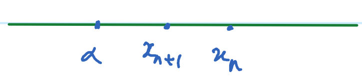
Lecture 13

Wednesday, February 5, 2020

We know that the bisection method has an a priori estimate: given an allowed error $\varepsilon > 0$, we know the number of bisection steps needed to get a root with error under ε. In particular,

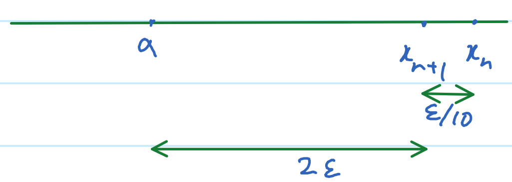
$$n \geqslant \log_2\left(\frac{b_0 - a_0}{\varepsilon}\right) - 1$$

Newton's method lacks an a priori estimate because it doesn't guarantee the convergence of sequence x_n. In particular, we don't know beforehand the number of iterations needed to get an estimate root with error under ε. However, one can impose an artificial condition to stop the iteration such as: $|x_{n+1} - x_n| < \frac{\varepsilon}{10}$.



We can stop the iteration (at step n) whenever the difference between x_n and x_{n+1} is less than $\varepsilon/10$

Note that the number 10 is not a special number. One can replace it by some number larger than 1. At the step n where we stop, strictly speaking we may not have $|x_n - \alpha| < \varepsilon$ as shown in the picture.

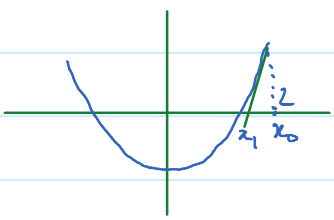


However, it is reasonable to hope that: when x_{n+1} is close to x_n (the difference is less than $\varepsilon/10$), then the sequence $x_n, x_{n+1}, x_{n+2}, \dots$ is close to the limit α (the true root). We know that the sequence converges quickly to α (with order of convergence $p=2$) Each term $x_0, x_1, x_2, \dots$ is "corrected" quickly by the next term. When x_{n+1} is close x_n, it is perceivable that there is not much correction needed from x_{n+1} to x_{n+2}, from x_{n+2} to x_{n+3}, and so on.

We want to implement the following procedure on Matlab:

1) Given an initial point x_0, a function f, and a number ε.
2) Use the iteration formula
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$
to evaluate $x_1, x_2, x_3, \dots$ consecutively
3) Stop at index n where $|x_n - x_{n+1}| < \varepsilon/10$.
4) Output is x_{n+1}.

How to do so? Let's consider an example where $f(x) = x^2 - 3$, $x_0 = 2$, $\varepsilon = 10^{-3}$.



The iteration formula is
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - 3}{2x_n}$$
$$= \frac{x_n}{2} + \frac{3}{2x_n}$$

We don't need to store the whole array $x_0, x_1, x_2, \dots$ We only need to store two variables u (for x_n) and v (for x_{n+1}) These two variables will be updated after each step

```
ε = 0.001;
u = 2;                  ← this is x0
v = u/2 + 3/2/u ;       ← this is x1
while (abs(u-v) >= ε/10) ← get out of the loop when |u-v| < ε/10
    u = v ;             ← update u
    v = v/2 + 3/2/v;    ← update v
end
v
```

[Further example on order of convergence is on the worksheet.]