Root Finding

- Bisection method
- False-position method
In general, we cannot exploit the function, e.g.:

\[ 2^{x^2} - 10x + 1 = 0 \]

and

\[ \cosh \left( \sqrt{x^2 + 1 - e^x} \right) + \log |\sin x| = 0 \]

Note: at times \( \exists \) multiple roots

* e.g., previous parabola and cosine
* we want at least one
* we may only get one (for each search)

Need a general, function-independent algorithm.
The root of a function $f(x)$ ($f: \mathbb{R} \rightarrow \mathbb{R}$) is simply some value $r$ for which the function is zero, that is, $f(r) = 0$.

This topic is broken into two major sub-problems:

There are five techniques which may be used to find the root of a univariate (single variable) function:
- Bisection method
- False-position method
- Newton's method
- Secant method
- Fixed point iteration

Given a vector-valued multivariate function $f(x)$ ($f: \mathbb{R}^n \rightarrow \mathbb{R}^n$), we will focus on a generalization of Newton's method to find a vector of values $r$ such that each of the functions is zero, that is, $f(r) = 0$. 
Intermediate Value Theorem (IVT)

Given a continuous real-valued function $f(x)$ defined on an interval $[a, b]$, then if $y$ is a point between the values of $f(a)$ and $f(b)$, then there exists a point $r$ such that $y = f(r)$.

As an example, consider the function $f(x) = \sin(x)$ defined on $[1, 6]$. The function is continuous on this interval, and the point $0.5$ lies between the values of $\sin(1) \approx 0.841$ and $\sin(6) \approx -0.279$. Thus, there is at least one point $r$ (there may be more) on the interval $[1, 6]$ such that $\sin(r) = 0.5$. In this case, $r \approx 2.61799$. This is shown in Figure 1.

![Graph of sin(x) with point]
The Bisection Method

Using the IVT to Bound a Root

Suppose we have a function $f(x)$ and an interval $[a, b]$ such that either the case that $f(a) > 0$ and $f(b) < 0$ or the case that $f(a) < 0$ and $f(b) > 0$, that is, $f(a)$ and $f(b)$ have opposite signs. Then, the value 0 lies between $f(a)$ and $f(b)$, and therefore, there must exist a point $r$ on $[a, b]$ such that $f(r) = 0$.

We may refine our approximation to the root by dividing the interval into two: find the midpoint $c = (a + b)/2$. In any real world problem, it is very unlikely that $f(c) = 0$, however if we are that lucky, then we have found a root. More likely, if $f(a)$ and $f(c)$ have opposite signs, then a root must lie on the interval $[a, c]$. The only other possibility is that $f(c)$ and $f(b)$ have opposite signs, and therefore the root must lie on the interval $[c, b]$. We may repeat this process numerous times, each time halving the size of the interval.
Example
An example of bisecting is shown in Figure 2. With each step, the midpoint is shown in blue and the portion of the function which does not contain the root is shaded in grey. As the iteration continues, the interval on which the root lies gets smaller and smaller. The first two bisection points are 3 and 4.

Figure 2. The bisection method applied to sin(x) starting with the interval [1, 5].
Bisection Method—Example

Intuitive, like guessing a number $\in [0, 100]$. 
HOWTO

Problem
Given a function of one variable, \( f(x) \), find a value \( r \) (called a root) such that \( f(r) = 0 \).

Assumptions
We will assume that the function \( f(x) \) is continuous.

Tools
We will use sampling, bracketing, and iteration.

Initial Requirements
We have an initial bound \([a, b]\) on the root, that is, \( f(a) \) and \( f(b) \) have opposite signs.

