CSC349A Numerical Analysis

Lecture 8

Rich Little

University of Victoria

2023

Table of Contents I

Roots or Zeros

Definition

If we have an equation with one variable: $f(x) = 0$ then a value $\hat{x}$ for which this equation holds is called a *root* of the equation or a *zero* of the function $f(x)$.

Families of functions for root finding

There are two main families which we will consider.

- Polynomials such as $f(x) = x^3 + x^2 + x + 1$ which are a special case will be treated separately. For polynomials we will find both real and complex roots.
- More general, non-polynomial algebraic functions and the so called transcedental functions, such as $f(x) = e^{-x} - x = 0$ will be what we will cover first and we will focus on finding one or more real roots.
- Although in some cases it is possible to derive the *root* analytically, in many cases this is impossible in which case our only option is to use numerical methods.

Usage of root finding

In practical engineering problems, frequently there are cases where we can not rearrange an equation so that the unknown quantity is on one side of the equation and all the known quantities are on the other side. For example consider the equation describing the free fall of the parachutist that we used to motivate numerical methods:

$$v(t) = \frac{gm}{c}(1 - e^{-ct/m}) \tag{1}$$

Suppose that instead of being given m, c, t and computing v, you want to know the value of c (drag coefficient) for a parchutist of mass m to attain a certain velocity after falling for time t.

Formulating as root finding

It is impossible to rearrange this equation so that c is on one side and all the known quantities are on the other. However it is trivial to express it as follows:

$$\frac{gm}{c}(1 - e^{-ct/m}) - v = 0 \tag{2}$$

where g,m,t,v are known. Now the task is to find the value of c that makes this equation equal to 0 i.e solve $f(c) = 0$. In this case there is no analytic solution and we have to use numerical root finding methods such as the ones we will cover in this course.

Numerical iteration

All numerical methods are iterative i.e given one or more initial approximations to a root x_t they compute a sequence of approximations.

$$lim_{i \to \infty} x_i = x_t \tag{3}$$

In the 1830s it was proved that there are no finite algorithms involving $+, -, *, /$ for computing the roots of polynomials of degree n if $n \geq 5$.

Table of Contents I

Bisection and bracketing

The **Bisection** method can be used to compute a zero of any function $f(x)$ that is continuous on an interval $[x_l, x_u]$ for which $f(x_l) \times f(x_u) < 0$.

Consider x_l and x_u as *two initial approximations* to a zero, say x_t, of $f(x)$. The new approximation is the midpoint of the interval $[x_l, x_u]$ which is $x_r = \frac{x_l + x_u}{2}$.

The algorithm

- If $f(x_r) = 0$, the x_r is the desired zero of $f(x)$. Otherwise, a new interval $[x_l, x_u]$ that is half the length of the previous interval is determined as follows.
- If $f(x_l) \times f(x_r) < 0$ then $[x_l, x_r]$ contains a zero, so set $x_u \leftarrow x_r$. Otherwise $f(x_u) \times f(x_r) < 0$ (necessarily) and $[x_r, x_u]$ contains a zero, so set $x_l \leftarrow x_r$.

The above procedure is repeated, continually halving the interval $[x_l, x_u]$, until $[x_l, x_u]$ is sufficiently small, at which time the midpoint $x_r = \frac{x_l + x_u}{2}$ will be arbitrarily close to a zero of $f(x)$.

Iterarions n for desired accuracy

Suppose you want the *absolute error* $< \varepsilon$, and that the length of the initial interval $[x_l, x_u]$ is Δx^0.

approximation	absolute error
$x_1 = \frac{x_l + x_u}{2}$	$\lvert x_t - x_1 \rvert \leq \frac{\Delta x^0}{2}$
x_2	$\lvert x_t - x_2 \rvert \leq \frac{\Delta x^0}{4}$
x_3	$\lvert x_t - x_3 \rvert \leq \frac{\Delta x^0}{8}$
$\ldots$	$\ldots$
x_n	$\lvert x_t - x_n \rvert \leq \frac{\Delta x^0}{2^n}$

Table: Relation of approximation and absolute error

Relating n and ε

Therefore $\frac{\Delta x^0}{2^n} < \varepsilon$ implies that $2^n > \frac{\Delta x^0}{\varepsilon}$ and $n > log_2\left(\frac{\Delta x^0}{\varepsilon}\right)$

or

$$n \ln 2 > \ln\left(\Delta x^0\right) - \ln(\varepsilon) \quad \text{and} \quad n > \frac{\ln\left(\Delta x^0\right) - \ln\left(\varepsilon\right)}{\ln 2}$$

Example 1

Find the positive root of $f(x) = x^2 - 3$ such that $|E_t| < 0.01$. Note that the true root is $x_t = \sqrt{3} \approx 1.73205$ so we will start with $x_l = 1$ and $x_u = 2$.

Example 1 continued

Convergence Criterion

As this is an iterative algorithm that computes a sequence of approximations:

$$x_0, x_1, x_2, \ldots, x_{i-1}, x_i, \ldots$$

to a zero x_t, recall that we can use the iterative approximation relative error:

$$|\varepsilon_a| = \left|\frac{x_i - x_{i-1}}{x_i}\right| = \left|1 - \frac{x_{i-1}}{x_i}\right|$$

is a good approximation to the actual relative $|\varepsilon_t|$ in x_i, and can be used to determine the accuracy of x_i.

Approximation error

Note that each approximation x_i is equal to $\frac{x_u+x_l}{2}$, and the previous approximation is either x_l or x_u. Therefore:

$$\left|x_i - x_{i-1}\right| = \left|\frac{x_l + x_u}{2} - \frac{2x_l}{2}\right| = \left|\frac{x_u - x_l}{2}\right|$$

thus

$$\left|\varepsilon_a\right| = \frac{\left|x_i - x_{i-1}\right|}{\left|x_i\right|} = \frac{\left|\frac{x_u - x_l}{2}\right|}{\left|\frac{x_u + x_l}{2}\right|} = \left|\frac{x_u - x_l}{x_u + x_l}\right|$$

Functions as arguments and global variables

- Numerical methods frequently are expressed as iterations that require you to evaluate a function (and sometimes its derivatives) for multiple points.
- Examples include Euler's method which requires the evaluation of the slope in the incremental update and the bisection method that requires the evaluation of the function at the boundaries and midpoint of the interval under consideration in each iteration
- From a programming perspective in all these cases we would like to abstract the numerical algorithm (Euler's method, Bisection, Newton/Raphson) from the specific function being evaluated.
- In the same way that we can generalize a function by adding a parameter or argument we would like to generalize our methods to arbitrary functions.

Bisection example using MATLAB

The pseudocode algorithm for the Bisection method can be found in Handout 8. I have implemented it in MATLAB in order to solve problem 5.17 (page 143) of the textbook.

Example The volume of liquid in a spherical tank is given by

$$V = \frac{\pi h^2(3R - h)}{3}$$

where h is the depth of the liquid and R is the radius. If $R = 3$, to what depth must the tank be filled so that it contains $V = 30m^3$ of water?

Why Bisection?

University of Victoria Computer Science

Advantages

- If $f(x)$ is continuous and if appropriate initial values x_l and x_u can be found, then the method is **guaranteed to converge**.

Disadvantages

- converges slowly (requires more iterations than other methods)
- it may be difficult to find appropriate initial values
- it cannot be used to compute a zero x_t of **even multiplicity** of a function $f(x)$; that is, if

$$f(x) = (x - x_t)^m g(x)$$

where m is a positive even integer and $g(x_t) \neq 0$