With these observations in mind, we will explore a second algorithm for finding the root of an equation. This algorithm is known as *Newton's Method* (or the *Newton-Raphson Method*). Like the Bisection Method, this solution technique uses an estimate of the root of an equation to generate an improved estimate of the root, which will become closer and closer to the actual solution as the algorithm is repeated. Unlike the Bisection Method, however, Newton's Method uses a *single* estimate of the root, and uses the projection of the slope of the function at the estimate to obtain the improved root estimate.

Newton's Method can be described as follows. Suppose we are trying to determine the root of the function $y(x)$. We begin by guessing the location of the value x that will make $y(x) = 0$. We will call this guess x_i. To generate an improved estimate of the root, we perform the following steps:

- We evaluate the true value of the function at this point, $y(x_i)$.
- We compute the slope of the function $y(x)$ at the point x_i. We call this slope value $y'(x_i)$. Mathematically, this is done by using principles of calculus to find the *first derivative* of the function $y(x)$, and substituting the value of x_i into the new equation. The result is the equation of a line tangent to the curve $y(x)$; this is illustrated in Figure 6.8.
- The point where this slope line intersects the x-axis is used as the improved estimate of the root, which we will call x_n. This point is located using Equation 6.19:

$$x_n = x_i - \frac{y(x_i)}{y'(x_i)} \tag{6.19}$$

This is illustrated in Figure 6.9.

These steps are repeated over and over again, until an estimate that is close enough to the true root is determined.

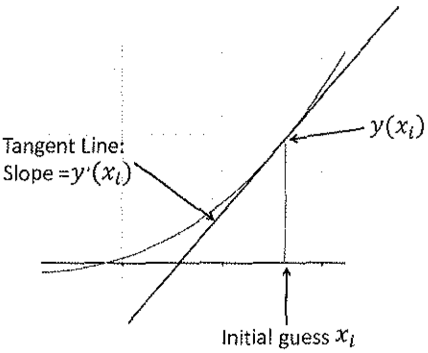

Figure 6.8

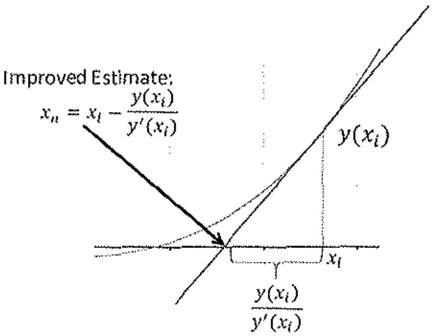

Figure 6.9

As an example, consider the function:

$$y = 3x^3 - 15x^2 - 20x + 50 \tag{6.20}$$

It is known from principles of calculus that the slope of the equation is as follows:

$$y' = 9x^2 - 30x - 20 \tag{6.21}$$

We will use Newton's Method to find the root of Equation 6.20.

To begin, we will arbitrarily guess that the root occurs at the point $x = 10$. This is strictly an arbitrary guess; we could certainly come up with a more accurate guess by plotting Equation 6.20 over a range of x values, but for this demonstration an arbitrary guess will suffice. Applying Newton's Method, we:

- Evaluate Equation 6.20 at the point $x_i = 10$. In this case:

$$\begin{aligned} y(10) &= 3(10)^3 - 15(10)^2 - 20(10) + 50 \\ &= 1350 \end{aligned} \tag{6.22}$$

- Evaluate the slope of the curve at the point $x_i = 10$. This is done by evaluating Equation 6.21 with $x_i = 10$. In this example, the slope is as follows:

$$\begin{aligned} y' &= 9(10)^2 - 30(10) - 20 \\ &= 580 \end{aligned} \tag{6.23}$$

- Using Equation 6.19, the improved root estimate can be obtained. In the example, this yields:

$$\begin{aligned} x_n &= 10 - \frac{1350}{580} \\ &= 7.6724 \end{aligned} \tag{6.24}$$

This is illustrated graphically in Figure 6.10.

