Given a continuous function \( f(x) \), a value \( x = w \) for which \( f(w) = 0 \) is called a root or zero of \( f \) and is a solution to the equation

\[
f(x) = 0.
\]

We exclude the case where \( f(x) = ax + b, a \neq 0 \), because the solution is \( x = -b/a \) and can be computed directly from the data describing \( f \). This is the linear case. (Compare, e.g., to the case \( f(x) = x^3 + 17 \).

This problem arises in (at least) two natural ways: (i) If we have two functions \( g(x) \) and \( h(x) \), it is of interest to know when \( g(x) = h(x) \). In this case we have a root problem for \( f(x) = g(x) - h(x) \), an example: \( g(x) = e^{-x} \) and \( h(x) = \sin(x) \); (ii) We have a function \( F(x) \) and we want to find where it is minimized or maximized. In this case we have a root problem for \( f(x) = F'(x) \).

All the methods we study share the feature that they “generate” a sequence of approximations \( P_0, P_1, \ldots \) that is intended to converge to a root \( w \) of \( f \) (by continuity, \( f(P_n) \rightarrow f(w) = 0 \) as \( n \rightarrow \infty \)).

1. Method 1 - Bisection: The method starts (STEP 0) with an interval \( I_0 = (u_0, v_0) \), \( u_0 < v_0 \), and \( f \) has opposite signs at the endpoints; thus \( f(u_0)f(v_0) < 0 \). By the intermediate value theorem, \( f \) has a root \( w \in I_0 \). We bisect \( I_0 \) with the midpoint, \( P_0 = (u_0 + v_0)/2 \). This is the initial approximation to \( w \). If \( f(P_0) = 0 \) we STOP. Otherwise we continue into the next step, STEP 1, with one of the halves (i) \( I_1 = (u_0, P_0) \) if \( f(u_0)f(P_0) < 0 \) or else (ii) \( I_1 = (P_0, v_0) \) if \( f(P_0)f(v_0) < 0 \) (Precisely one of these two situations must hold - WHY?). Clearly \( |I_1| = \frac{1}{2}|I_0| = (v_0 - u_0)/2 \) (\( |I| = v - u \) denotes the length of the interval \( I = (u, v) \)).

In STEP \( n > 0 \) we have (from the previous step) an interval \( I_n = (u_n, v_n) \), and \( f \) has opposite signs at the endpoints \( f(u_n)f(v_n) < 0 \). By the intermediate value theorem, \( f \) has a root \( w \in I_n \). We bisect \( I_n \) with

\[
P_n = (u_n + v_n)/2. \tag{1}
\]

If \( f(P_n) = 0 \) we STOP. Otherwise we continue into the next step, STEP \( n + 1 \), with one of the halves (i) \( I_{n+1} = (u_n, P_n) \) if \( f(u_n)f(P_n) < 0 \) or else (ii) \( I_{n+1} = (P_n, v_n) \) if \( f(P_n)f(v_n) < 0 \) (Again, precisely one of these two situations must hold). Clearly \( |I_{n+1}| = \frac{1}{2}|I_n| = (v_n - u_n)/2 \).

- Let \( e_n = P_n - w \) denote the error if we stop at STEP \( n \) and take \( P_n \), the \( n^{th} \) bisection, as an approximation of the root \( w \). Notice that \( |e_n| < |I_n|/2 \) because \( P_n \) and \( w \) are in the same half of \( I_n \). Clearly \( |I_n|/2 = (|I_{n-1}|/2)/2 = \cdots = |I_0|/2^n \rightarrow 0 \) as \( n \rightarrow \infty \). This proves that the bisection method converges when started correctly.

- We can know in advance how many bisections steps will assure a suitably small error. Given \( \varepsilon > 0 \), suppose it is required that \( |e_n| < \varepsilon \) if we stop at STEP \( n \). Then from \( |e_n| < (v_0 - u_0)/2^{n+1} \), we deduce that \( n > \log_2((v_0 - u_0)/\varepsilon) - 1 \) steps are sufficient. In a computer implementation of the bisection method, we might also like to require that \( |f(P_n)| \) is small before we accept \( P_n \) as a suitable approximation to \( w \).
2. Method 2 - Regula-Falsi Suppose $u_n < v_n$ and $f(u_n)f(v_n) < 0$. We will use more information about $f$ than the mere fact that it has opposite signs at the endpoints of $I_n = (u_n, v_n)$. Motivated by the observation that when $I_n$ is small enough, $f$ “looks like” a straight line on this interval, we divide $I_n$ by the point where the line through $A = (u_n, f(u_n))$ and $B = (v_n, f(v_n))$ meets the x-axis. This is the point whose x-coordinate is

$$P_n = \frac{u_n f(v_n) - v_n f(u_n)}{f(v_n) - f(u_n)}.$$  \hspace{1cm} (2)

Regula-falsi IS bisection except that it uses the above instead of $P_n = (u_n + v_n)/2$.

