Solutions to Worksheet on Deriving Root-finding Algorithms

Secant method. The formula looks like the Newton method, but with the tangent slope $m = f'(x_n)$ replaced by the secant slope $m = (f(x_n) - f(x_{n-1}))/(x_n - x_{n-1})$:

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

In applying this formula we would have to check that two consecutive f-values were not the same: $f(x_n) - f(x_{n-1}) \neq 0$. (*What would you do if this check fails?*)

Method of "false position". The formula is very similar to the secant method:

$$c_{n+1} = a_n - \frac{f(a_n)(b_n - a_n)}{f(b_n) - f(a_n)}$$

Again we would have to check that two consecutive f-values were not the same. (*What would you do if this check fails?*) In any case we need to use c_{n+1} to update the bracket, just as in the bisection method:

if $f(c_{n+1}) == 0$

    stop

else if $f(c_{n+1})f(a_n) > 0$

    $a_{n+1} = c_{n+1}, \quad b_{n+1} = b_n$

else

    $a_{n+1} = a_n, \quad b_{n+1} = c_{n+1}$

Halley's method. One sets-up a quadratic polynomial,

$$p(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(x_n)}{2}(x - x_n)^2,$$

and solves this equation for x_{n+1}:

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

Actually it is most convenient to solve for "$x_{n+1} - x_n$"; in any case use the quadratic formula:

$$x_{n+1} - x_n = \frac{-f'(x_n) \pm \sqrt{(f'(x_n))^2 - 4(f''(x_n)/2)f(x_n)}}{2(f''(x_n)/2)}.$$

Simplified slightly we get:

$$(*) \qquad x_{n+1} = x_n - \frac{f'(x_n) \mp \sqrt{(f'(x_n))^2 - 2f''(x_n)f(x_n)}}{f''(x_n)}.$$

Regarding the choice of plus or minus at each step, one should go to the root that is closest. Thus one chooses plus or minus so as to make the numerator of the fraction in $(*)$ smaller in absolute value. But there are other issues. The formula fails if $f''(x_n) = 0$; in this case it makes sense to revert to a step of Newton's method. (*The function is nearly linear near x_n in that case.*) Also there is no (real) solution to the quadratic if $(f'(x_n))^2 < 2f''(x_n)f(x_n)$. One can sketch that case to see the problem geometrically. Other strategies must be tried in that case.

The benefit of Halley's method is that it is *cubically* convergent, so the number of correct digits *triples* at each iteration. However, it is not actually an improvement over Newton's method because the amount of computation per step is larger, and because the user of Halley's method has to compute both the first *and second* derivatives of f.