Additional rootfinding methods. We briefly describe some other iteration methods, which may be superior to the Newton or secant methods in some situations. We begin with three methods based on quadratic polynomial approximations, as with Muller's method.

The first and most straightforward method uses the quadratic Taylor approximation. Expand $f(\alpha)$ about x_0, to obtain

$$0 = f(x_0)+(\alpha-x_0)f'(x_0)+\frac{(\alpha-x_0)^2}{2}\cdot f''(x_0)+\frac{(\alpha-x_0)^3}{6}\cdot f^{(3)}(\xi_0), \qquad (2.4.5)$$

with ξ_0 between α and x_0. Dropping the error term, solve

$$0 = f(x_0) + wf(x_0) + \frac{w^2}{2}f''(x_0) \qquad (2.4.6)$$

We choose the smallest root in magnitude, and then define

$$x_1 = x_0 + w \tag{2.4.7}$$

Iterate this process in the obvious way, replacing x_0, x_1 by x_n, x_{n+1}. If $f'(\alpha)\neq 0$ and if $f(x)$ is sufficiently differentiable about α, then this iteration method can be shown to have a cubic order of convergence. Disadvantages to the method include the need to calculate $f''(x)$ and the need to calculate a square root, at each step.

Our second and third methods involve using the inverse function to $f(x)$. For this, assume $f'(\alpha)\neq 0$. Let $x=h(y)$ be the inverse to $y=f(x)$, defined for all x in some interval about α. Then by definition,

$$y = f(h(y)), \qquad y \text{ near } 0=f(\alpha), \tag{2.4.8}$$
$$x = h(f(x)), \qquad x \text{ near } \alpha.$$

For the derivatives of $h(y)$, differentiate (2.4.8) to obtain

$$h'(y) = 1/f'(x), \quad h''(y) = -f''(x)/[f'(x)]^3 \tag{2.4.9}$$

with $x=h(y)$.

For our second method, expand $h(y)$ about $y_0=f(x_0)$:

$$h(y)=h(y_0)+(y-y_0)h'(y_0)+\frac{(y-y_0)^2}{2}h''(t_0)+\frac{(y-y_0)^3}{6}h^{(3)}(\zeta),$$

with ζ between y and y_0. Let $y=0$ and use $\alpha=h(0)$ to obtain

$$\alpha = h(y_0) - y_0h'(y_0) + \tfrac{1}{2}y_0^2h''(y_0) - \tfrac{1}{6}y_0^3h^{(3)}(\zeta_0), \tag{2.4.10}$$

ζ_0 between 0 and y_0. Drop the error term to obtain an improved approximation to α. Also use (2.4.8), (2.4.9) to convert to formulas using $f(x)$ and x_0. This yields the new approximation

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{[f(x_0)]^2f''(x_0)}{2[f'(x_0)]^3} \tag{2.4.11}$$

Iterate this formula, replacing x_0, x_1 by x_n, x_{n+1}. We leave it to a problem to show that (2.4.10) can be used to prove a cubic order of convergence for (2.4.11).

Our third method is based on using quadratic interpolation to $x=h(y)$, regarding y as the independent variable. The quadratic polynomial interpolating $h(y)$ at y_0, y_1, y_2 is

$$q(y) = h(y_2) + (y-y_2)h[y_2,y_1] + (y-y_2)(y-y_1)h[y_2,y_1,y_0],$$

following the earlier formula (2.4.1) used in defining Muller's method. We are assuming x_0, x_1, x_2 are approximations to α, and we let $y_i=f(x_i)$. Setting $y=0$ and assuming $y(0) \doteq h(0)=\alpha$, we have

$$x_1 = h(y_2) - y_2 h[y_2,y_1] + y_2 y_1 h[h_2,y_1,y_0] \qquad (2.4.12)$$

We leave to a problem the conversion of this to a formula involving only $f(x)$ and the points x_0, x_1, x_2. The error analysis requires the use of the error formula for quadratic interpolation, given in §3.1; and it follows the same type of development as used in the convergence analysis of the secant method. The order of convergence is the same as for Muller's method, $p \doteq 1.84$.

Another class of methods are based on the formula

$$x_{n+1} = x_n - \frac{f(x_n)}{a_n} \qquad (2.4.13)$$

where a_n is chosen in some way other than that for the Newton and secant methods. For example, we may choose an a_n that is not updated or changed with each new iterate. The following three methods are examples of (2.4.13).

Suppose we update Newton's method only every second iteration. This leads to

$$z_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad x_{n+1} = z_{n+1} - \frac{f(z_{n+1})}{f'(x_n)}, \quad n \geq 0 \qquad (2.4.14)$$

The iterates x_n will converge to α with order p=3. The cost of producing each iterate x_n is two evoluations of f and one of f'. Whether or not this method is preferable to the regular Newton's method will depend to some extent on the number of equations being solved. With systems of nonlinear equations, the generalization of (2.4.14) gains an advantage over Newton's method. This will be taken up again in (2.11.9) of §2.11, when Newton's method for systems is discussed.

Using the idea implicit in (2.4.14), we update the secant methods derivative approximation a_n every second step. This leads to the iteration formula

$$z_{n+1} = x_n - \frac{f(x_n)}{f[z_n, x_n]}, \quad x_{n+1} = z_n - \frac{f(z_{n+1})}{f[z_n, x_n]}, \quad n \geq 0. \tag{2.4.15}$$

The order of convergence of x_n to α is $p=1+\sqrt{2} \doteq 2.4$, under suitable smoothness assumptions on f(x) and with $f'(\alpha) \neq 0$. The cost of each iteration is two evaluations of f. Thus this is a strong competitor to Newton's method, which has order of convergence p=2, with function evaluations of f and f' for each iterate. For a discussion of further extensions of (2.4.14) and (2.4.15), where a_n is fixed for $m \geq 2$ steps, see Potra-Ptak (1984, pp. 112-119).

Our last method is somewhat similar in spirit to the last two methods, but is best treated as a new method. Given x_0, z_0 as estimates of α, define

$$\begin{aligned} x_{n+1} &= x_n - f(x_n) \div f'((x_n + z_n)/2), \\ z_{n+1} &= x_{n+1} - f(x_{n+1}) \div f'((x_n + z_n)/2) \end{aligned} \tag{2.4.16}$$

The rate of convergence to α of each of the sequences $\{x_n\}$ and $\{z_n\}$ is $p=1+\sqrt{2}$. The cost is one evaluation of f(x) and one evaluation of $f'(x)$. Thus its cost is the same as for Newton's method, but it converges more rapidly. For an analysis of this method and generalizations, see Werner (1982).