- Regula-falsi converges if it is started correctly, but not because $|I_n| \to 0$ (simple examples show this statement to be false). This underlies the problem with using regula-falsi in practice - at what step, $n$, should it be stopped? Since $|I_n|$ may remain large, we can only stop when $|f(P_n)|$ is small but unfortunately, this is no guarantee that $e_n$ is small.

- You should study handout 1 (through the homepage) - “Informative Traces of Bisection and Regula-Falsi”.

3. Fixed Point Iteration A value $x = u$ is a fixed point of a function $h(x)$ if $h(u) = u$. Fixed points are thus the x-coordinates of the points where the graph of $h$ meets the line $y = x$. There is a beautiful algorithm to find fixed points. It is called fixed point iteration (FPI), or functional iteration:

- Guess $P_0$
- $n \leftarrow 0$
- WHILE $P_n \neq h(P_n)$ DO
  - $P_{n+1} \leftarrow h(P_n)$
  - $n \leftarrow n + 1$
- ENDWHILE
- RETURN $P_n$ (it is a fixed point)

We might hope that $P_n \to w$ but we should not expect it to stop in a finite number of steps with $P_n = h(P_n)$. To stop the above algorithm in practice, we would require $|P_n - h(P_n)|$ to be small, say less than $\varepsilon$. The condition in the WHILE would then be WHILE $|P_n - h(P_n)| \geq \varepsilon$ DO. We then return $P_n$, an approximate fixed point, after $n$ steps.

(a) Contraction mapping Principle: A function $h(x)$ is a contraction on an interval $I = (a, b)$ if there is a constant $k < 1$ such that for all pairs $u, v \in (a, b)$,

$$|h(u) - h(v)| \leq k|u - v|;$$

ie., $h(u)$ and $h(v)$ are closer than $u$ and $v$ were. Therefore application of $h$ “contracts”, or brings function values closer than their arguments were. The mean value theorem implies that $h$ is a contraction if $|h'(x)| \leq k$ for all $x \in (a, b)$, some $k < 1$.

The contraction mapping principle states that if (A) $h(w) = w$, (B) $h$ is a contraction on an interval $I = (w - \delta, w + \delta)$ for some $\delta > 0$, and (C) $P_0 \in I$, then $P_n \to w$ (in other
words, the FPI algorithm above produces approximations \( P_n = h(P_{n-1}) \) that converge to a fixed point \( w = h(w) \). In fact if we knew that some \( P_j \in I \) that is enough in condition C), since we could just (re)start the iterations at \( P_j \).

Sometimes it is difficult to find an interval \( I \) satisfying condition (B). An alternative version of the theorem uses condition (B'), “\( h \) is a contraction on an interval \( I \) that contains the fixed point \( w \) and satisfies the condition that \( h(x) \in I \) whenever \( x \in I \).”

(b) **Relevance to Root-Finding:** Suppose we want to find roots of \( f(x) \). Define

\[
g(x) = x - \phi(x)f(x),
\]

where (i) \( \phi \) is continuous and (ii) \( \phi(x) = 0 \) implies \( f(x) = 0 \). Clearly \( g(w) = w \) if and only if \( f(w) = 0 \); i.e., the roots of \( f \) are the fixed points of \( g \). Our approach will be to specify the function \( \phi(x) \) in (3) and then do FPI on the resulting \( g(x) \):

\[
P_{n+1} \leftarrow g(P_n).
\]

Each different way we choose \( \phi(x) \) in (3) and apply FPI to the resulting \( g(x) \) gives a new root-finding method for \( f(x) \) [trite example: \( \phi(x) = 1 \)]. If \( P_n \to w = g(w) \), this FPI has produced a root-finding method that converged to a root of \( f(x) \); i.e., it “worked”.

(c) **Convergence Rate of FPI:** If FPI converges, \( P_n \to w = g(w) \), so the errors \( e_n \equiv P_n - w \to 0 \). The question is how rapidly?

Since \( P_{n+1} = g(P_n) \) (def. of FPI) and \( w = g(w) \) (def. of fixed point),

\[
|e_{n+1}| = |P_{n+1} - w| = |g(P_n) - g(w)|. \tag{4}
\]

Applying the mean value theorem [see also Taylor’s theorem, \( n = 0 \) (Course Notes 3, eq (8))], there is a point \( \theta_n \) between \( P_n \) and \( w \) for which \( g(P_n) - g(w) = g'(\theta_n)(P_n - w) \).

