

## (4.8) Newton's Method

Often, when we solve an equation, we would like to find an “exact” solution. For example, one solution of  $x^2 + x - 1 = 0$  is  $x = -\frac{1}{2} + \frac{\sqrt{5}}{2}$ . But sometimes, all we need is an approximate solution:  $x = 0.618034$ . This might be because we're trying to solve an applied problem for which only a decimal value is needed. Or it might be because an exact solution is impractical or unavailable.

**Example.** One solution of a cubic equation of the form  $x^3 + px + q = 0$  is given by Cardano's formula:

$$x = \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}}$$

For example, the equation  $x^3 + 3x - 7 = 0$  happens to have exactly one real root, and the formula gives

$$x = \sqrt[3]{\frac{7 + \sqrt{53}}{2}} + \sqrt[3]{\frac{7 - \sqrt{53}}{2}}$$

This isn't too bad; we can use a calculator to verify that  $x = 1.406288$  if we need the decimal value. But Cardano's formula has an interesting quirk: if the equation happens to have three real roots, the formula gives an expression involving cube roots of complex numbers. For example, for the equation  $x^3 - 9x + 10 = 0$ , Cardano's formula gives

$$x = \sqrt[3]{-5 + \sqrt{2}i} + \sqrt[3]{-5 - \sqrt{2}i}$$

It is by no means obvious that  $x$  is actually equal to 2. (The other two roots of this equation are  $1 \pm \sqrt{6}$ .) The remedy at this point is to use DeMoivre's formula from trigonometry to find cube roots of complex numbers. But no one can blame you if you decide this is too complicated to pursue.

The situation for fourth-degree polynomial equations (“quartic” equations) is even worse: the formulas for their exact solutions are very complicated, and can also lead to situations where roots of complex numbers occur even when the solution is not complex. As far as fifth degree equations are concerned, it has been proven that there is no formulas, involving fifth roots and so on, for the solutions. In that circumstance, no exact solution is available. This is also true for most equations involving “transcendental” functions such as  $\sin x$  or  $\cos x$ . For example: we cannot write an exact expression (involving familiar functions in only finitely many terms) for the solution of  $\cos x = x$ .

So by now it is clear that we cannot find exact solutions for many equations. We can only find decimal approximations for the solutions of many equations. But often, that's all we need anyway. So we need an effective technique for finding approximations of solutions to equations.

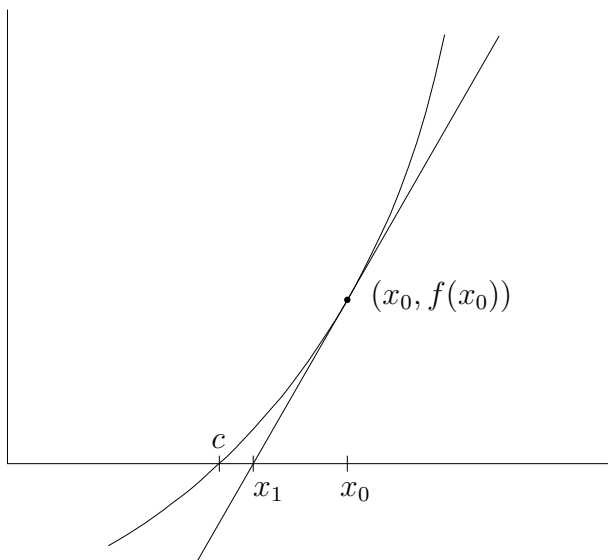
Actually, in this day and age, all anyone ever needs to do is use technology. Software, such as behind the Wolfram Alpha web site ([www.wolframalpha.com](http://www.wolframalpha.com)), or programmed into graphing calculators, will find roots to high precision (8 or more decimal places).

But our goal here is understanding: How does such technology work?

One simple algorithm is the bisection algorithm described in my notes for section 2.5. This technique works well, and is easy to program on a computer. But it is fairly slow—it takes dozens of steps to achieve high precision, and it would be tedious to do by hand. Here, we will learn of a simple but powerful method for finding high precision approximations to solutions of many equations. It was the method of choice for finding solutions of equations by hand, in the era before computer technology became pervasive (before the 1980s). It is now still widely used in software.

