ITERATIVE METHODS FOR SIMPLE ROOTS

August 30, 2021

1 Introduction

We consider the methods for determining the roots of the equation

$$f(x) = 0 \tag{1}$$

which may be given explicitly as a polynomial of degree n in x or $f(x)$ may be defined as a transcendental function.
There are two types of methods that can be used to find the roots of the equation (1).

1. *Direct methods* : These methods give the exact value of the roots in a finite number of steps. These methods determine all the roots at the same time.

2. *Iterative methods* : These methods are based on the idea of successive approximations. Starting with one or more initial approximations to the root, we obtain a sequence of iterates x_k which in the limit converges to the root. These methods determine one or two roots at a time.

2 Iterative methods for simple roots

Definition : A root ξ is called a simple root of $f(x) = 0$, if $f(\xi) = 0$ and $f^{'}(\xi) \neq 0$. Then, we can also write$f(x) = (x - \xi)g(x)$, where $g(x)$ is bounded and $g(\xi) \neq 0$.

2.1 Bisection Method

If the function $f(x)$ satisfies $f(a_0)f(b_0) < 0$, then the equation $f(x) = 0$ has atleast one real root or an odd number of real roots in the interval (a_0, b_0). If $m_1 = \frac{1}{2}(a_0 + b_0)$ is the mid point of this interval, then the root will lie either in the interval (a_0, m_1) or in the interval (m_1, b_0) provided that $f(m_1) \neq 0$. If $f(m_1) = 0$, then m_1 is the required root. Repeating this procedure a number of times, we obtain the bisection method

$$m_{k+1} = a_k + \frac{1}{2}(b_k - a_k), \quad k = 0, 1, \ldots$$

where

$$(a_{k+1},\ b_{k+1}) = \begin{cases} (a_k,\ m_{k+1}), & \text{if } f(a_k)f(m_{k+1}) < 0 \\ (m_{k+1},\ b_k), & \text{if } f(m_{k+1})f(b_k) < 0 \end{cases}$$

We take the midpoint of the last interval as an approximation to the root. This method always converges, if $f(x)$ is continuous in the interval $[a,\ b]$ which contains the root.

Algorithm

1. Choose initial guesses x_0 and x_1 such that $f(x_0)f(x_1) < 0$
2. Choose pre-specified tolerable error ϵ.
3. Calculate new approximated root as $x_2 = \frac{(x_0+x_1)}{2}$
4. Calculate $f(x_0)f(x_2)$
   (a) if $f(x_0)f(x_2) < 0$ then $x_0 = x_0$ and $x_1 = x_2$
   (b) if $f(x_0)f(x_2) > 0$ then $x_0 = x_2$ and $x_1 = x_1$
   (c) if $f(x_0)f(x_2) = 0$ then goto (8)
5. if $|f(x_2)| > \epsilon$ then goto (3) otherwise goto (6)
6. Display x_2 as root.
7. Stop.

2.2 Secant Method

In this method, we approximate the graph of the function $y = f(x)$ in the neighbourhood of the root by a straight line (secant) passing through the points $(x_{k-1},\ f_{k-1})$ and (x_k, f_k), where $f_k = f(x_k)$ and take the point of intersection of this line with the x-axis as the next iterate. We thus obtain

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f_k - f_{k-1}} f_k,\ \ k = 1, 2, \ldots$$

or

$$x_{k+1} = \frac{x_{k-1}\ f_k - x_k\ f_{k-1}}{f_k - f_{k-1}},\ \ k = 1, 2, \ldots$$

where x_{k-1} and x_k are two consecutive iterates. In this method, we need two initial approximations x_0 and x_1. This method is also called the *chord* method. The order of the method is obtained as

$$p = \frac{1}{2}(1 + \sqrt{5}) \approx 1.62$$

If the approximations are chosen such that $f(x_k{-}1)f(x_k) < 0$ for each k, then the method is known as *Regula-Falsi* method and has linear (first order) rate of convergence. Both these methods require one function evaluation per iteration.

Algorithm

1. Start
2. Define function as $f(x)$
3. Input initial guesses x_0 and x_1, tolerable error (ϵ) and maximum iteration (N)
4. Initialize iteration counter $i = 1$
5. If $f(x_0) = f(x_1)$ then print "Mathematical Error" and goto (11) otherwise goto (6)
6. Calcualte $x_2 = x_1 - \frac{(x_1 - x_0)}{f(x_1) - f(x_0)} * f(x_1)$
7. . Increment iteration counter $i = i + 1$
8. If $i >= N$ then print "Not Convergent" and goto (11) otherwise goto (9)
9. If $|f(x_2)| > \epsilon$ then set $x_0 = x_1,\ x_1 = x_2$ and goto (5) otherwise goto (10)
10. Print root as x_2
11. Stop

2.3 Newton-Raphson method

In this method, we approximate the graph of the function $y = f(x)$ in the neighbourhood of the root by the tangent to the curve at the point (x_k, f_k) and take its point of intersection with the x-axis as the next iterate. We have the Newton-Raphson method as

$$x_{k+1} = x_k - \frac{f_k}{f'_k}, \quad k = 0, 1, \ldots$$

and its order is $p = 2$. This method requires one function evaluation and one first derivative evaluation per iteration.

