1.4 Newton's Method, Secant Method, and their variations

The idea behind Newton's method is simple: if $f(x)$ is smooth (continuously differentiable), with $f'(r) \neq 0$, we can approximate it with a line near x_0 (near initial guess), and the point where this line intersect zero is a better guess of our root (zero) value.

Newton Method Derivation:

Let's start with the point-slope equation of line through the initial point/guess (x_0, y_0), as in the figure below

$$y - y_0 = m(x - x_0)$$

The next approximation x_1 corresponds to the intersection of this line with the horizontal axis $y = 0$, therefore:

$$0 - y_0 = m(x_1 - x_0)$$

Divide by the slope of the tangent line, which equals $\boldsymbol{m = f'(x_0)}$, and solve for x_1:

$$x_1 = x_0 - \frac{y_0}{m} \equiv x_0 - \frac{f(x_0)}{f'(x_0)} \qquad (1)$$

Repeating this many times, we obtain the **Newton's method** iteration:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \equiv g(x_n)$$

It converges *much* faster than prior methods: apply it to $f(x) = x^2 - 2$ with $x_0 = 1$, and you will obtain an approximation for $\sqrt{2}$ correct to 5 digits in 3 iterations! ☺ (this is conjectured to be the ancient Babylonian method)

Secant Method Derivation:

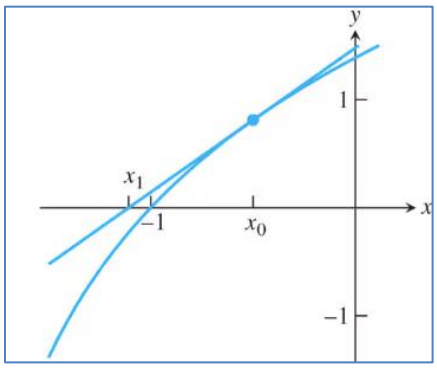
Secant method idea is the same, except that we use a **secant line** instead of the tangent line to approximate our function. Two points x_0 and x_1 are needed for a secant line, with slope equal to $\boldsymbol{m = \frac{f(x_1) - f(x_0)}{x_1 - x_0}}$. Plug this into Eq. (1):

$$x_2 = x_1 - \frac{y_1}{m} \equiv x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)}$$

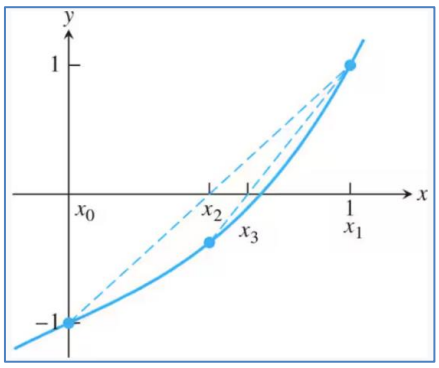
Repeating this step, we obtain the Secant method, which is a 2-point iteration (2nd order difference equation):

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \equiv g(x_n, x_{n-1})$$

Newton's method = tangent line approximation:



Secant method = Secant line approximation:



As we will show, **Newton's method** converges faster, but requires knowledge of derivative. Even if the derivative is known, the **secant method is more efficient** – requiring only one new evaluation $f(x_n)$ per step -- but converges slower.

Steffensen's Method: variation of the Newton's Method (not in textbook):

If we don't explicitly know the derivative (a very typical situation), we can approximate it by the slope of the secant line using two points that are closer to each other than x_n is to x_{n-1}. The definition of derivative implies that

$$f'(x) \approx \frac{f(x+h)-f(x)}{h}$$

where *h* approaches 0. Now, as we approach the root of $f(r)=0$, the quantity that is truly very small is $h = f(x)$ itself (this value is also known as the *residual*, related to the so-called *backward error* of the method). Therefore, the slope of the secant line closest to the slope of the tangent line is obtained by plugging $h = f(x)$ into the above equation:

$$m \equiv f'(x) \approx \frac{f(x+f(x))-f(x)}{f(x)}$$

Plugging this as slope *m* into Eq. (1) on the previous page, we obtain:

$$x_1 = x_0 - \frac{y_0}{m} \equiv x_0 - \frac{f(x_0)^2}{f(x_0+f(x_0))-f(x_0)}$$

Repeating this many times, we obtain the Steffensen's one-point iteration

$$x_{n+1} = x_n - \frac{f(x_n)^2}{f(x_n+f(x_n))-f(x_n)} \equiv g(x_n)$$

The visual illustration of this method is the same as for the Newton's method (see previous page). The advantage is that no knowledge of derivative is required, while the efficiency is the same (two function evaluations per step).

Newton's Method: Derivation #2 (optional but recommended)

Let's denote $g(x) = x + h(x)f(x)$, where $h(x)$ is any smooth non-zero function. This obviously converts our root equation $f(r)=0$ to the fixed-point equation $g(r)=r$.

Now, let's ask a question: for which $h(x)$ does the iteration $x_{n+1} = g(x_n)$ converge most quickly? As we learned previously, iteration error contracts at each step by a factor $C = |g'(r)|$, so we should aim for $C = g'(r) = 0$:

$$\begin{aligned} g'(r) &= 1 + h'(r)f(r) + h(r)f'(r) \\ &= 1 + h(r)f'(r) = 0, \end{aligned}$$

where we used $f(r)=0$. Solving this yields $h(r) = -\frac{1}{f'(r)}$, therefore we should pick $h(x) = -\frac{1}{f'(x)}$. Plugging this back into our original equation $g(x) = x + h(x)f(x)$ immediately leads to the Newton's iteration: nice!

Quadratic Convergence of Newton's Method:

Derivation #2 above shows that Newton's method corresponds to the optimal choice $g'(r) = 0$, which tells us that the convergence should be at least quadratic. To see this, recall the derivation of the iteration convergence condition: Plugging $\boldsymbol{x_n = r + \epsilon_n}$ into our iteration $\boldsymbol{x_{n+1} = g(x_n)}$, we obtain:

$$\boldsymbol{x_{n+1} = r + \epsilon_{n+1} = g(r + \epsilon_n).}$$

If $g(x)$ is smooth near the root, we can expand it around *r* to 2nd order (since the linear term will be zero):

$$\boldsymbol{r + \epsilon_{n+1} = g(r + \epsilon_n) = g(r) + g'(r)\epsilon_n + \frac{g''(\alpha)}{2}\epsilon_n^2}$$

where $\alpha \in (r, x_n)$. Since $\boldsymbol{r = g(r)}$, and we have shown above that $\boldsymbol{g'(r) = 0}$, we obtain $\boldsymbol{\epsilon_{n+1} = \frac{g''(\alpha)}{2}\epsilon_n^2}$

Therefore, this method is faster than all we have examined so far: it is *at least* **quadratically convergent**. Let us examine the coefficient in the above error relationship, relating $g''(x)$ to $f(x)$.

Newton's method error: approach #1

Use 2nd order Taylor expansion for $f(r)$ around x_n:

$$\boldsymbol{0 = f(r) \equiv f(x_n) + f'(x_n)(r - x_n) + \frac{f''(\alpha)}{2}(r - x_n)^2}$$

where $\alpha \in (r, x_n)$. Let's now divide by the slope, $f'(x_n)$:

$$0 = \frac{f(x_n)}{f'(x_n)} + r - x_n + \frac{f''(\alpha)}{2f'(x_n)}(r - x_n)^2$$

Note that the 1st and the 3rd terms combine to form the next iteration of the Newton's method, with a negative sign:

$$0 = r - x_{n+1} + \frac{f''(\alpha)}{2f'(x_n)}(r - x_n)^2$$

Denoting (as we usually do) $\boldsymbol{x_n = r + \epsilon_n}$, and solving for ϵ_{n+1}, we obtain:

$$\boldsymbol{\epsilon_{n+1} = \frac{f''(\alpha)}{2f'(x_n)}\epsilon_n^2}$$

Therefore, close to the root we have:

$$\boldsymbol{\epsilon_{n+1} \approx M\epsilon_n^2} \quad \text{where} \quad \boldsymbol{M \equiv \frac{f''(r)}{2f'(r)}}$$