Iteration Process
Given the interval \([a, b]\), define \( c = (a + b)/2 \). Then
- if \( f(c) = 0 \) (unlikely in practice), then halt, as we have found a root,
- if \( f(c) \) and \( f(a) \) have opposite signs, then a root must lie on \([a, c]\), so assign \( b = c \),
- else \( f(c) \) and \( f(b) \) must have opposite signs, and thus a root must lie on \([c, b]\), so assign \( a = c \).
Halting Conditions

There are three conditions which may cause the iteration process to halt:

As indicated, if \( f(c) = 0 \).

1. We halt if both of the following conditions are met:
   - The width of the interval (after the assignment) is sufficiently small, that is \( b - a < \varepsilon \text{step} \).
   - The function evaluated at one of the end point \( |f(a)| \) or \( |f(b)| < \varepsilon \text{abs} \).

2. If we have iterated some maximum number of times, say \( N \), and have not met Condition 1, we halt and indicate that a solution was not found.

3. If we halt due to Condition 1, we state that \( c \) is our approximation to the root. If we halt according to Condition 2, we choose either \( a \) or \( b \), depending on whether \( |f(a)| < |f(b)| \) or \( |f(a)| > |f(b)| \), respectively.

If we halt due to Condition 3, then we indicate that a solution may not exist (the function may be discontinuous).
Figure 1.2: Conditions where the bisection method will work and where it fails
Thus, with the seventh iteration, we note that the final interval, \([1.7266, 1.7344]\), has a width less than 0.01 and \(|f(1.7344)| < 0.01\), and therefore we chose \(b = 1.7344\) to be our approximation of the root.
Thus, after the 11th iteration, we note that the final interval, [3.2963, 3.2968] has a width less than 0.001 and $|f(3.2968)| < 0.001$ and therefore we chose $b = 3.2968$ to be our approximation of the root.
Use bisection to find the root of \( f(x) = x^3 - 10x^2 + 5 = 0 \) that lies in the interval \((0.6, 0.8)\).

**Solution** The best way to implement the method is to use the table shown below. Note that the interval to be bisected is determined by the sign of \( f(x) \), not its magnitude.

<table>
<thead>
<tr>
<th>( x )</th>
<th>( f(x) )</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>1.616</td>
<td>−</td>
</tr>
<tr>
<td>0.8</td>
<td>−0.888</td>
<td>(0.6, 0.8)</td>
</tr>
<tr>
<td>((0.6 + 0.8)/2 = 0.7)</td>
<td>0.443</td>
<td>(0.7, 0.8)</td>
</tr>
<tr>
<td>((0.8 + 0.7)/2 = 0.75)</td>
<td>−0.203</td>
<td>(0.7, 0.75)</td>
</tr>
<tr>
<td>((0.7 + 0.75)/2 = 0.725)</td>
<td>0.125</td>
<td>(0.725, 0.75)</td>
</tr>
<tr>
<td>((0.75 + 0.725)/2 = 0.7375)</td>
<td>−0.038</td>
<td>(0.725, 0.7375)</td>
</tr>
<tr>
<td>((0.725 + 0.7375)/2 = 0.73125)</td>
<td>0.044</td>
<td>(0.73125, 0.7375)</td>
</tr>
<tr>
<td>((0.7375 + 0.73125)/2 = 0.73438)</td>
<td>0.003</td>
<td>(0.73125, 0.73438)</td>
</tr>
<tr>
<td>((0.7375 + 0.73438)/2 = 0.73594)</td>
<td>−0.017</td>
<td>(0.73438, 0.73594)</td>
</tr>
<tr>
<td>((0.73438 + 0.73594)/2 = 0.73516)</td>
<td>−0.007</td>
<td>(0.73438, 0.73516)</td>
</tr>
<tr>
<td>((0.73438 + 0.73516)/2 = 0.73477)</td>
<td>−0.002</td>
<td>(0.73438, 0.73477)</td>
</tr>
<tr>
<td>((0.73438 + 0.73477)/2 = 0.73458)</td>
<td>0.000</td>
<td>−</td>
</tr>
</tbody>
</table>

The final result \( x = 0.7346 \) is correct within four decimal places.
Questions

Question 1

Approximate the root of \( f(x) = x^3 - 3 \) with the bisection method starting with the interval \([1, 2]\) and use \( \varepsilon_{\text{step}} = 0.1 \) and \( \varepsilon_{\text{abs}} = 0.1 \).

