CSE330(Numerical Methods)

LECTURE 6 – ROOTS OF EQUATIONS

[BRACKETING METHOD: BISECTION METHOD]

INSTRUCTOR: NAHIAN IBN HASAN

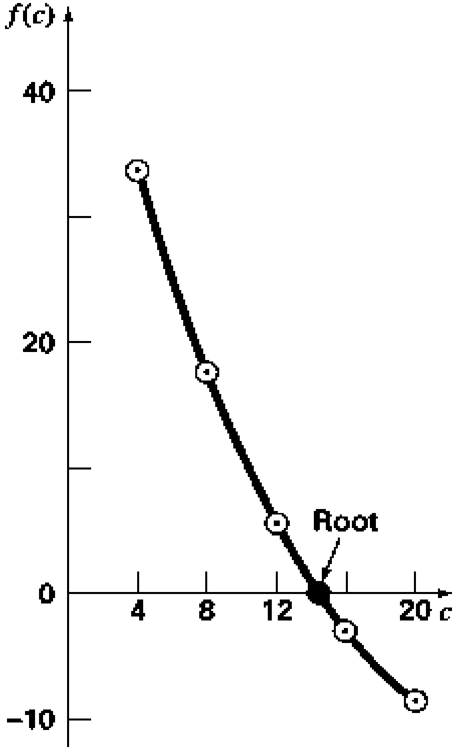
LECTURER, DEPARTMENT OF EEE,

BRAC UNIVERSITY, DHAKA, BANGLADESH

Overview

- In graphical method, we have seen that, a continuous function changes sign from positive to negative or (vice versa) at the vicinity of a root.

- In general, if f(x) is real and continuous in the interval from x_l to x_u and $f(x_l)$ and $f(x_u)$ have opposite signs, that is, $f(x_l).f(x_u) < 0$, then there is at least one real root between x_l and x_u.



- The bisection method is alternatively called **binary chopping**, **interval halving**, or **Bolzano's method**.

- It is one type of incremental search method in which the interval is always divided in half. If a function changes sign over an interval, the function value at the midpoint is evaluated. The location of the root is then determined as lying at the midpoint of the subinterval within which the sign change occurs. The process is repeated to obtain refined estimates.

Bisection Method: Algorithm

Step 1 :

Choose lower x_l and upper x_u guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that $f(x_l) \cdot f(x_u) < 0$. Also, you can take help using graphical approach from lesson 5.

Step 2:

An estimate of the root x_r is determined by $x_r = \frac{x_l + xu}{2}$

Step 3:

Make the following evaluations to determine in which subinterval the root lies:

a) If $f(x_l).f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.

b) If $f(x_l).f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.

c) If $f(x_l).f(x_r) = 0$, the root equals x_r; terminate the computation.

Bisection Method: Termination Criteria

- The step 3 of bisection method mentions the termination criteria. But, the code only terminates when the x_r exactly overlaps with the true value. Otherwise, the code will continue until the computer memory is fully consumed.

- Hence, we always calculate the approximate error percentage in every iteration using the **percentage approximate relative error** from lesson 2.

$$\epsilon_a = \left| \frac{x_r^{new} - xrold}{x_r^{new}} \right| \times 100\%$$

Here, x_r^{new} is the root for the present iteration, and x_r^{old} is the root for the previous iteration. When ϵ_a is lower than a pre-specified tolerance limit of ϵ_s , then the iteration stops and x_r^{new} is considered as the root.

- In the exam, you will not have access to any computer, hence, you will have to try for various values of x_l and x_u initially, so that $f(x_l) \cdot f(x_u) < 0$.

Bisection Method: Example

Example 1:

Find out the root of the equation f(x) = x^2 - 8x + 12 using bisection method.

Solution:

Iteration 1:

- Initially we need to take two approximate values x_l and x_u. Say, x_l = 1 and x_u = 4.
- Hence, $f(x_l)$ = 1 – 8 + 12 = 5 and $f(x_u)$ = 16 – 8*4 + 12 = -4. Since, $f(x_l)$. $f(x_u)$ < 0, our approximation is correct. So, we have a root between these two points [1,4].
- Now, new approximated root, x_r = (1+4)/2 = 2.5;
- $f(x_r)$ = f(2.5) = -1.75
- $f(x_l)$. $f(x_r)$ = 5 * -1.75 = -8.75 < 0. Hence, the root is in the lower half. So, we set x_u = x_r = 2.5

Bisection method: Example

Iteration 2:

