Topics Covered

Introduction

Newton's method is a technique that consists of finding a root of a non linear equation $f(x) = 0$ on a interval $[a, b]$, where f is a two times differentiable function. Newton's method is based on Taylor's approximation and has a nice geometric interpretation.

Derivation of the method

Let $f(x)$ a two times differentiable function on $[a, b]$ and we are looking for a root p of $f(x)$, that is, p such that $f(p) = 0$. Let p_0 an initial guess such that $f'(p_0) \neq 0$. The Taylor approximation of $f(x)$ around p_0 is:

$$f(x) = f(p_0) + f'(p_0)(x - p_0) + \frac{f''(p_0)}{2}(x - p_0)^2 + \frac{f'''(x)}{6}(x - p_0)^3 + \dots$$

Evaluated at p, we obtain:

$$0 = f(p_0) + f'(p_0)(p - p_0) + \frac{f''(p_0)}{2}(p - p_0)^2 + \frac{f'''(p)}{6}(x - p_0)^3 + \dots$$

By neglecting the terms of order higher or equal than 2, we obtain:

$$p \approx p_0 - \frac{f(p_0)}{f'(p_0)} := p_1$$

Hence, from the initial guess p_0, Netwon's method generates a sequence $(p_n)_{n\geq 0}$ defined by

$$p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}$$

Algorithm 1: Netwon's method

Require: A function function f, atolerance ε, max # of iteration N and an initial guess p_0.

```
for all n such that 0 ≤ n ≤ N do
    p_{n+1} = p_n − f(p_n)/f'(p_n)
    if |p_{n+1} − p_n| ≤ ε then
        Stop
    end if
end for
if n=N then
    The method failed after N iterations.
else
    The method converges after n + 1 iterations and p_{n+1} is the root of f(x).
end if
```

Example 1

Write a Matlab program that implements Newton's method in finding the root of the function $f(x) = e^x - x$, going from $x_0 = 0$ and a tolerance of $\varepsilon = 10^{-10}$.

#1

Analysis of Convergence and Geometrical Interpretation

Newton's method is particular case of the fixed-point method and therefore its convergence is guaranteed by the fixed-point theorem.

Secante Method

Although Netwon's method has many advantages, it necessitates to compute the derivative $f'(x)$ of $f(x)$. If the function f is complex, computing its derivative might be also complex. To avoid this difficulty, we replace the slope $f'(p_n)$ at each iteration by its approximate:

$$f'(p_n) = \frac{f(p_n) - f(p_{n-1})}{p_n - p_{n-1}}$$

This is equivalent of replacing of the tangent line (at p_n) in Newton's method by the secant line passing by $(p_{n-1}, f(p_{n-1}))$ and $(p_n, f(p_n))$:

$$y = f(p_n) + \frac{f(p_n) - f(p_{n-1})}{p_n - p_{n-1}}(x - p_n)$$

For example, starting form x_0 and x_1, we have:

$$y = f(p_1) + \frac{f(p_1) - f(p_0)}{p_1 - p_0}(x - p_1)$$

This line intersect the $x-$axis when $y = 0$. That is:

$$p_2 = p_1 - \frac{f(p_1)(p_1 - p_0)}{p_1 - p_0}$$

We repeat the process with $(p_1, f(p_1))$ and $(p_2, f(p_2))$. We generate therefore the sequence $(p_n)_{n\geq 0}$ defined by the relationship:

$$p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{p_n - p_{n-1}}.$$

Algorithm 2: Secant method

Require: A function function f, atolerance ε, max # of iteration N and two initial guesses p_0 and p_1.

for all n such that $0 \leq n \leq N$ **do**
  $p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{p_n - p_{n-1}}$
  if $|p_{n+1} - p_n| \leq \varepsilon$ **then**
    Stop
  end if
end for
if n=N **then**
  The method failed after N iterations.
else
  The method converges after n iterations and p_{n+1} is the root of $f(x)$.
end if

Example 2

Repeat Example 1 with the Secant Method. Which method converges faster?

#2