Answer: 1.4375

Question 2

Approximate the root of \( f(x) = x^2 - 10 \) with the bisection method starting with the interval \([3, 4]\) and use \( \varepsilon_{\text{step}} = 0.1 \) and \( \varepsilon_{\text{abs}} = 0.1 \).

Answer: 3.15625 (you need a few extra steps for \( \varepsilon_{\text{abs}} \))
% Root finding by bi-section
f=inline('a*x -1','x','a');

a=2

figure(1)
clf
hold on
ezplot('a*x -1')
x1=0;
x2=1.5;
x=[x1 x2];
eps=1e-3;
err=max(abs(x(1)-x(2)),abs(f(x(1),a)-f(x(2),a)));
while (err>eps & f(x(1),a)*f(x(2),a) <= 0)
xo=x;
x=[xo(1) 0.5*(xo(1)+xo(2))];
if ( f(x(1),a)*f(x(2),a) > 0 )
x=[0 0.5*(xo(1)+xo(2)) xo(2)];
end
x0 = 1.5;
x = [x1 x2];
eps = 1e-3;
err = max(abs(x(1) - x(2)), abs(f(x(1), a) - f(x(2), a)));
while (err > eps & f(x(1), a) * f(x(2), a) <= 0)
    xo = x;
    x = [xo(1) 0.5*(xo(1)+xo(2))];
    if (f(x(1), a) * f(x(2), a) > 0)
        x = [0.5*(xo(1)+xo(2)) xo(2)]
    end
end

err = max(abs(x(1) - x(2)), abs(f(x(1), a) - f(x(2), a)));
pause
b = plot(x, f(x, a), '.b');
set(b, 'MarkerSize', 20);
grid on;
end
تعریف تابع درون دیک‌ام- فایل

```matlab
function y=gholi(x)

% UNTITLED2 Summary of this function goes here
% Detailed explanation goes here

y = x.^3 - 2

end
```
format long
eps_abs = 1e-5;
eps_step = 1e-5;
a = 0.0;
b = 2.0;
while (b - a >= eps_step || ( abs( gholi(a) ) >= eps_abs && abs( gholi(b) ) >= eps_abs ) )
    c = (a + b)/2;
    if ( gholi(c) == 0 )
        break;
    elseif ( gholi(a)*gholi(c) < 0 )
        b = c;
    else
        a = c;
end
end
Introduction

The false-position method is a modification on the bisection method: if it is known that the root lies on \([a, b]\), then it is reasonable that we can approximate the function on the interval by interpolating the points \((a, f(a))\) and \((b, f(b))\). In that case, why not use the root of this linear interpolation as our next approximation to the root?

Figure 1. A function on an interval \([6, 8]\).
The bisection method would have us use 7 as our next approximation, however, it should be quite apparent that we could easily interpolate the points (6, f(6)) and (8, f(8)), as is shown in Figure 2, and use the root of this linear interpolation as our next end point for the interval.

This should, and usually does, give better approximations of the root, especially when the approximation of the function by a linear function is a valid. This method is called the false-position method, also known as the reguli-falsi. Later, we look at a case where the false-position method fails because the function is highly non-linear.
Halting Conditions

The halting conditions for the false-position method are different from the bisection method. If you view the sequence of iterations of the false-position method in Figure 3, you will note that only the left bound is ever updated, and because the function is concave up, the left bound will be the only one which is ever updated. Thus, instead of checking the width of the interval, we check the change in the end points to determine when to stop.

