1 Newton's Method (Section 3.2)

Newton's Method is an efficient iterative method for solving nonlinear equations, assuming it works. Let r be a root of some function, and let x be an approximation to r. Then, our goal is to find an estimate of r, or $r = x_{m+1} = x_m + h$, where $x_{m+1}, x_m, h \in \mathbb{R}$. If f'' exists and is continuous, then by the Taylor series, we have

$$0 = f(r) = f(x_{m+1}) = f(x_m) + hf'(x_m) + \mathcal{O}(h^2).$$

Then, $h = \frac{f(x_m)}{f'(x_m)}$ will be part of an updating scheme. For a sufficiently small h (i.e., x is near r), then we can reasonably ignore the $\mathcal{O}(h^2)$ term.

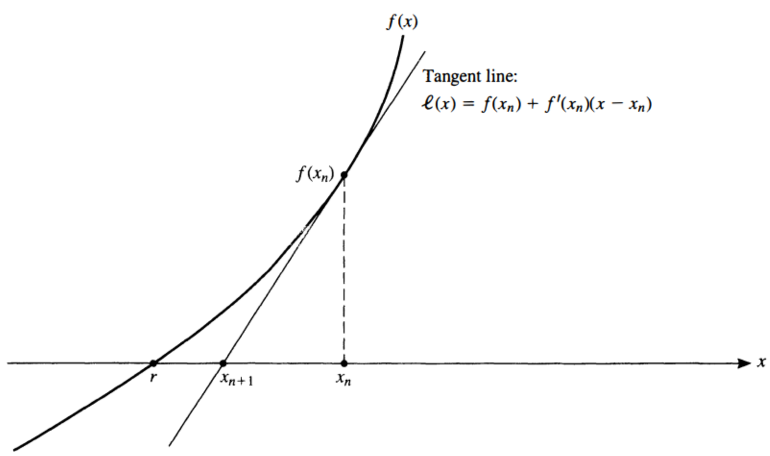
1.1 Newton Iteration in 1-Dimension

For $m = 0, 1, 2, \ldots$, we have

$$x_{m+1} = x_m - \frac{f(x_m)}{f'(x_m)}.$$

In other words, Newton's method begins with an estimate x_0 of r, and then defines inductively. If we let $x_{m+1} = x$, then the linearization at x_m is

$$f(x_{m+1}) = f(x) \approx f(x_m) + (x - x_m)f'(x_m) = \ell(x) = 0.$$



1.1.1 The Algorithm

Let

- M: the maximum number of iterations.
- δ: the step tolerance such that $|x_{m+1} - x_m| < \delta$.
- ϵ: the convergence tolerance $|f(x_{m+1})| < \epsilon$.

With a suitable x_0 being the starting point, the algorithm is as follows.

Algorithm 1 Newton's Algorithm

```
1:  function NEWTON(x_0, M, δ, ϵ)
2:      v ← f(x_0)
3:      if |v| < ϵ then
4:          return
5:      end if
6:      for k ← 1 to M do
7:          x_1 ← x_0 − v / f'(x_0)
8:          v ← f(x_1)
9:          if |x_1 − x_0| < δ or |f(x_1)| < ϵ then
10:             break
11:         end if
12:         x_0 ← x_1
13:     end for
14:     return v
15: end function
```

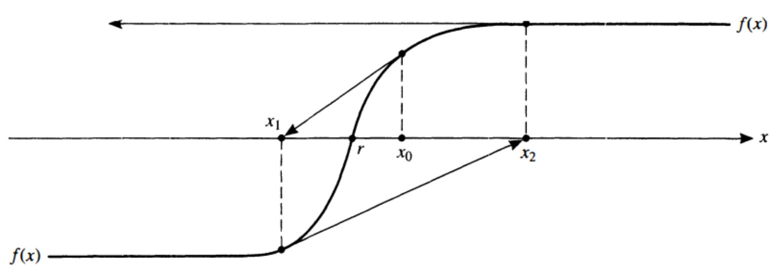
One thing to note: the algorithm will fail if $f'(x_0) = 0$ due to the division in line 7.

1.1.2 Requirements

Defining the correct starting point x_0 is important. A bad starting point can result in nonconvergence.

The function itself also matters. Other problems include when $f'(x_0) = 0$ or $f'(x_0)$ is infinite.

(Example.) Consider the following function,



For this function, if $|x_0 - r| < |x_1 - r|$, then $|x_m - r| < |x_{m+1} - r|$ and nonconvergence happens. In particular, the shape of the curve is such that for certain starting values, the sequence $[x_n]$ diverges.

1.1.3 Error Analysis

Let the error be defined by $e_m = x_m - r$. Assume that $f(r) = 0 \neq f'(r)$ and f'' is continuous. Then,

$$\begin{aligned} e_{m+1} &= x_{m+1} - r \\ &= \left(x_m - \frac{f(x_m)}{f'(x_m)} \right) - r \\ &= e_m - \frac{f(x_m)}{f'(x_m)} \\ &= \frac{e_m f'(x_m) - f(x_m)}{f'(x_m)}. \end{aligned}$$

We can now incorporate a Taylor expansion,

$$0 = f(r) = f(x_m - e_m) = f(x_m) - e_m f'(x_m) + \frac{1}{2} e_m^2 f''(\xi)$$

for some aribitrary ξ between x_m and r that makes the equation equal. Then,

$$\begin{aligned} -(f(x_m) - e_m f'(x_m)) &= \frac{1}{2} e_m^2 f''(\xi) \\ \implies e_{m+1} &= \frac{\frac{1}{2} e_m^2 f''(\xi_m)}{f'(x_m)} \\ \implies e_{m+1} &\approx C e_m^2, \end{aligned}$$

where C is a bound of $\frac{\frac{1}{2} f''(\xi_m)}{f'(x_m)}$. So, in Newton's method, we have a quadratic convergence so that $e_{m+1} \leq C e_m^2$ or $|x_{m+1} - r| \leq C|x_m - r|^2$.