Using this in (4), and assuming \( g' \) is continuous,

\[
\frac{|e_{n+1}|}{|e_n|} = |g'(\theta_n)| \to |g'(w)|. \tag{5}
\]

I. Assuming \( |g'(w)| \neq 0 \) (and we may assume it is < 1), \( |g'(w)| \) is the fraction by which \( |e_n| \) is reduced if we take one more FPI step and stop with \( e_{n+1}, n \) large. This is linear convergence, where - in the limit - errors are reduced by a fixed fraction in each step.

II. If \( g'(w) = 0 \) both numerator and denominator of the ratio in (5) converge to zero, but the numerator converges strictly faster. In this case Taylors theorem, \( n = 1 \), shows (since \( g'(w) = 0 \)) that \( g(P_n) - g(w) = \frac{1}{2}g''(\theta_n)(P_n - w)^2 \) so using (4), and assuming the continuity of \( g'' \),

\[
\frac{|e_{n+1}|}{|e_n|^2} = \frac{1}{2}|g''(\theta_n)| \to \frac{1}{2}|g''(w)|. \tag{6}
\]

Assuming \( g''(w) \neq 0 \) the error on the next step is about \( |g''(w)|/2 \) times the square of the current error, \( n \) large. This is quadratic convergence. In general, the order of convergence \( k \), of FPI, is defined by

\[
k = \min (j > 0 : g^{(j)}(w) \neq 0); \tag{3}
\]
order \( k = 1 \) is linear convergence, order 2 is quadratic, etc. If the order of convergence is \( k \) and \( g^{(k)} \) is continuous, then
\[
\frac{e_{n+1}}{e_n} = \frac{1}{k!} |g^{(k)}(\theta_n)| \to \frac{1}{k!} |g^{(k)}(w)|,
\]
a non-zero constant.

4. **Method 3 - Chord Method:** There is a parameter \( m \neq 0 \) for which we choose a fixed, constant value. Using \( \phi(x) = 1/m \) in (3), do FPI on \( g(x) = x - f(x)/m \). Thus
\[
P_{n+1} = P_n - \frac{1}{m} f(P_n) = g(P_n). \tag{7}
\]
Rearranging the above expression we see that
\[
m = \frac{f(P_n) - 0}{P_n - P_{n+1}}
\]
so the chord method chooses \( P_{n+1} \) as the x-coordinate of the point where the line of slope \( m \) through \((P_n, f(P_n))\) meets the x-axis.

- **convergence:** For the chord method \( |g'(x)| = |1 - f'(x)/m| \). Thus we know that if \( w \) is a root of \( f \) and if \( 0 < f'(x)/m < 2 \) for all values of \( x \in I = (w - \delta, w + \delta) \), then iterations in (7) will converge as long as \( P_0 \in I \) (in fact if we knew that some \( P_j \in I \) that is enough, since we just (re)start the iterations at \( P_j \)).

- **convergence rate:** Suppose the iterations in (7) converge. Since \( g'(w) = 1 - f'(w)/m = 0 \) only if \( m = f'(w) \), we conclude that the chord method is linear except for a single choice of \( m \) as \( f'(w) \), in which (lucky) case it has at least a quadratic convergence rate.

5. **Method 4 - Newton’s Method:** Take \( \phi(x) = 1/f'(x) \) in (3) and do FPI on \( g(x) = x - f(x)/f'(x) \). Thus
\[
P_{n+1} = P_n - \frac{f(P_n)}{f'(P_n)} = g(P_n). \tag{8}
\]
Rearranging the above expression we see that
\[
f'(P_n) = \frac{f(P_n) - 0}{P_n - P_{n+1}}
\]
so Newton’s method chooses \( P_{n+1} \) as the x-coordinate of the point where the tangent line to \( f \) at \( x = P_n \) meets the x-axis.

- **convergence:** For Newton’s method
\[
g'(x) = \frac{f(x)f''(x)}{(f'(x))^2}.
\]
If (i) \( f'' \) is continuous, (ii) \( f(w) = 0 \), and (iii) \( f'(w) \neq 0 \) then \( g'(w) = 0 \) and \( g' \) is continuous. Therefore there is an interval \( I = (w - \delta, w + \delta) \) on which \( |g'(x)| < 1 \). This proves that Newton’s method converges if \( P_0 \) is close enough to \( w \) (unfortunately it is hard in some cases to know precisely what “close enough” means). This convergence result is still true when \( f'(w) = 0 \) (i.e., (iii) fails and we have a tangency root), but the proof argument used above no longer works.
• **convergence rate:** Suppose the iterations in (8) converge and that \( f'(w) \neq 0 \). The equation above shows \( g'(w) = 0 \), so in the case of a non-tangency root, Newton’s method is at least quadratic. It is not difficult to show that when Newton’s method converges to a tangency root \( w \) (i.e., \( f(w) = 0 \) and \( f'(w) = 0 \)), the rate is linear.

