MATH 4700 Numerical Analysis I - Notes

Developing a Hybrid Root-finding Algorithm

Background

Numerical methods, such as root finding algorithms, have both advantages and disadvantages. Some methods are steady and reliable, but convergence rate may be slow. Bisection method is one such algorithm. It guarantees to converge as long as the starting interval contains the root, however, the convergence rate is only linear. Other root-finding algorithms, such as Newton's method or Secant method, provide faster rate of convergence, but may not be so reliable. Their convergence heavily depends on the starting point, which may be chosen arbitrarily [1, 2].

In an attempt to construct a fast and reliable algorithm, one may combine features that make Bisection method reliable with features that make Newton's or Secant method fast. Such hybrid algorithm will work from any starting point or starting interval with eventual fast convergence.

The next sections will give a review of Bisection and Secant methods as we have learned in the lecture, and a discussion on how to combine these two algorithms.

Bisection method

Recall the main idea of the Bisection method as discussed in the lecture: It starts with an initial interval $[a_0, b_0]$ that contains the root of $f(x)$. The interval is halved at each iteration. The subinterval that does not contain the root is omitted, while the other half will be carried on to the next iteration. We call this interval *the bracketing interval*. After sufficient number of iterations have been performed, the length of the bracketing interval will be small, and the midpoint of this interval is taken to be the approximation to the actual root. The outline of the Bisection method is as follows [2]:

Given an initial interval $[a_0, b_0]$ that contains the root. For $k = 0, 1, 2, ...$

1. Compute the midpoint $c_{k+1} = a_k + \frac{1}{2}(b_k - a_k)$.
2. If $f(c_{k+1})f(a_k) < 0$, set $a_{k+1} = a_k$, $b_{k+1} = c_{k+1}$.
3. If $f(c_{k+1})f(b_k) < 0$, set $b_{k+1} = b_k$, $a_{k+1} = c_{k+1}$.
4. Update k, and repeat the above steps.

Using the fact that the length of the subsequent interval is half of the length of the current interval, one can derive the error formula for the Bisection method

$$|\alpha - c_k| \leq (1/2)^k (b - a),$$

where α is the actual root and c_k is the kth approximation to the root (or the midpoint of the kth interval).

Secant method

Secant method can be seen as one of the variants of Newton's method. Instead of using the tangent line approximation that requires the computation of the derivative $f'(x_k)$, Secant method uses secant line that passes through the previous two iterates, x_k and x_{k-1}. By replacing the derivative term on the Newton's method formula by

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

we arrive at the Secant method formula:

$$x_{k+1} = x_k - f(x_k)\Big[\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}\Big].$$

Hybrid method

To obtain higher convergence rate, the hybrid method uses Secant method whenever possible, and only switches to Bisection when the Secant iterate is moving further away from the root. It begins with calculating the Secant iterate using the endpoints of the initial interval. The algorithm keeps performing Secant method as long as the iterates produced are within the bracketing interval. At each iteration, the Secant iterates are used to update the bracketing interval so that it shrinks as fast as possible. When a Secant iterate falls outside the bracketing interval, the algorithm switches to Bisection method for that particular iteration. The outline of the hybrid algorithm can be written as follows:

Given an initial interval $[a_1, b_1]$ that contains the root.
Set $k = 1, x_0 = a_1, x_1 = b_1$ and perform the following:

1. Compute the Secant iterate c using x_{k-1} and x_k.
2. If $c < a_k$ or $c > b_k$, then perform Bisection method, update the bracketing interval $[a_{k+1}, b_{k+1}]$, and update x_k and x_{k+1}.

   Else update bracketing interval using Secant iterate c.
3. Set $k = k + 1$ and repeat steps 1-3.

References

[1] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th. ed., Cengage Learning, 2010.

[2] J. F. Epperson, *An Introduction to Numerical Methods and Analysis*, 2nd. ed., Wiley, 2013.