- Now, after first iteration, we have $x_l = 1$ and $x_u = 2.5$.
- Hence, $f(x_l) = f(1) = 5$ and $f(x_u) = f(2.5) = -1.75$.
- New approximated root, $x_r = (1+2.5)/2 = 1.75$;
- $f(x_r) = f(1.75) = 1.0625$
- $\epsilon_a = |(x_r^{new} - x_r^{old})/x_r^{new}| \times 100\% = |(1.75 - 2.5)/1.75| \times 100\% = 42.857\%$
- $f(x_l) \cdot f(x_r) = 5 * 1.0625 = 5.3125 > 0$. Hence, the root is in the upper half. So, we set $x_l = x_r = 1.75$

Iteration 3:

- After second iteration, we have $x_l = 1.75$ and $x_u = 2.5$.
- Hence, $f(x_l) = f(1.75) = 1.0625$ and $f(x_u) = f(2.5) = -1.75$.
- New approximated root, $x_r = (1.75+2.5)/2 = 2.125$;
- $f(x_r) = f(2.125) = -0.484375$
- $\epsilon_a = |(x_r^{new} - x_r^{old})/x_r^{new}| \times 100\% = |(2.125 - 1.75)/2.125| \times 100\% = 17.6\%$
- $f(x_l) \cdot f(x_r) = 1.0625 * -0.484375 = -0.5146484 < 0$. Hence, the root is in the lower half. So, we set $x_u = x_r = 2.125$

Bisection Method: Example

Iteration 4:

- After third iteration, we have $x_l = 1.75$ and $x_u = 2.125$.
- Hence, $f(x_l) = f(1.75) = 1.0625$ and $f(x_u) = f(2.125) = -\ 0.484375$
- New approximated root, $x_r = (1.75+2.125)/2 = 1.9375$;
- $f(x_r) = f(1.9375) = 0.25390625$
- $\epsilon_a = |(x_r^{new} - x_r^{old})/x_r^{new}| \times 100\% = |(1.9375 - 2.125)/1.9375| \times 100\% = 9.677\%$
- $f(x_l) \,.\, f(x_r) = 1.0625 * 0.25390625 = 0.26977539 > 0$. Hence, the root is in the upper half. So, we set $x_l = x_r = 1.9375$

Iteration 5:

- After fourth iteration, we have $x_l = 1.9375$ and $x_u = 2.125$.
- Hence, $f(x_l) = f(1.9375) = 0.25390625$ and $f(x_u) = f(2.125) = -\ 0.484375$
- New approximated root, $x_r = (1.9375+2.125)/2 = 2.03125$
- $f(x_r) = f(2.03125) = -0.124023437$
- $\epsilon_a = |(x_r^{new} - x_r^{old})/x_r^{new}| \times 100\% = |(2.03125 - 1.9375)/2.03125| \times 100\% = 4.615\%$
- $f(x_l) \,.\, f(x_r) = 0.25390625 * -0.124023437 = -0.031490325 < 0$. Hence, the root is in the lower half. So, we set $x_u = x_r = 2.03125$

Bisection Method: Example

Result:

- Observe the value of approximate error (ϵ_a). The error decreases with increasing iterations. If we continue with more iterations, we will see that the error decreases further. In our code, we will set a pre-specified value ϵ_s . When the error ϵ_a gets lower than ϵ_s , we would consider the last iteration's x_r as our result. For example, if ϵ_s = 5%; then we see in the 5th iteration our error (ϵ_a) is lower (4.615%). Hence, we consider x_r in the 5th iteration (2.03125) as our final root.

- Obviously, if we want more precision, we would set the ϵ_s at lower value. Then more iterations will be needed. But we will have better approximation to the real root.

Bisection Method: Limitations

- In the example, the function is of second order. Hence, there should be two roots. But in bisection method, we get one root only.
- To find out the other root, we need to go through the whole process again but with different approximations of x_l and x_u.
- We have to be careful in choosing new x_l and x_u. Since, we might accidentally converge to the same root again, which is really very troublesome. In this respect graphical approximation from **lesson 5** might be useful.
- Through bisection method, we cannot determine similar roots (i.e. when the function is tangent to the x-axis).
- A shortcoming of the bisection method is that, in dividing the interval from x_l to x_u into equal halves, no account is taken of the magnitudes of $f(x_l)$ and $f(x_u)$.

Conclusion

In this lesson we have

- learnt the algorithm of bisection method for finding roots of a function
- implemented the bisection algorithm for solving a function practically
- known about the limitations of bisection algorithm

- Please try out the MATLAB code given with this lesson implementing the bisection algorithm. Run the code for different functions and find out the root. Try to change the pre-specified error limit and see how the accuracy improves.

In the next lesson we will learn another bracketing method of finding out roots of a function, named 'False Position Method'

Thank You