Remark: If f is $C^2(\mathbb{R})$ is increasing, is convex (i.e., $f''(x) > 0$ for all x), and has a zero, then Newton's Method converges to it from any starting point.

(Example.) Let $R > 0$ and $x = \sqrt{R}$. Then,

$$f(x) = x^2 - R = 0$$

and

$$f'(x) = 2x.$$

Then, the iteration corresponds to

$$x_{m+1} = x_m - \frac{f(x_m)}{f'(x_m)} = x_m - \frac{x_m^2 - R}{2x_m} = \frac{1}{2}\left(x_m + \frac{R}{x_m}\right).$$

1.2 Newton Iteration in 2-Dimensions

Suppose $f_1(x_1^*, x_2^*) = 0$ and $f_2(x_1^*, x_2^*) = 0$. In other words, we have two variables and two functions. The goal is to linearize $x_1^* = x_i + h_i$, where h_i is some small pertubation (error) and $1 \leq i \leq 2$. Then,

$$0 = f_1(x_1^*, x_2^*) = f_1(x_1 + h_1, x_2 + h_2) \approx f_1(x_1, x_2) + h_1 \frac{\partial f_1}{\partial x_1} + h_2 \frac{\partial f_1}{\partial x_2}$$

$$0 = f_2(x_1^*, x_2^*) = f_2(x_1 + h_1, x_2 + h_2) \approx f_2(x_1, x_2) + h_1 \frac{\partial f_2}{\partial x_1} + h_2 \frac{\partial f_2}{\partial x_2}.$$

From there, we can create a system

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}}_{J} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}.$$

Here, the J is the Jacobian matrix. The goal is to solve for $\begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$. By subtracting a term from the above equation, we have

$$\begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} = -J \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \implies (-J)^{-1} \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}.$$

The iteration, then, is

$$\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \end{bmatrix} + \begin{bmatrix} h_1^{(k)} \\ h_2^{(k)} \end{bmatrix} \qquad k \geq 0.$$

Remarks:

- Just like in the one-dimensional case where $f'(x_0) \neq 0$, for this iterative formula to be well-defined, the Jacobian matrix must be nonsingular.
- The vector, $\begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$, is computed by solving a system (e.g., using Gaussian elimination).

1.3 Generalization to Many Dimensions

Given m variables and m equations, then

$$f_i(x_1, x_2, \ldots, x_m) = 0 \qquad 1 \leq i \leq m.$$

Then,

$$F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \quad H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_m \end{bmatrix}.$$

Then, we can linearize the vector function,

$$0 = F(X + H) \approx F(X) + F'(X)H,$$

where

$$F'(x) = J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_m} \end{bmatrix} \in \mathbb{R}^{m \times m}.$$

So, Newton's method for m nonlinear equations in m variables is given by

$$X^{(k+1)} = X^{(k)} + H^{(k)},$$

where the Jacobian system is

$$\underbrace{F'(X^{(k)})}_{J(X^{(k)})} H^{(k)} = -F(X^{(k)}).$$

1.3.1 The Algorithm

So, in the iterative algorithm, we have

$$J(X^{(k)})H^{(k)} = -F(X^{(k)}).$$

Recall that $X^{(k+1)} = X^{(k)} + H^{(k)}$. With this in mind, our algorithm takes in the following:

- M: the maximum number of iterations.
- ϵ: the tolerance, $||F(x^{(k)})||_2 \leq \epsilon$
- δ: the step, $||x^{(k+1)} - x^{(k)}||_2 \leq \delta$.

Algorithm 2 Newton's Algorithm in Multiple Dimensions

function NEWTON($X^{(0)}, M, \epsilon, \delta$)
    $X^{(k)} \leftarrow X^{(0)}$
    $F^{(k)} = F(X^{(k)})$
    for $k \leftarrow 1$ to M **do**
        $J^{(k)} \leftarrow F'(X^{(k)})$
        $H^{(k)} \leftarrow -J^{(k)} \backslash F^{(k)}$ ▷ Solve the system.
        $X^{(k)} \leftarrow X^{(k)} + H^{(k)}$
        $F^{(k)} \leftarrow F(X^{(k)})$
        if $||F^{(k)}||_2 \leq \epsilon$ or $||H^{(k)}||_2 \leq \delta$ **then**
            break
        end if
    end for
    return $X^{(k)}$
end function

(Example.) Suppose we have

$$F(x) = \begin{bmatrix} x_1x_2 - x_3^2 - 1 \\ x_1x_2x_3 - x_1^2 + x_2^2 - 2 \\ e^{x_1} - e^{x_2} + x_3 - 3 \end{bmatrix} = 0$$

and $X^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. Then,

$$F'(x) = J(x) = \begin{bmatrix} x_2 & x_1 & -2x_3 \\ x_2x_3 - 2x_1 & x_1x_3 + 2x_2 & x_1x_2 \\ e^{x_1} & -e^{x_2} & 1 \end{bmatrix}.$$