Figure 3. Four iterations of the false-position method on a concave-up function.
The Effect of Non-linear Functions

If we cannot assume that a function may be interpolated by a linear function, then applying the false-position method can result in worse results than the bisection method. For example, Figure 4 shows a function where the false-position method is significantly slower than the bisection method. Such a situation can be recognized and compensated for by falling back on the bisection method for two or three iterations and then resuming with the false-position method.

Figure 4. Twenty iterations of the false-position method on a highly-nonlinear function
**Problem**
Given a function of one variable, \( f(x) \), find a value \( r \) (called a root) such that \( f(r) = 0 \).

**Assumptions**
We will assume that the function \( f(x) \) is continuous.

**Tools**
We will use sampling, bracketing, and iteration.

**Initial Requirements**
We have an initial bound \([a, b]\) on the root, that is, \( f(a) \) and \( f(b) \) have opposite signs. Set the variable \( \text{step} = l \).

**Iteration Process**
Given the interval \([a, b]\), define \( c = (a f(b) - b f(a))/(f(b) / f(a)) \). Then if \( f(c) = 0 \) (unlikely in practice), then halt, as we have found a root, if \( f(c) \) and \( f(a) \) have opposite signs, then a root must lie on \([a, c]\), so assign

- \( \text{step} = b - c \) and assign \( b = c \),
else if \( f(c) \) and \( f(b) \) must have opposite signs, and thus a root must lie on \([c, b]\), so assign \( \text{step} = c - a \) and assign \( a = c \).
Linear interpolation works by drawing a straight line between two points and returning the appropriate point along the straight line. We can write the point $c$ as

$$c = a - \frac{b - a}{f(b) - f(a)} f(a) = b - \frac{b - a}{f(b) - f(a)} f(b) = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

1. Let $c = a - \frac{b - a}{f(b) - f(a)} f(a) = b - \frac{b - a}{f(b) - f(a)} f(b) = \frac{af(b) - bf(a)}{f(b) - f(a)}$

2. If $f(c) = 0$ then $c$ is an exact solution.

3. If $f(a) < 0$ and $f(c) > 0$ then the root lies in $[a, c]$.

4. Else the root lies in $[c, b]$.
Figure 1.4: Linear Interpolation: This method works by drawing a line from one end of the interval to the other and calculating the intercept, $c$, with the axis. Depending on the value of $f(c)$ we can redefine our interval.
Example 1

Consider finding the root of \( f(x) = x^2 - 3 \). Let \( \varepsilon_{\text{step}} = 0.01 \), \( \varepsilon_{\text{abs}} = 0.01 \) and start with the interval \([1, 2]\).

Table 1. False-position method applied to \( f(x) = x^2 - 3 \).

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>f(a)</td>
<td>f(b)</td>
<td>c</td>
<td>f(c)</td>
<td>Update</td>
<td>Step Size</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>------</td>
<td>------</td>
<td>---</td>
<td>------</td>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td>1.0</td>
<td>2.0</td>
<td>-2.00</td>
<td>1.00</td>
<td>1.6667</td>
<td>-0.2221</td>
<td>a = c</td>
<td>0.6667</td>
</tr>
<tr>
<td>1.6667</td>
<td>2.0</td>
<td>-0.2221</td>
<td>1.0</td>
<td>1.7273</td>
<td>-0.0164</td>
<td>a = c</td>
<td>0.0606</td>
</tr>
<tr>
<td>1.7273</td>
<td>2.0</td>
<td>-0.0164</td>
<td>1.0</td>
<td>1.7317</td>
<td>0.0012</td>
<td>a = c</td>
<td>0.0044</td>
</tr>
</tbody>
</table>

Thus, with the third iteration, we note that the last step \( 1.7273 \to 1.7317 \) is less than 0.01 and \( |f(1.7317)| < 0.01 \), and therefore we chose \( b = 1.7317 \) to be our approximation of the root.