Checking the actual value of Equation 6.20 at this improved estimate yields a value of $y = 368.494$. If this is close enough to zero, we could stop here. If not, we would set $x_i = 7.6724$, and repeat the algorithm again. Table 6.2 shows seven iterations of Newton's Method; note that after five iterations, the root has settled on the same value to four decimal places.

Table 6.2 Seven Iterations of Newton's Method for Equation 6.20

x_i	y	y'	x_n
10.0000	1350.00	580.0000	7.6724
7.6724	368.4941	279.6210	6.3546
6.3546	87.0049	152.7887	5.7851
5.7851	13.1273	107.6559	5.6632
5.6632	0.5457	98.7502	5.6577
5.6577	0.0011	98.3529	5.6577
5.6577	0.0000	98.3521	5.6577

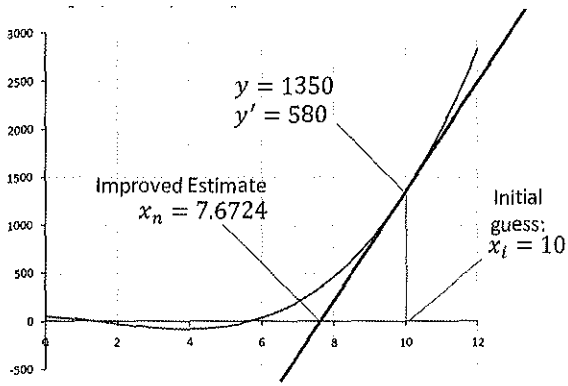

Figure 6.10

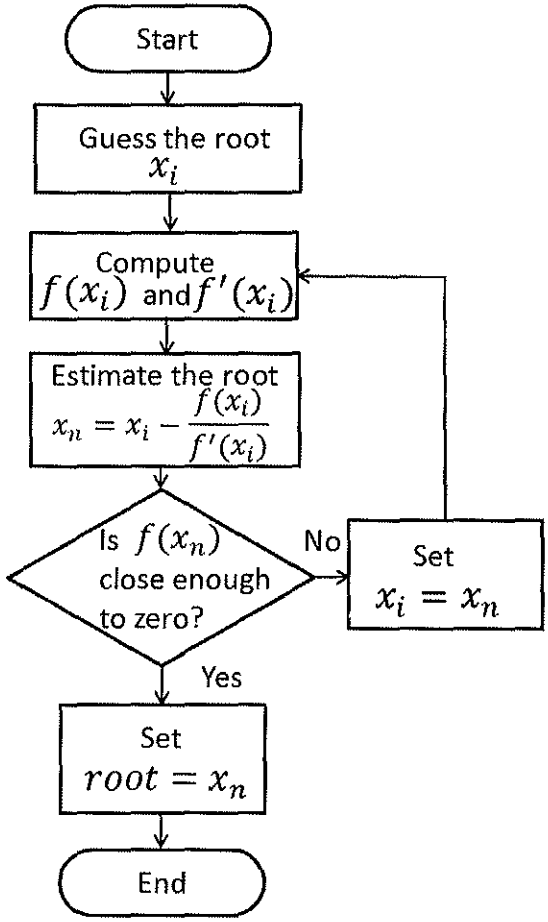

Figure 6.11

A flowchart for this method is shown in Figure 6.11.

Comparison of the Bisection Method and Newton's Method yields a major difference:

- The Bisection Method requires two initial guesses, with one on each side of the root. Such a method is known as a *bracketing method.*
- Newton's Method requires a single initial guess, with no restrictions placed on it (other than that the slope must not be zero at the initial guess). Such a method is called an *open method.*

Bracketing methods and open methods have other fundamental differences that should be addressed:

- As long as there is one root between the two initial guesses in a bracketing method, the algorithm is guaranteed to find the root, to whatever final precision is specified. As such, we say that bracketing methods have "guaranteed convergence." Open methods are *not* guaranteed to find a root; they sometimes "diverge," or fail to ever reach a stopping point at an appropriate root. Programmers who implement open methods must take care to check for divergence, and stop their programs when it is detected.
- While convergence is not guaranteed in an open method, in most cases convergence will occur much quicker in an open method than in a bracketing method; the desired precision of the solution will be reached with fewer iterations in an open method.