**Newton's method.** This is a method for solving equations of the form  $f(x) = 0$ , where  $f$  is differentiable. In Newton's method, you begin with a reasonable guess as to the solution of the equation, say  $x_0$ . Then you perform a certain calculation to obtain a better approximation,  $x_1$ . Then you do another such calculation to obtain the next approximation,  $x_2$ . And then you do another such calculation to obtain the next approximation  $x_3$ . Typically (but not always), these successive approximations rapidly converge to the true solution.

The recipe for finding the next approximation is simple, geometrically. Suppose your initial guess is  $x_0$ . To find the next approximation  $x_1$ , what you do is look at the curve  $y = f(x)$ . Find the tangent line at the point  $(x_0, f(x_0))$ , and see where it crosses the  $x$ -axis. Call that point  $x_1$ . Find the diagram below, it seems reasonable to believe that  $x_1$  will be closer to the true solution  $x = c$ .



All we need is a formula for  $x_1$ . The tangent line has slope  $m = f'(x_0)$ , so the tangent

line is

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

or

$$y - f(x_0) = f'(x_0)(x - x_0)$$

or

$$y = f'(x_0)(x - x_0) + f(x_0).$$

Now the point  $x = x_1$  is supposed to be where the tangent line crosses the  $x$ -axis. So set  $y = 0$  and solve for  $x$  in the equation of the tangent line. That is, solve

$$0 = f'(x_0)(x - x_0) + f(x_0)$$

for  $x$ . We get

$$f'(x_0)(x - x_0) = -f(x_0)$$

or

$$x - x_0 = -\frac{f(x_0)}{f'(x_0)}$$

so

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

That is, we have

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Now we have our new approximation,  $x_1$ . To find the next approximation  $x_2$ , we do the same thing: Construct the tangent line at  $x = x_1$ , and see where that intersects the  $x$ -axis. This proceeds exactly as above, but with every  $x_0$  replaced by  $x_1$ . This gives

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}.$$

To find the next approximation,  $x_3$ , we use

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}.$$

In general, we begin with an initial guess  $x_0$ , and then for  $n > 0$ , we compute the  $(n+1)$ th approximation from the  $n$ th approximation using:

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

This algorithm is an example of an iterative or recursive algorithm, where a new approximation is computed from the previous approximation via a certain formula.

**Example.** Try  $x^3 + 3x - 7 = 0$ . Here,  $f(x) = x^3 + 3x - 7$  and  $f'(x) = 3x^2 + 3$ . So the iteration

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

becomes

$$x_{n+1} = x_n - \frac{x_n^3 + 3x_n - 7}{3x_n^2 + 3}.$$

We will begin with  $x_0 = 1$ . Then

$$x_1 = x_0 - \frac{x_0^3 + 3x_0 - 7}{3x_0^2 + 3} = 1 - \frac{1^3 + 3(1) - 7}{3(1^2) + 3} = 1.5.$$

So  $x_1 = 1.5$ . We obtain  $x_2$  using

$$x_2 = x_1 - \frac{x_1^3 + 3x_1 - 7}{3x_1^2 + 3} = 1.5 - \frac{1.5^3 + 3(1.5) - 7}{3(1.5^2) + 3} = 1.410256.$$

Now we obtain  $x_3$  using

$$x_3 = x_2 - \frac{x_2^3 + 3x_2 - 7}{3x_2^2 + 3} = 1.410256 - \frac{1.410256^3 + 3(1.410256) - 7}{3(1.410256^2) + 3} = 1.406295.$$

Next we get

$$x_4 = x_3 - \frac{x_3^3 + 3x_3 - 7}{3x_3^2 + 3} = 1.406295 - \frac{1.406295^3 + 3(1.406295) - 7}{3(1.406295^2) + 3} = 1.406288.$$

We notice that we have the same approximation as given earlier to the root of the equation, using just four steps of this algorithm. If we “turn the crank” one more time, we discover that  $x_5$  is the same number as  $x_4$  (to six decimal places). This is how you tell if you are done: you reach a point where the new approximation is the same as the old approximation (to the number of decimals you are using). The approximation should be accurate to that many decimal places.

