APPM 4650 — Newton's method for functions

Consider $f(x) \in C^2[a,b]$ with a root $c \in [a,b]$. Note $C^1[a,b]$ means that the function $f(x)$, $f'(x)$ and $f''(x)$ are continuous on the interval $[a,b]$. Our goal is to use this additional smoothness in the function to create a technique for finding the root p. The idea is to follow the roots of tangent lines until the method converges. In other words, to derive Newton's method, we will utilize Taylor's approximations.

Derivation:
Suppose $f \in C^2[a,b]$. Let $p_0 \in [a,b]$ such that $f'(p_0) \neq 0$ be our initial guess for the root. We will assume that p_0 is close to the root p; i.e. $|p - p_0|$ is "small."
Then by Taylor we know that

$$f(p) = f(p_0) + (p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2} f''(\eta)$$

for some η in the interval between p and p_0.
We are only going to approximate f locally by its tangent line.

$$f(p) \approx f(p_0) + (p - p_0)f'(p_0)$$

If p is the root of f, then it is also the root of the tangent line and we write p in terms of p_0, etc.

$$\begin{aligned} 0 &= f(p_0) + (p - p_0)f'(p_0) \\ p &= p_0 - \frac{f(p_0)}{f'(p_0)} \end{aligned}$$

In general p is not the root of f but it is always the root of the tangent line. Thus we can use this to create our fixed point iteration.

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})} \quad \text{for } n \geq 1$$

Draw a picture of an iteration

We can now create a Pseudocode for Newton's method.

PSEUDOCODE: Newton's method

Input: p_0 = initial guess, ϵ =tolerance, $N_{\max}$= max number of iterations
$f(x)$ and $f'(x)$
Output: p^*= the approximate root and ier the error message
(I like to say $ier = 0$ is success and $ier \neq 0$ is failure. Different ier values can mean different sources of failure.)

---

(1) **for** $i = 1, \ldots, N_{\max}$
(2) Set $p = p_0 - f(p_0)/f'(p_0)$ (calculate p_i)
(3) If $|p - p_0| < \epsilon$ (Check if it is time to stop)
Set $p^* = p$, $ier = 0$ and exit
end if
(4) Set $p_0 = p$. (update)
end
(5) Set $ier = 1$ and display "Newton ran out of iterations."

There are other options for stopping criteria. What are they?

- $|f(p)| < 10^2\epsilon_M$ (Is the value close to 0? This could be arbitrarily off)
- $|p - p_0| < \epsilon$ (Absolute distance between iterates)
- $\frac{|p-p_0|}{|p|} < \epsilon$ if $p \neq 0$ (relative distance between iterates. This gives a better approximate of how many digits are correct in the approximation.)

A key decision we made in creating this algorithm was dropping the $\frac{f''(\eta)}{2}(p - p_0)^2$ term. The only way that this is reasonable is if $|p - p_0|$ is really small. This choice has implications on the convergence of the method.

Theorem 0.1. *(2.6) Let $f \in C^2[a, b]$. If $p \in [a, b]$ such that $f(p) = 0$ and $f'(p) \neq 0$ then there exists a $\delta > 0$ such that Newton's method generates a sequence $\{p_n\}_{n=1}^{\infty}$ converging to p for any initial guess $p_0 \in (p - \delta, p + \delta)$.*

Proof. See textbook. □

Question: It is reasonable to ask: How do we know if the guess is close enough?
Answer: Use our fixed point theory. Set $g(x) = x - \frac{f(x)}{f'(x)}$ and see if $|g'(x)| < 1$ for the initial guess.

Secant method
Question: What happens if we do not have access to the derivative?
Answer: We can approximate it.
Recall from calculus that

$$f'(p_{n-1}) = \lim_{x \to p_{n-1}} \frac{f(x) - f(p_{n-1})}{x - p_{n-1}}.$$

Unfortunately, the computer cannot compute this limit so we need to do something different.

If the sequence $\{p_n\}_{n=1}^{\infty}$ converges to p, then p_{n-2} is "close" to p_{n-1}. Thus we can approximate the derivative by

$$f'(p_{n-1}) \approx \frac{f(p_{n-2}) - f(p_{n-1})}{p_{n-2} - p_{n-1}}.$$

Plugging this approximation into Newton's method, we get

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-2} - p_{n-1})}{f(p_{n-2}) - f(p_{n-1})}$$

This is called *Secant method*. Why? We are following the roots of secant lines.
Draw a picture of an iteration

PSEUDOCODE: Secant method

Input: p_0, p_1 = initial guesses, ϵ =tolerance, $N_{\max}$= max number of iterations
$f(x)$
Output: p^*= the approximate root and *ier* the error message
(I like to say $ier = 0$ is success and $ier \neq 0$ is failure. Different *ier* values can mean different sources of failure.)

---

(1) Set $q_0 = f(p_0)$ and $q_1 = f(p_1)$
(2) **for** $i = 1, \ldots, N_{\max}$
(3) Set $p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0)$ (calculate p_i)
(4) If $|p - p_1| < \epsilon$ (Check if it is time to stop)
Set $p^* = p$, $ier = 0$ and exit
end if
(5) Set $p_0 = p_1$, $q_0 = q_1$. (update data)
$p_1 = p$, $q_1 = f(p)$
end
(6) Set $ier = 1$ and display "Secant ran out of iterations."