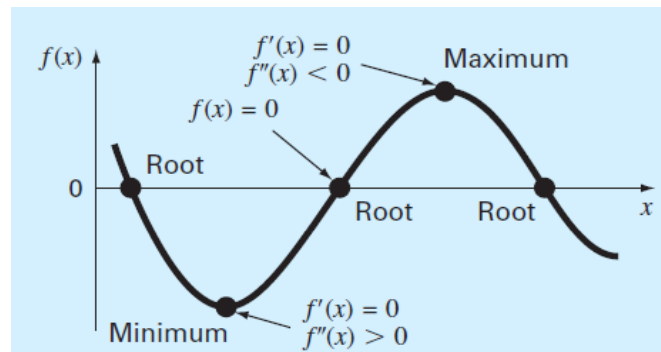


Lecture 05: Roots and Optimization: Bracketing Methods



Roots of Equations



$$f(x) = ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- **Roots** are sometimes called the **zeros** of the equation.
- **Methods:**
 - Direct methods
 - Approximate solution techniques

Roots of Equations



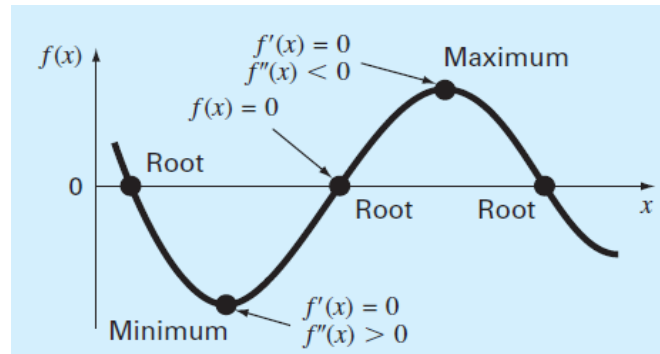
- An approximate solution technique → **graphical methods**.
- Plot the function and determine where it crosses the x axis.
- Although graphical methods are useful for obtaining rough estimates of roots, they are limited because of their lack of precision.
- An alternative approach is to use **trial and error**.
- This technique consists of guessing a value of x and evaluating whether $f(x)$ is zero.
- If not (as is almost always the case), another guess is made, and $f(x)$ is again evaluated to determine whether the new value provides a better estimate of the root.

Optimization



- Besides roots, another feature of interest to engineers and scientists are a function's **minimum** and **maximum** values → referred to as **optimization**.
- From a numerical standpoint, such optimization methods are similar in spirit to the root-location methods.
- Both involve guessing and searching for a location on a function.
- Root location involves searching for the location where the function equals zero.
- In contrast, optimization involves searching for the function's extreme points.

Difference between Roots and Optima



Roots: Bracketing Methods



- Example**

Medical studies have established that a bungee jumper's chances of sustaining a significant vertebrae injury increase significantly if the free-fall velocity exceeds 36 m/s after 4 s of free fall. Your boss at the bungee-jumping company wants you to determine the mass at which this criterion is exceeded given a drag coefficient of 0.25 kg/m.

You know from your previous studies that the following analytical solution can be used to predict fall velocity as a function of time:

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right)$$

Try as you might, you cannot manipulate this equation to explicitly solve for $m \rightarrow$ you cannot isolate the mass on the left side of the equation.

Roots: Bracketing Methods



- Example
 - Solution

$$f(m) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) - v(t)$$

- The value of m that makes $f(m) = 0$ is, therefore, the root of the equation.

Roots in Engineering and Science

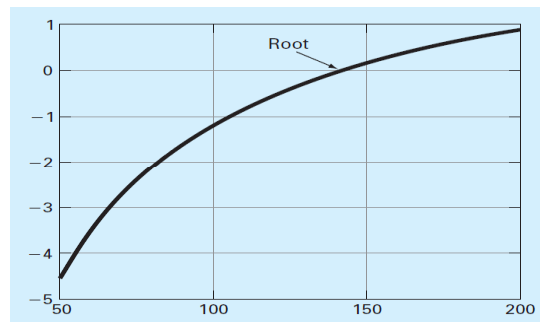


Fundamental Principle	Dependent Variable	Independent Variable	Parameters
Heat balance	Temperature	Time and position	Thermal properties of material, system geometry
Mass balance	Concentration or quantity of mass	Time and position	Chemical behavior of material, mass transfer, system geometry
Force balance	Magnitude and direction of forces	Time and position	Strength of material, structural properties, system geometry
Energy balance	Changes in kinetic and potential energy	Time and position	Thermal properties, mass of material, system geometry
Newton's laws of motion	Acceleration, velocity, or location	Time and position	Mass of material, system geometry, dissipative parameters
Kirchhoff's laws	Currents and voltages	Time	Electrical properties (resistance, capacitance, inductance)

Example 1: Graphical Methods



Problem Statement. Use the graphical approach to determine the mass of the bungee jumper with a drag coefficient of 0.25 kg/m to have a velocity of 36 m/s after 4 s of free fall. Note: The acceleration of gravity is 9.81 m/s^2 .