Note: a more rigorous analysis can only provide an upper bound on the error, as follows:

$$\boldsymbol{|\epsilon_{n+1}| \le M\epsilon_n^2} \quad \text{where} \quad \boldsymbol{M \equiv max_{[a,b]} \left|\frac{f''(x)}{2f'(x)}\right|}$$

In above, the maximum is taken over some interval [*a*, *b*] containing all of the iterates x_n and our root *r* (this interval is called the *trapping region* of the iteration). It can be shown that a sufficient condition for convergence is $\boldsymbol{M|\epsilon_0| < 1}$ (provided the iterates remain in the trapping region)

Newton's method error: approach #2 (direct but messy)

Result on the previous page implies that $\boldsymbol{\epsilon_{n+1} \approx \frac{g''(r)}{2}\epsilon_n^2}$ for any quadratically convergent method, where for Newton's iteration we have $\boldsymbol{g(x) \equiv x - \frac{f(x)}{f'(x)}}$.

Let's then obtain $\boldsymbol{M \equiv \frac{g''(r)}{2}}$ directly:

$$\begin{cases} g'(x) = 1 - \dfrac{[f']^2 - f \cdot f''}{[f']^2} = \dfrac{f \cdot f''}{[f']^2} \\ g''(x) = \dfrac{(f'f'' + ff''')[f']^2 - 2ff'[f'']^2}{[f']^4} \end{cases}$$

Since $f(r) = 0$, the 2nd derivative at the root simplifies to $g''(r) = \dfrac{(f'f'')[f']^2}{[f']^4} = \dfrac{f''(r)}{f'(r)}$

Thus, we proved the equality between two ways of writing the constant in the error formula (neat!): $\boldsymbol{M \equiv \frac{g''(r)}{2} = \frac{f''(r)}{2f'(r)}}$

Note that for a multiple root (zero of order higher than 1), for example $f(x) = x^m$ (with $m > 1$), we have $f'(r) = 0$. Looking at the expressions above, *M* will be infinite! Indeed, Newton's method fails to converge quadratically in this case. Fortunately, we can rescue the method with some modification (see next page)

Secant method: convergence order

Secant method is similar to the Newton's method, and its error also has a similar behavior; because it's a two-point method, the error at the next step depends on the error of both guesses that are used at each iteration:

$$\boldsymbol{\epsilon_{n+1} \approx M\epsilon_n\epsilon_{n-1}} \text{ where } \boldsymbol{M \equiv \frac{f''(r)}{2f'(r)}}$$

Note that the convergence order is not obvious from this equation: let's examine it further. Multiplying both sides of the above equation by *M* and denoting $\delta_n \equiv M\epsilon_n$, we obtain a simplified relationship:

$$\boldsymbol{\delta_{n+1} \approx \delta_n\delta_{n-1}} \quad (2)$$

If the method converges with order *p* (yet unknown), then $\boldsymbol{\delta_{n+1} \approx \delta_n^p}$ and $\boldsymbol{\delta_n \approx \delta_{n-1}^p}$. Let's plug this into the above equation:

$$\begin{cases} \delta_{n+1} \approx \delta_n^p \approx \left(\delta_{n-1}^p\right)^p \equiv \delta_{n-1}^{p^2} \\ \delta_n\delta_{n-1} \approx \delta_{n-1}^p\delta_{n-1} \equiv \delta_{n-1}^{p+1} \end{cases}$$

Since these two are approximately equal given Eq. 2 above, we obtain a quadratic equation for convergence order *p*:

$$\boldsymbol{p^2 = p + 1}$$

Therefore, we have to solve $\boldsymbol{p^2 - p - 1 = 0}$, therefore $\boldsymbol{p_{1,2} = \frac{1 \pm \sqrt{5}}{2}}$. The larger of these two numbers (the solution with the plus sign) is called the golden ratio, and represents convergence order of the Secant method: neat! Thus, convergence order of the Secant method is $\boldsymbol{p = \frac{1+\sqrt{5}}{2} \approx 1.618.}$ Thus, convergence is super-linear (since $p > 1$), but not as fast as the Newton's method (since $p < 2$).