Algorithm

1. Define function as $f(x)$
2. Define first derivative of $f(x)$ as $g(x)$
3. Input initial guess (x_0), tolerable error (ϵ) and maximum iteration (N)
4. Initialize iteration counter $i = 1$
5. If $g(x_0) = 0$ then print "Mathematical Error" and goto (11) otherwise goto (6)
6. Calcualte $x_1 = x_0 - \frac{f(x_0)}{g(x_0)}$
7. Increment iteration counter $i = i + 1$
8. If $i >= N$ then print "Not Convergent" and goto (11) otherwise goto (9)
9. If $|f(x_1)| > \epsilon$ then set $x_0 = x_1$ and goto (5) otherwise goto (10)
10. Print root as x_1
11. Stop

Example: Find the interval of equation $x^3 - x - 4 = 0$. in which the smallest positive root lies and also, determine the roots correct to two decimal places using the bisection method.

Solution:
We find $f(0) = -4,\ f(1) = -4,\ f(2) = 2$.
Therefore, the root lies in the interval $(1,\ 2)$. The sequence of intervals using the bisection method is obtained as

k	a_{k-1}	b_{k-1}	m_k	$f(m_k)f(a_{k-1})$
1	1	2	1.5	> 0
2	1.5	2	1.75	> 0
3	1.75	2	1.875	< 0
4	1.75	1.875	1.8125	> 0
5	1.75	1.8125	1.78125	> 0
6	1.78125	1.8125	1.796875	< 0
7	1.78125	1.796875	1.7890625	> 0
8	1.7890625	1.796875	1.792969	> 0
9	1.792969	1.796875	1.794922	> 0
10	1.794922	1.796875	1.795898	> 0.

After 10 iterations, we find that the root lies in the interval $(1.795898,\ 1.796875)$. Therefore, the approximate root is $m = 1.796387$. The root correct to two decimal places is 1.80.

Example: Given the following equation : $x^4 - x - 10 = 0$, determine the initial approximations for finding the smallest positive root. Use these to find the root correct to three decimal places with the following methods:
(a) Secant method, (b) Regula-Falsi method, (c) Newton-Raphson method.
Solution:
We find that $f(0) = -10,\ f(1) = -10,\ f(2) = 4.$
Hence, the smallest positive root lies in the interval $(1,\ 2)$.
The Secant method gives the iteration scheme

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f_k - f_{k-1}} f_k,\ \ k = 1, 2, \ldots$$

With $x_0 = 1,\ x_1 = 2$, we obtain the sequence of iterates

$$x_2 = 1.7143,\ x_3 = 1.8385,\ x_4 = 1.8578,\ x_5 = 1.8556,\ x_6 = 1.8556.$$

The root correct to three decimal places is 1.856.

The Regula-Falsi method gives the iteration scheme

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f_k - f_{k-1}} f_k,\ \ k = 1, 2, \ldots$$

and $f_k f_{k-1} < 0$.
With $x_0 = 1,\ x_1 = 2$, we obtain the sequence of iterates

$$\begin{array}{lll}
x_2 = 1.7143, & f(x_2) = -3.0776, & \xi \in (x_1,\ x_2), \\
x_3 = 1.8385, & f(x_3) = -0.4135, & \xi \in (x_1, x_3), \\
x_4 = 1.8536, & f(x_4) = -0.0487, & \xi \in (x_1, x_4), \\
x_5 = 1.8554, & f(x_5) = -0.0045, & \xi \in (x_1, x_5), \\
x_6 = 1.8556. & &
\end{array}$$

The root correct to three decimal places is 1.856.

The Newton-Raphson method gives the iteration scheme

$$x_{k+1} = x_k - \frac{f_k}{f'_k}, \quad k = 0, 1, \ldots$$

With $x_0 = 2$, we obtain the sequence of iterates

$$x_1 = 1.8710,\ x_2 = 1.8558,\ x_3 = 1.8556.$$

Hence, the root correct to three decimal places is 1.856.

2.4 Fixed point Iteration

Fixed point : A point, say, s is called a fixed point if it satisfies the equation $x = g(x)$.

Fixed point Iteration : The transcendental equation $f(x) = 0$ can be converted algebraically into the form $x = g(x)$ and then using the iterative scheme with the recursive relation

$$x_{i+1} = g(x_i), \qquad i = 0, 1, 2, \ldots,$$

with some initial guess x_0 is called the fixed point iterative scheme.
Algorithm : Given an equation $f(x) = 0$

1. Convert $f(x) = 0$ into the form $x = g(x)$
2. Let the initial guess be x_0
3. Compute
$$x_{i+1} = g(x_i), \qquad i = 0, 1, 2, \ldots,$$
4. Repeat till $|x_{i+1} - g(x_i)| \leq \epsilon$ (where i is the iteration number)

Example : Find a root of $x^4 - x - 10 = 0$
Solution :
Consider another function $g(x) = (x + 10)^{1/4}$ and the fixed point iterative scheme

$$x_{i+1} = (x_i + 10)^{1/4}, \qquad i = 0, 1, 2, \ldots$$

let the initial guess x_0 be 1.0, 2.0 and 4.0

i	0	1	2	3	4	5	6
x_i	1.0	1.82116	1.85424	1.85553	1.85558	1.85558	
x_i	2.0	1.861	1.8558	1.85559	1.85558	1.85558	
x_i	4.0	1.93434	1.85866	1.8557	1.85559	1.85558	1.85558