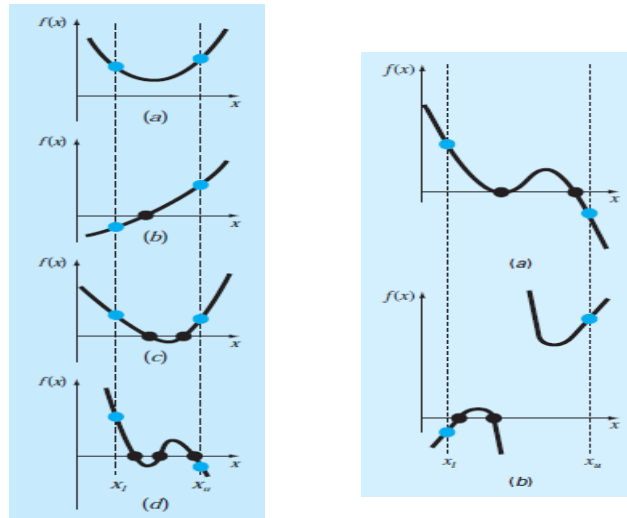


Graphical Methods



- Graphical techniques are of limited practical value because they are not very precise.
- However, graphical methods can be utilized to obtain rough estimates of roots.
- These estimates can be employed as **starting guesses** for numerical methods discussed later in this lecture.
- Also, graphical interpretations are useful for understanding the properties of the functions and anticipating the pitfalls of the numerical methods.

Graphical Methods



Graphical Methods



- In general, if $f(x_j)$ and $f(x_u)$ have opposite signs, there are an **odd number of roots** in the interval.
- If $f(x_j)$ and $f(x_u)$ have the same sign, there are either **no roots** or an **even number of roots** between the values.
- Although these generalizations are usually true, there are cases where they do not hold.
- For example, functions that are tangential to the x axis and discontinuous functions can violate these principles.
- An example of a function that is tangential to the axis is the cubic equation $f(x) = (x - 2)(x - 2)(x - 4)$.
- Notice that $x = 2$ makes two terms in this polynomial equal to zero. Mathematically, $x = 2$ is called a **multiple root**.

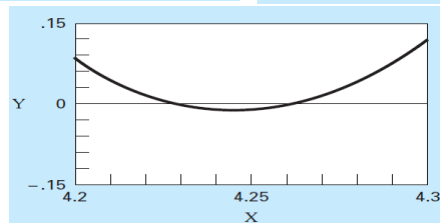
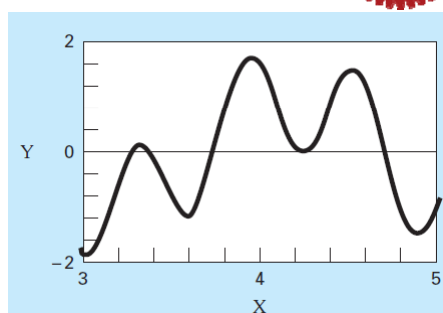
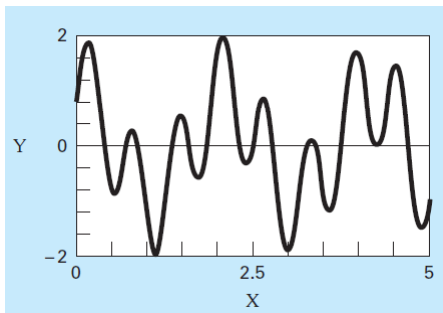
Example 2: Graphical Methods

Problem Statement. Computer graphics can expedite and improve your efforts to locate roots of equations. The function

$$f(x) = \sin 10x + \cos 3x$$

has several roots over the range $x = 0$ to $x = 5$. Use computer graphics to gain insight into the behavior of this function.

Example 2: Graphical Methods



Numerical Methods



- Two major classes of methods available:
 - **Bracketing methods** → As the name implies, these are based on two initial guesses that bracket the root that are on either side of the root.
 - **Open methods** → These methods can involve one or more initial guesses, but there is no need for them to bracket the root.
- For well-posed problems, the bracketing methods always work but converge slowly.
- In contrast, the open methods do not always work (i.e., they can diverge), but when they do they usually converge quicker.

Bracketing Methods



- These methods exploit the fact that a function typically changes sign in the vicinity of a root.
- These techniques are called bracketing methods because two initial guesses for the root are required.
- As the name implies, these guesses must “bracket,” or be on either side of, the root.