Higher-Order Methods: Halley's Method

Halley's method can be obtained from quadratic Taylor approximation to the function curve transformed as $x = F(y)$ obtained by inverting the function $y = f(x)$. This inversion is desirable to ensure that the resulting parabola intersects the $y = 0$ axis. Combined with implicit differentiation to express derivatives of F(y) in terms of derivatives of $f(x)$, and finding the intersection of the parabola with $y = 0$, one obtains the following, 3rd order accurate method:

$$\boldsymbol{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f''(x_n)f(x_n)^2}{2[f'(x_n)]^3}}$$

With the error approximately given by

$$\epsilon_{n+1} \approx \frac{3[f''(r)]^2 - f'(r)\,f'''(r)}{6[f'(r)]^2}\epsilon_n^3$$

As this formula shows, this method is third order.

The main disadvantage is that accurate first and second derivatives of the original function are needed.

Example: apply it to $f(x) = x^2 - 2$ to see how quickly this iteration converges to $\sqrt{2}$: in **just 3 iterations** starting from $x_0 = 1$, you will reach **machine precision**!

Newton's method: modifications for multiple roots

Definition: Continuous function $f(x)$ has a zero of order *m* (root of multiplicity *m*) at $x = r$ if it can be written as $f(x) = (x - r)^m h(x)$, where $h(x)$ is also continuous, and $h(r) \neq 0$.

Simple examples:

(1) $f(x) = x^3 \cos(x)$ has a zero of order 3 at $r = 0$, and zeros of order 1 at odd multiples of $\frac{\pi}{2}$

(2) $f(x) = sin^4 x$ has a zero of order 4 at any integer multiple of π.

If you apply the Newton's method to an even simpler example, $f(x) = x^2$, you will see that the error is reduced by a factor of ½ at each iteration, so its convergence is linear, not quadratic. However, if you plug this function into the following modification of Newton's method, you will converge to true root $r = 0$ in a single iteration! ☺

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}$$

In the above formula, *m* is the order of the zero (for our example $f(x) = x^2$, set $m = 2$).

What if we do not know *a priori* the order of the zero we are looking for? Turns out that there is another, more complicated modification, which automatically determines the order and finds the root at the same time:

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{f'(x_n)^2 - f(x_n)f''(x_n)}$$

Quadratic Convergence of the Modified Newton's Method

Consider again the modified Newton's method iteration: $x_{n+1} = g(x_n) \equiv x_n - \boldsymbol{m}\frac{f(x_n)}{f'(x_n)}$

Assume that our function has zero *r* of order *m* (see previous page): $f(x) = (x - r)^m h(x)$

Plug this into the modified Newton's iteration function above:

$$g(x) \equiv x - \boldsymbol{m}\frac{f(x)}{f'(x)} = x - \frac{\boldsymbol{m}(x-r)^m h(x)}{\boldsymbol{m}(x-r)^{m-1}h(x) + (x-r)^m h'(x)}$$

$$= x - \frac{\boldsymbol{m}(x-r)h(x)}{\boldsymbol{m}\,h(x) + (x-r)h'(x)}$$

$$= x - \frac{x-r}{1 + \frac{(x-r)h'(x)}{\boldsymbol{m}\,h(x)}}$$

Evaluating the above expression at x_n yields: $x_{n+1} = g(x_n) = x_n - \frac{x_n - r}{1 + \frac{(x_n - r)h'(x)}{mh(x)}}$

$$\equiv x_n - \frac{\epsilon_n}{1 + \epsilon_n \frac{h'(x_n)}{mh(x_n)}}$$

$$= x_n - \epsilon_n\left[1 - \epsilon_n \frac{h'(x_n)}{mh(x_n)} + O(\epsilon_n^2)\right]$$

Subtracting *r* on both sides of this equation, we obtain

$$\boldsymbol{\epsilon_{n+1}} = \epsilon_n - \epsilon_n\left[1 - \epsilon_n \frac{h'(x_n)}{mh(x_n)} + O(\epsilon_n^2)\right]$$

$$= \frac{h'(x_n)}{mh(x_n)}\boldsymbol{\epsilon_n^2} + O(\epsilon_n^3)$$

The proof of quadratic convergence is complete!

