We want to find the numbers $x$ for which $f(x) = 0$, with $f$ a given function. Here, we denote such roots or zeroes by the Greek letter $\alpha$. Rootfinding problems occur in many contexts. Sometimes they are a direct formulation of some physical situation; but more often, they are an intermediate step in solving a much larger problem.

**An example with annuities** Suppose you are paying into an account an amount of $P_{in}$ per period of time, for $N_{in}$ periods of time. The amount you are deposited is compounded at an interest rate of $r$ per period of time. Then at the beginning of period $N_{in} + 1$, you will withdraw an amount of $P_{out}$ per time period, for $N_{out}$ periods. In order that the amount you withdraw balance that which has been deposited including interest, what is the needed interest rate? The equation is

$$P_{in} \left[ (1 + r)^{N_{in}} - 1 \right] = P_{out} \left[ 1 - (1 + r)^{-N_{out}} \right]$$

We assume the interest rate $r$ holds over all $N_{in} + N_{out}$ periods.
As a particular case, suppose you are paying in $P_{in} = \$1,000$ each month for 40 years. Then you wish to withdraw $P_{out} = \$5,000$ per month for 20 years. What interest rate do you need? If the interest rate is $R$ per year, compounded monthly, then $r = R/12$. Also, $N_{in} = 40 \cdot 12 = 480$ and $N_{out} = 20 \cdot 12 = 240$. Thus we wish to solve

$$1000 \left[ \left( 1 + \frac{R}{12} \right)^{480} - 1 \right] = 5000 \left[ 1 - \left( 1 + \frac{R}{12} \right)^{-240} \right]$$

What is the needed yearly interest rate $R$? The answer is 2.92%. How did I obtain this answer?

This example also shows the power of compound interest.
THE BISECTION METHOD

Most methods for solving \( f(x) = 0 \) are iterative methods. We begin with the simplest of such methods, one which most people use at some time.

We assume we are given a function \( f(x) \); and in addition, we assume we have an interval \([a, b]\) containing the root, on which the function is continuous. We also assume we are given an error tolerance \( \varepsilon > 0 \), and we want an approximate root \( \tilde{\alpha} \) in \([a, b]\) for which

\[
|\alpha - \tilde{\alpha}| \leq \varepsilon
\]

We further assume the function \( f(x) \) changes sign on \([a, b]\), with

\[
f(a) f(b) < 0
\]
Algorithm **Bisect**(*f*, *a*, *b*, *ε*). **Step 1**: Define

\[ c = \frac{1}{2} (a + b) \]

**Step 2**: If \( b - c \leq \varepsilon \), accept \( c \) as our root, and then stop.

**Step 3**: If \( b - c > \varepsilon \), then check compare the sign of \( f(c) \) to that of \( f(a) \) and \( f(b) \). If

\[ \text{sign}(f(b)) \cdot \text{sign}(f(c)) \leq 0 \]

then replace \( a \) with \( c \); and otherwise, replace \( b \) with \( c \). Return to **Step 1**.

Denote the initial interval by \([a_1, b_1]\), and denote each successive interval by \([a_j, b_j]\). Let \( c_j \) denote the center of \([a_j, b_j]\). Then

\[ |\alpha - c_j| \leq b_j - c_j = c_j - a_j = \frac{1}{2} (b_j - a_j) \]

Since each interval decreases by half from the preceding one, we have by induction,

\[ |\alpha - c_n| \leq \left(\frac{1}{2}\right)^n (b_1 - a_1) \]
EXAMPLE Find the largest root of

\[ f(x) \equiv x^6 - x - 1 = 0 \]