Incremental Search



- In general, if $f(x)$ is real and continuous in the interval from x_l to x_u and $f(x_l)$ and $f(x_u)$ have opposite signs, i.e., $f(x_l) f(x_u) < 0$, then there is at least one real root between x_l and x_u .
- **Incremental search methods** capitalize on this observation by locating an interval where the function changes sign.

Incremental Search



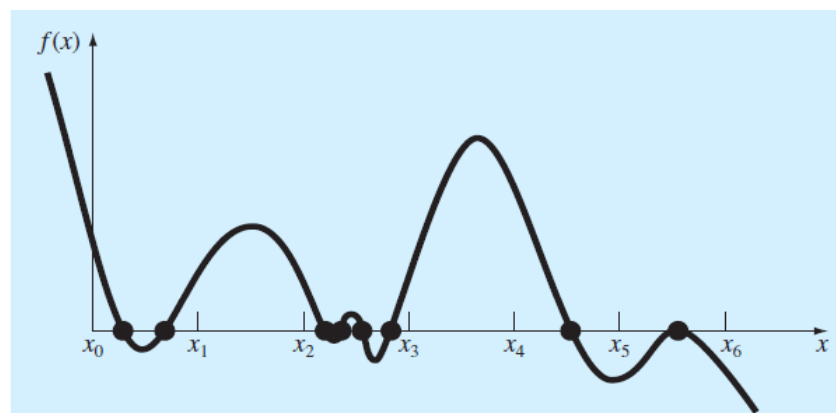
- Then the location of the sign change (and consequently, the root) is identified more precisely by dividing the interval into a number of subintervals.
- Each of these subintervals is searched to locate the sign change.
- The process is repeated and the root estimate refined by dividing the subintervals into finer increments.

Incremental Search



- A potential problem with an incremental search is the choice of the increment length.
- If the length is too small, the search can be very time consuming.
- On the other hand, if the length is too great, there is a possibility that closely spaced roots might be missed .
- The problem is compounded by the possible existence of multiple roots.

Incremental Search

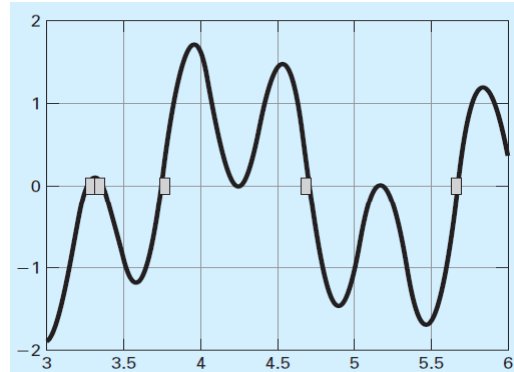


Example 3 : Incremental Search

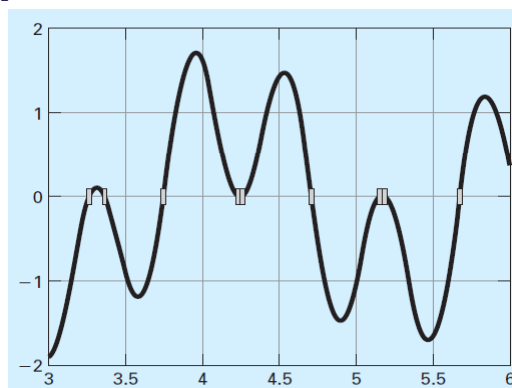


Problem Statement. Use the M-file `incsearch` (Fig. 5.4) to identify brackets within the interval $[3, 6]$ for the function:

$$f(x) = \sin(10x) + \cos(3x) \quad (5.4)$$



Example 3 : Incremental Search



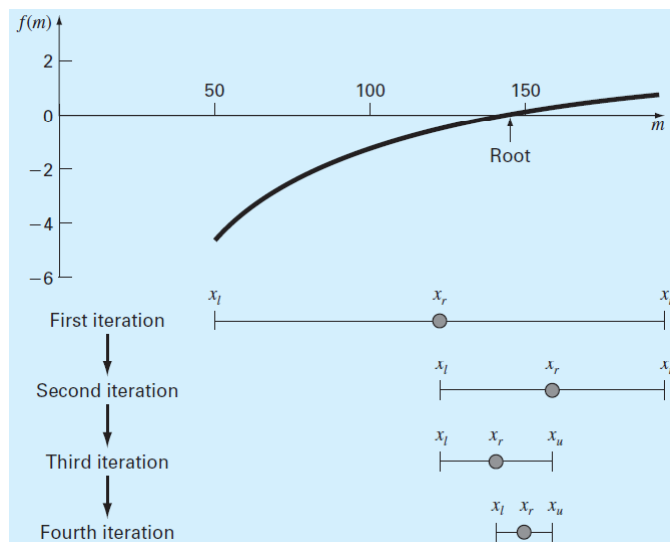
- The foregoing example illustrates that **brute-force methods** such as incremental search are not foolproof.
- You would be wise to supplement such automatic techniques with any other information that provides insight into the location of the roots.