**Example.** Find the cube root of 5. This number is the solution of  $x^3 = 5$ . Let  $f(x) = x^3 - 5$ . Now  $f'(x) = 3x^2$  and the iteration is

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

or

$$x_{n+1} = x_n - \frac{x_n^3 - 5}{3x_n^2}.$$

Now start with the guess  $x_0 = 2$ . Then

$$x_1 = x_0 - \frac{x_0^3 - 5}{3x_0^2} = 2 - \frac{2^3 - 5}{3(2^2)} = 1.75.$$

Then

$$x_2 = x_1 - \frac{x_1^3 - 5}{3x_1^2} = 2 - \frac{1.75^3 - 5}{3(1.75^2)} = 1.710884.$$

If we keep going, we find

$$x_3 = 1.709976$$

$$x_4 = 1.709976$$

and we see we already have the solution accurate to the precision shown.

**Example.** An algorithm to find square roots: To find  $\sqrt{a}$ , we would use  $f(x) = x^2 - a$ . So the solution to  $x^2 = a$  or  $x^2 - a = 0$  is approximated by the iteration

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

or (with a little algebra)

$$x_{n+1} = \frac{1}{2} \left( x_n + \frac{a}{x_n} \right).$$

So to find  $\sqrt{26}$ , we would use

$$x_{n+1} = \frac{1}{2} \left( x_n + \frac{26}{x_n} \right).$$

Beginning with  $x_0 = 5$ , we get

$$x_1 = \frac{1}{2} \left( x_0 + \frac{26}{x_0} \right) = \frac{1}{2} \left( 5 + \frac{26}{5} \right) = 5.1$$

$$x_2 = \frac{1}{2} \left( x_1 + \frac{26}{x_1} \right) = \frac{1}{2} \left( 5.1 + \frac{26}{5.1} \right) = 5.099020$$

$$x_3 = 5.099020$$

$$x_4 = 5.099020$$

and we already have six decimals after just three steps.

**Example.** Now we try the equation  $\cos x = x$ . This is  $\cos x - x = 0$  so let  $f(x) = \cos x - x$ . Then  $f'(x) = -\sin x - 1$  and the iteration becomes

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

Starting with  $x_0 = 0.75$  (a plausible guess from looking at a graph of the cosine function), we obtain

$$x_1 = x_0 + \frac{\cos x_0 - x_0}{\sin x_0 + 1} = 0.75 + \frac{\cos 0.75 - 0.75}{\sin 0.75 + 1} = 0.7391111$$

$$x_2 = 0.7390851$$

$$x_3 = 0.7390851$$

so evidently 0.7390851 is the solution to  $\cos x = x$ . (Here  $x$  is in radians.)

Newton's method doesn't always work so well. If you choose a poor initial guess  $x_0$ , the method might take a long time to converge to the solution. If the equation has more than one solution, you need to be careful with your choice of  $x_0$  to find the desired solution. But sometimes, the graph  $y = f(x)$  curves in such a way that a poorly chosen  $x_0$  might lead to subsequent approximations going wildly astray. This can happen if  $f'(x_0)$  is very small; or worse,  $f'(x_0) = 0$ , in which case,  $x_1$  isn't even defined. The following example illustrates what can happen.

**Example.** Solve  $\sqrt[3]{x} = 0$  using Newton's method. Of course, the solution is 0, so we wouldn't bother doing this in real life. But here's what happens. Here,  $f(x) = \sqrt[3]{x} - 0 = \sqrt[3]{x}$  and  $f'(x) = 1/(3\sqrt[3]{x^2})$ . So the iteration is

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

which simplifies to just

$$x_{n+1} = -2x_n$$

So if we start with  $x_0 = 1$ , we get  $x_1 = -2$ ,  $x_3 = 4$ ,  $x_4 = -8$ , and so on.