accurate to within \( \epsilon = 0.001 \). With a graph, it is easy to check that \( 1 < \alpha < 2 \). We choose \( a = 1, \ b = 2 \); then \( f(a) = -1, \ f(b) = 61 \), and the requirement \( f(a) f(b) < 0 \) is satisfied. The results from \( \text{Bisect} \) are shown in the table. The entry \( n \) indicates the iteration number \( n \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>( a )</th>
<th>( b )</th>
<th>( c )</th>
<th>( b - c )</th>
<th>( f(c) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0000</td>
<td>2.0000</td>
<td>1.5000</td>
<td>0.5000</td>
<td>8.8906</td>
</tr>
<tr>
<td>2</td>
<td>1.0000</td>
<td>1.5000</td>
<td>1.2500</td>
<td>0.2500</td>
<td>1.5647</td>
</tr>
<tr>
<td>3</td>
<td>1.0000</td>
<td>1.2500</td>
<td>1.1250</td>
<td>0.1250</td>
<td>-0.0977</td>
</tr>
<tr>
<td>4</td>
<td>1.1250</td>
<td>1.2500</td>
<td>1.1875</td>
<td>0.0625</td>
<td>0.6167</td>
</tr>
<tr>
<td>5</td>
<td>1.1250</td>
<td>1.1875</td>
<td>1.1562</td>
<td>0.0312</td>
<td>0.2333</td>
</tr>
<tr>
<td>6</td>
<td>1.1250</td>
<td>1.1562</td>
<td>1.1406</td>
<td>0.0156</td>
<td>0.0616</td>
</tr>
<tr>
<td>7</td>
<td>1.1250</td>
<td>1.1406</td>
<td>1.1328</td>
<td>0.0078</td>
<td>-0.0196</td>
</tr>
<tr>
<td>8</td>
<td>1.1328</td>
<td>1.1406</td>
<td>1.1367</td>
<td>0.0039</td>
<td>0.0206</td>
</tr>
<tr>
<td>9</td>
<td>1.1328</td>
<td>1.1367</td>
<td>1.1348</td>
<td>0.0020</td>
<td>0.0004</td>
</tr>
<tr>
<td>10</td>
<td>1.1328</td>
<td>1.1348</td>
<td>1.1338</td>
<td>0.00098</td>
<td>-0.0096</td>
</tr>
</tbody>
</table>
Recall the original example with the function.

\[ f(r) = P_{in} \left[ (1 + r)^{N_{in}} - 1 \right] - P_{out} \left[ 1 - (1 + r)^{-N_{out}} \right] \]

Checking, we see that \( f(0) = 0 \). Therefore, with a graph of \( y = f(r) \) on \([0, 1]\), we see that \( f(x) < 0 \) if we choose \( x \) very small, say \( x = .001 \). Also \( f(1) > 0 \). Thus we choose \([a, b] = [.001, 1]\). Using \( \varepsilon = .000001 \) yields the answer

\[ \tilde{\alpha} = .02918243 \]

with an error bound of

\[ |\alpha - c_n| \leq 9.53 \times 10^{-7} \]

for \( n = 20 \) iterates. We could also have calculated this error bound from

\[ \frac{1}{2^{20}} (1 - .001) = 9.53 \times 10^{-7} \]
Suppose we are given the initial interval \([a, b] = [1.6, 4.5]\) with \(\varepsilon = .00005\). How large need \(n\) be in order to have

\[|\alpha - c_n| \leq \varepsilon\]

Recall that

\[|\alpha - c_n| \leq \left(\frac{1}{2}\right)^n (b - a)\]

Then ensure the error bound is true by requiring and solving

\[\left(\frac{1}{2}\right)^n (b - a) \leq \varepsilon\]

\[\left(\frac{1}{2}\right)^n (4.5 - 1.6) \leq .00005\]

Dividing and solving for \(n\), we have

\[n \geq \log\left(\frac{2.9}{.00005}\right) = 15.82\]

Therefore, we need to take \(n = 16\) iterates.
ADVANTAGES AND DISADVANTAGES

Advantages: 1. It always converges.
2. You have a guaranteed error bound, and it decreases with each successive iteration.
3. You have a guaranteed rate of convergence. The error bound decreases by $\frac{1}{2}$ with each iteration.

Disadvantages: 1. It is relatively slow when compared with other rootfinding methods we will study, especially when the function $f(x)$ has several continuous derivatives about the root $\alpha$.
2. The algorithm has no check to see whether the $\varepsilon$ is too small for the computer arithmetic being used. [This is easily fixed by reference to the machine epsilon of the computer arithmetic.]

We also assume the function $f(x)$ is continuous on the given interval $[a, b]$; but there is no way for the computer to confirm this.