Week 2: Lecture no 1

Instructor: Amber Shaikh

Root Finding Methods of single Variable equation:

There are two types of numerical methods

1) Bracketing Methods

Bisection Method
False Position Method

2)Open Method

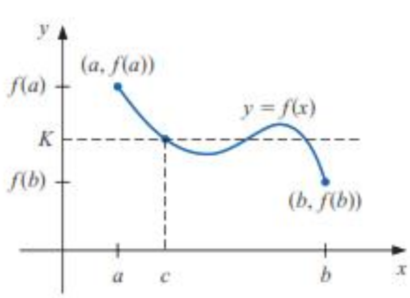
Newton Raphson Method
Fixed Point Iteration Method
Secant Method

1) Bracketing Methods

Bisection Method:
The first technique, based on the Intermediate Value Theorem, is called the Bisection, or Binary-search, method.
According to intermediate value theorem,

If $f \in C[a, b]$ and K is any number between $f(a)$ and $f(b)$, then there exists a number c in (a, b) for which $f(c) = K$.

Figure 1.7 shows one choice for the number that is guaranteed by the Intermediate Value Theorem. In this example there are two other possibilities.



Algorithm for bisection method:

1. Set the interval [a,b] and choose a tolerance level.
2. Compute the midpoint c = (a+b)/2
3. Evaluate f(c)
4. If f(c) = 0 or the interval is within the tolerance, return c as the solution.
5. If f(a)*f(c) < 0, set b=c and go to step 2.
6. If f(b)*f(c) < 0, set a=c and go to step 2.
7. Repeat the steps until the solution is within the desired tolerance.

Stopping Criteria:

$$|p_N - p_{N-1}| < \varepsilon, \tag{2.1}$$

$$\frac{|p_N - p_{N-1}|}{|p_N|} < \varepsilon, \quad p_N \neq 0, \quad \text{or} \tag{2.2}$$

$$|f(p_N)| < \varepsilon. \tag{2.3}$$

Unfortunately, difficulties can arise using any of these stopping criteria. For example, there are sequences $\{p_n\}_{n=0}^{\infty}$ with the property that the differences $p_n - p_{n-1}$ converge to zero while the sequence itself diverges. (See Exercise 17.) It is also possible for $f(p_n)$ to be close to zero while p_n differs significantly from p. (See Exercise 16.) Without additional knowledge about f or p, Inequality (2.2) is the best stopping criterion to apply because it comes closest to testing relative error.

When using a computer to generate approximations, it is good practice to set an upper bound on the number of iterations. This eliminates the possibility of entering an infinite loop, a situation that can arise when the sequence diverges (and also when the program is incorrectly coded). This was done in Step 2 of Algorithm 2.1 where the bound N_0 was set and the procedure terminated if $i > N_0$.

In above paragraph, Algorithm 2.1 is referenced from Burden and Faires book available at GCR.

Another benefit of the bisection method is that the number of iterations required to attain an absolute error can be computed *a priori*—that is, before starting the computation. This can be seen by recognizing that before starting the technique, the absolute error is

Formula For Obtaining No of iterations: $n = \log_2\left(\frac{b-a}{absolute\ error}\right)$

Finding Roots of Equation in Python:

```
from scipy.optimize import fsolve
```

For example, if $f(x) = 2x^3 - x^2 + x - 1$, we have both

$$f(-4) \cdot f(4) < 0 \quad \text{and} \quad f(0) \cdot f(1) < 0,$$

Try to take shortest possible interval

Example

Show that $f(x) = x^3 + 4x^2 - 10 = 0$ has a root in $[1, 2]$, and use the Bisection method to determine an approximation to the root that is accurate to at least within 10^{-4}.

Because $f(1) = -5$ and $f(2) = 14$ the Intermediate Value Theorem 1.11 ensures that this continuous function has a root in $[1, 2]$.

For the first iteration of the Bisection method we use the fact that at the midpoint of $[1, 2]$ we have $f(1.5) = 2.375 > 0$. This indicates that we should select the interval $[1, 1.5]$ for our second iteration. Then we find that $f(1.25) = -1.796875$ so our new interval becomes $[1.25, 1.5]$, whose midpoint is 1.375. Continuing in this manner gives the values in Table 2.1. After 13 iterations, $p_{13} = 1.365112305$ approximates the root p with an error

$$: |1.365234375 - 1.365112305| = 0.000122070.$$

n	a_n	b_n	p_n	$f(p_n)$
1	1.0	2.0	1.5	2.375
2	1.0	1.5	1.25	−1.79687
3	1.25	1.5	1.375	0.16211
4	1.25	1.375	1.3125	−0.84839
5	1.3125	1.375	1.34375	−0.35098
6	1.34375	1.375	1.359375	−0.09641
7	1.359375	1.375	1.3671875	0.03236
8	1.359375	1.3671875	1.36328125	−0.03215
9	1.36328125	1.3671875	1.365234375	0.000072
10	1.36328125	1.365234375	1.364257813	−0.01605
11	1.364257813	1.365234375	1.364746094	−0.00799
12	1.364746094	1.365234375	1.364990235	−0.00396
13	1.364990235	1.365234375	1.365112305	−0.00194

so the approximation is correct to at least within 10^{-4}. The correct value of p to nine decimal places is $p = 1.365230013$. Note that p_9 is closer to p than is the final approximation p_{13}. You might suspect this is true because $|f(p_9)| < |f(p_{13})|$, but we cannot be sure of this unless the true answer is known.

The Bisection method, though conceptually clear, has significant drawbacks. It is relatively slow to converge (that is, N may become quite large before $|p - p_N|$ is sufficiently small), and a good intermediate approximation might be inadvertently discarded. However, the method has the important property that it always converges to a solution, and for that reason it is often used as a starter for the more efficient methods we will see later in this chapter.

Practice Questions:

For doing practice of above topic Use Exercise 2.1 (Numerical Analysis by Burden and Faires 9th Edition)
For matching your answers you can take help from manual solution but keep in mind available manual solution is of 8th Edition so may be solution of some questions won't match.