6. **Secant Method:** If we don’t know \( f' \) but still want to use Newton’s method, we could replace \( f'(P_n) \) in (8) by the approximation

\[
f'(P_n) \approx \frac{f(P_n) - f(P_{n-1})}{P_n - P_{n-1}}.
\]

This gives the iteration for the secant method,

\[
P_{n+1} = \frac{P_{n-1}f(P_n) - P_nf(P_{n-1})}{f(P_n) - f(P_{n-1})}, \quad n \geq 1.
\]

(9)

It is *not* a fixed point iteration (in fact, compare (9) with (2)). It needs \( P_0 \) and \( P_1 \) to start, and each iteration is a function of the previous two. \( P_{n+1} \) is the \( x \)-coordinate of the point where the line joining \( A = (P_{n-1}, f(P_{n-1})) \) and \( B = (P_n, f(P_n)) \) meets the \( x \)-axis. When the iterations in (9) converge to a non-tangency root \( w \),

\[
\frac{\varepsilon_{n+1}}{\varepsilon_n} \to c > 0
\]

so its rate is clearly faster than linear but slower than quadratic. In fact it may be shown that \( \varepsilon_{n+1}/\varepsilon_n^{(1+\sqrt{5})/2} \to C > 0 \). The exponent is about 1.618.

7. **Acceleration of Convergence:** Instead of taking \( P_{n+1} = g(P_n) \), as in FPI, we will use \( P'_{n+1} \) as the \( x \)-coordinate of the point where the line joining \( A = (P_{n-1}, g(P_{n-1})) \) and \( B = (P_n, g(P_n)) \) meets the \( y = x \) (looking at the graph of \( g \) near a fixed point shows why this may be a good idea). Using \( P_{n+1} = g(P_n), \ P_n = g(P_{n-1}) \), and a little algebra,

\[
P'_n = P_{n+1} - \frac{(P_{n+1} - P_n)^2}{P_{n+1} - 2P_n + P_{n-1}}.
\]

(10)

\( P'_{n+1} \) is called the acceleration of \( P_{n+1} \). Writing \( \Delta P_j = P_j - P_{j-1} \) and \( \Delta^2 P_j = \Delta(\Delta P_j) = \Delta P_j - \Delta P_{j-1} = P_j - 2P_{j-1} + P_{j-2} \), we get Aitken’s delta-squared formula:

\[
P'_n = P_{n+1} - \frac{(\Delta P_{n+1})^2}{\Delta^2 P_{n+1}}.
\]

\( P'_{n+1} \) may be better than \( P_{n+1} \) because of the following: Suppose \( a_0, a_1, \ldots \) is a sequence of numbers that converges to \( w \) at a linear rate (and \( a_i \neq w \)). Apply the acceleration formula to \( a_2, a_3, \ldots \) (i.e., \( a'_i = a_i - (\Delta a_i)^2/\Delta^2 a_i, i \geq 2 \)) to obtain \( a'_2, a'_3, \ldots \). Then

\[
\frac{|a'_i - w|}{|a_i - w|} \to 0;
\]

i.e., the accelerated sequence converges to the same limit, only faster. There are two main ways to use the acceleration idea.
• **Aitkin’s Method:** $P_n$ denotes the approximations of any linear method (regula-falsi, chord, Newton with a tangency root, etc.). Just accelerate each $P_i$ and stop at step $n$ if $|f(P'_n)| < \varepsilon$ (or if $|P'_n - P'_{n-1}|$ is small).

• **Steffanson’s Method:** The basic method is some linearly converging FPI, like Newton with a tangency root. From $P_0$ we do two FPI steps, $P_1 = g(P_0), P_2 = g(P_1)$. At this point we accelerate $P_2$ by

$$Q_0 = P_2 - \frac{(\Delta P_2)^2}{\Delta^2 P_2}.$$  

The basic iteration starts from $Q_i$. Two FPI steps yield $P_1 = g(Q_i)$ and $P_2 = g(P_1)$ and $Q_{i+1} = P_2 - (\Delta P_2)^2/(\Delta^2 P_2)$ is the acceleration of $P_2$. We stop when $|Q_i - Q_{i-1}| < \varepsilon$. You should study Handout number 3 illustrating the value of acceleration.