Note that after three iterations of the false-position method, we have an acceptable answer (1.7317 where \( f(1.7317) = -0.0044 \)) whereas with the bisection method, it took seven iterations to find a (notable less accurate) acceptable answer (1.71344 where \( f(1.73144) = 0.0082 \))
Example 2

Consider finding the root of \( f(x) = e^{-x}(3.2 \sin(x) - 0.5 \cos(x)) \) on the interval \([3, 4]\), this time with \( \varepsilon_{\text{step}} = 0.001, \varepsilon_{\text{abs}} = 0.001 \).

Table 2. False-position method applied to \( f(x) = e^{-x}(3.2 \sin(x) - 0.5 \cos(x)) \).

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>f(a)</th>
<th>f(b)</th>
<th>c</th>
<th>f(c)</th>
<th>Update</th>
<th>Step Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>4.0</td>
<td>0.047127</td>
<td>-0.038372</td>
<td>3.5513</td>
<td>-0.023411</td>
<td>b = c</td>
<td>0.4487</td>
</tr>
<tr>
<td>3.0</td>
<td>3.5513</td>
<td>0.047127</td>
<td>-0.023411</td>
<td>3.3683</td>
<td>-0.0079940</td>
<td>b = c</td>
<td>0.1830</td>
</tr>
<tr>
<td>3.0</td>
<td>3.3683</td>
<td>0.047127</td>
<td>-0.0079940</td>
<td>3.3149</td>
<td>-0.0021548</td>
<td>b = c</td>
<td>0.0534</td>
</tr>
<tr>
<td>3.0</td>
<td>3.3149</td>
<td>0.047127</td>
<td>-0.0021548</td>
<td>3.3010</td>
<td>-0.00052616</td>
<td>b = c</td>
<td>0.0139</td>
</tr>
<tr>
<td>3.0</td>
<td>3.3010</td>
<td>0.047127</td>
<td>-0.00052616</td>
<td>3.2978</td>
<td>-0.00014453</td>
<td>b = c</td>
<td>0.0032</td>
</tr>
<tr>
<td>3.0</td>
<td>3.2978</td>
<td>0.047127</td>
<td>-0.00014453</td>
<td>3.2969</td>
<td>-0.000036998</td>
<td>b = c</td>
<td>0.0009</td>
</tr>
</tbody>
</table>

Thus, after the sixth iteration, we note that the final step, 3.2978 → 3.2969 has a size less than 0.001 and \(|f(3.2969)| < 0.001\) and therefore we chose \( b = 3.2969 \) to be our approximation of the root.

In this case, the solution we found was not as good as the solution we found using the bisection method (\( f(3.2963) = 0.000034799 \)) however, we only used six instead of eleven iterations.
Question 1

Approximate the root of $f(x) = x^3 - 3$ with the false-position method starting with the interval $[1, 2]$ and use $\varepsilon_{\text{step}} = 0.1$ and $\varepsilon_{\text{abs}} = 0.1$. Use five decimal digits of accuracy.

Answer: 1.4267

Question 2

Approximate the root of $f(x) = x^2 - 10$ with the false-position method starting with the interval $[3, 4]$ and use $\varepsilon_{\text{step}} = 0.1$ and $\varepsilon_{\text{abs}} = 0.1$.

Answer: 3.16
function y = ff(x)

% UNTITLED2 Summary of this function goes here

% Detailed explanation goes here

y = x.^3 - 2

end
format long
eps_abs = 1e-5;
eps_step = 1e-5;
a = 0.0;
b = 2.0;
step_size = Inf;
while (step_size >= eps_step || (abs(ff(a)) >= eps_abs && abs(ff(b)) >= eps_abs))
    c = (ff(a)*b - ff(b)*a)/(ff(a) - ff(b));
    if (ff(c) == 0)
        break;
    elseif (ff(a)*ff(c) < 0)
        step_size = b - c;
        b = c;
    else
        step_size = c - a;
        a = c;
end
end