According to above, close to the multiple root the error can be approximated as $\boldsymbol{\epsilon_{n+1} \approx M\epsilon_n^2}$ where $M \equiv \frac{\boldsymbol{h'(r)}}{\boldsymbol{mh(r)}}$

Similar method can be used for the more general modification of Newton's method that involves 2nd derivative, but with more algebra, obviously.

Newton's method for *N* functions of *N* variables

In practice, Newton's method is often applied to solve a more complex problem of solving *N* smooth but non-linear equations for *N* unknowns (this problem arises for instance when solving a non-linear PDE using an implicit solver). The extension to this case is remarkably straightforward:

Consider a smooth N-dimensional vector field **F**(**X**) on an N-dimensional domain: $\mathbb{R}^N \xrightarrow{F(X)} \mathbb{R}^N$

We can expand this vector field to first order in a Taylor series around current guess $\mathbf{X}_n$:

$$F(X) \approx F(X_n) + DF(X_n)(X - X_n)$$

where $DF(X_n)$ is the Jacobian matrix of $F(X)$ evaluated at X_n. At the next guess X_{n+1} we want this to be zero; solving $F(X_{n+1}) = 0$ yields a direct analog of the scalar Newton's rule:

$$X_{n+1} = X_n - \left[DF(X_n)\right]^{-1} F(X_n)$$

The 2nd term is a product of an N x N matrix (the inverse of the Jacobian) and an N-dimensional vector field F.

Example:

Find the intersection of a closed curve $x^4 + y^4 = 1$ with the cubic $y = x^3$, numerically.

Denote $X = \begin{pmatrix} x \\ y \end{pmatrix}$. The vector field we want to be zero is $F(X) = \begin{pmatrix} x^4 + y^4 - 1 \\ x^3 - y \end{pmatrix}$ with Jacobian $DF(X) = \begin{pmatrix} 4x^3 & 4y^3 \\ 3x^2 & -1 \end{pmatrix}$

Inverting a 2x2 matrix is easy: $\left[DF(X)\right]^{-1} = \frac{1}{\det[DF]} \begin{pmatrix} 1 & 4y^3 \\ 3x^2 & -4x^3 \end{pmatrix} = \frac{1}{4x^2\left(x + 3y^3\right)} \begin{pmatrix} 1 & 4y^3 \\ 3x^2 & -4x^3 \end{pmatrix}$

Now, we can compute $\left[DF(X_n)\right]^{-1} F(X_n) = \frac{1}{4x^2\left(x + 3y^3\right)} \begin{pmatrix} 1 & 4y^3 \\ 3x^2 & -4x^3 \end{pmatrix} \begin{pmatrix} x^4 + y^4 - 1 \\ x^3 - y \end{pmatrix}$

$$= \frac{1}{4x^2\left(x + 3y^3\right)} \begin{pmatrix} x^4 + 4y^3x^3 - 3y^4 - 1 \\ 3x^2\left(y^4 - 1\right) - x^6 + 4x^3y \end{pmatrix}$$

Slightly simplifying and plugging this into our iteration, we obtain:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \frac{1}{4\left(x_n + 3y_n^3\right)} \begin{pmatrix} x_n^2 + 4y_n^3x_n - \dfrac{1 + 3y_n^4}{x_n^2} \\ 3\left(y_n^4 - 1\right) - x_n^4 + 4x_ny_n \end{pmatrix}$$

Figure below shows four steps of this iteration (filled red circles), which quickly move to the intersection/solution:

