	0.075
6	2.2875	2.325	2.30625	0.0375
7	2.2875	2.30625	2.29688	0.01875
8	2.29688	2.30625	2.30156	0.009375
9	2.30156	2.30625	2.30391	0.0046875
10	2.30391	2.30625	2.30508	0.00234375
11	2.30391	2.30508	2.30449	0.00117188
12	2.30391	2.30449	2.3042	0.000585937
13	2.3042	2.30449	2.30435	0.000292969
14	2.30435	2.30449	2.30442	0.000146484
15	2.30435	2.30442	2.30438	0.0000732422
16	2.30435	2.30438	2.30436	0.0000366211
17	2.30436	2.30438	2.30437	0.0000183105
18	2.30437	2.30438	2.30438	0.00000915527
19	2.30437	2.30438	2.30438	0.00000457764
20	2.30438	2.30438	2.30438	0.00000228882
21	2.30438	2.30438	2.30438	0.00000114441
22	2.30438	2.30438	2.30438	5.72205×10^{-7}

For the Bisection Method, the estimated root r_1^* is given below.

```
Root is 2.304377 after 22 iterations
```

While for the Regula Falsi Method, the corresponding estimated root r_2^* is given below.

```
Root is 2.304377 after 11 iterations
```

From these results, we can conclude that the Regula Falsi Method is faster than the Bisection Method in obtaining the root of the function $f(h)$ given in Equation 1.2.2 since it was able to perform and find a good estimate of the root in only 11 iterations. Observing the final rows of Tables 1.2.1 and 1.2.2, the lengths of the final intervals are 5.72205×10^{-7} and 4.44089×10^{-16} for the Bisection and Regula Falsi methods, respectively. This implies that the estimated root we obtained using the Regula is more accurate and close to the actual value of the root. This conclusion is further supported by evaluating the function values $f(r_1^*)$ and $f(r_2^*)$.
Bisection Method: $f(r_1^*) =$

```
2.172116E-07
```

Regula Falsi Method: $f(r_2^*) =$

```
9.852476E-17
```

We plot the given function and how the two methods are performed graphically to analyze the results we obtain.

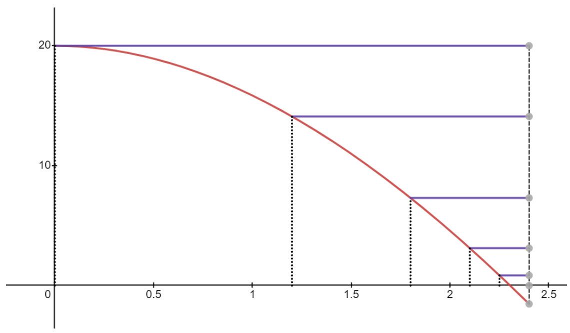

Figure 1.2.2: Plot of the function f over the interval $[0, 2.4]$ applied with Bisection Method.

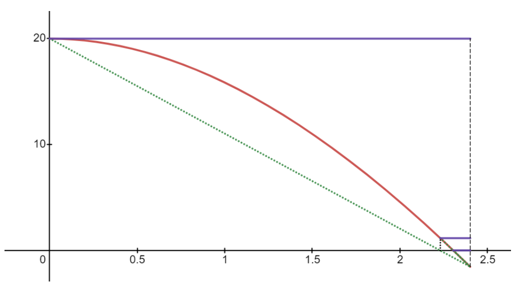

Figure 1.2.3: Plot of the function f over the interval $[0, 2.4]$ applied with Regula Falsi Method.

Notice how the length of the interval (in purple) every each iteration changes for both methods. For the first iteration of the Bisection, the interval length decreases by a factor of $\frac{1}{2}$ while for Regula Falsi, the change factor is higher. The faster decrease in interval length for the Regula

Falsi resulted the method to perform with fewer iterations than Bisection. This is a result of one of the disadvantages of the Bisection Method in which it requires a larger number of iterations to obtain a good result and thus, slower convergence.

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