Bisection Method



- The bisection method is also known as binary chopping, interval halving, or Bolzano's method.
- It is a variation of the incremental search method in which **the interval is always divided in half**.
- If a function changes sign over an interval, the function value at the **midpoint** is evaluated.
- The location of the root is then determined as lying within the subinterval where the sign change occurs.
- The subinterval then becomes the interval for the next iteration.
- The process is repeated until the root is known to the required precision.

Bisection Method



Bisection Method



Step 1: Choose lower x_l and upper x_u guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that $f(x_l)f(x_u) < 0$.

Step 2: An estimate of the root x_r is determined by

$$x_r = \frac{x_l + x_u}{2}$$

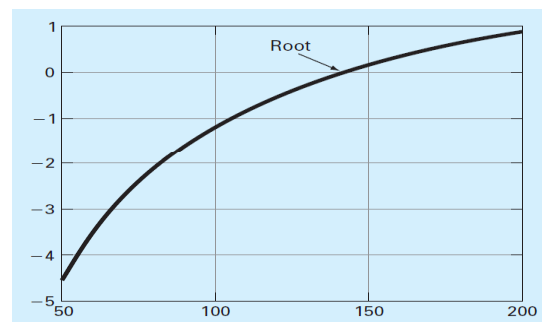
Step 3: Make the following evaluations to determine in which subinterval the root lies:

- (a) If $f(x_l)f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.
- (b) If $f(x_l)f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.
- (c) If $f(x_l)f(x_r) = 0$, the root equals x_r ; terminate the computation.

Example 4: Bisection Method



Problem Statement. Use the *bisection method* to determine the mass of the bungee jumper with a drag coefficient of 0.25 kg/m to have a velocity of 36 m/s after 4 s of free fall. Note: The acceleration of gravity is 9.81 m/s².



Example 4: Bisection Method



$$x_r = \frac{50 + 200}{2} = 125 \quad \text{exact value of the root is } 142.7376.$$

$$|\varepsilon_t| = \left| \frac{142.7376 - 125}{142.7376} \right| \times 100\% = 12.43\%$$

$$f(50)f(125) = -4.579(-0.409) = 1.871$$

$$x_l = 125 \text{ to } x_u = 200. \quad x_r = \frac{125 + 200}{2} = 162.5$$

$$|\varepsilon_t| = 13.85\%$$

$$f(125)f(162.5) = -0.409(0.359) = -0.147 \quad x_r = \frac{125 + 162.5}{2} = 143.75$$

$$\varepsilon_t = 0.709\%$$

Termination Criteria and Error Estimates



$$\varepsilon_a = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}} \right| 100\%$$

- When ε_a becomes less than a prespecified stopping criterion ε_s , the computation is terminated.

Example 5: Bisection Method



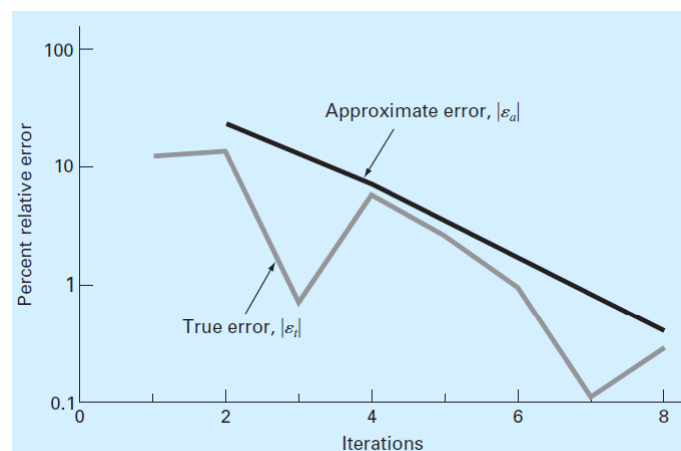
Problem Statement. Continue Example 5.3 until the approximate error falls below a stopping criterion of $\epsilon_s = 0.5\%$. Use Eq. (5.2) to compute the errors.

Solution. The results of the first two iterations for Example 5.3 were 125 and 162.5. Substituting these values into Eq. (5.5) yields

$$|\epsilon_a| = \left| \frac{162.5 - 125}{162.5} \right| 100\% = 23.08\%$$

Iteration	x_l	x_u	x_r	$ \epsilon_a $ (%)	$ \epsilon_t $ (%)
1	50	200	125		12.43
2	125	200	162.5	23.08	13.85
3	125	162.5	143.75	13.04	0.71
4	125	143.75	134.375	6.98	5.86
5	134.375	143.75	139.0625	3.37	2.58
6	139.0625	143.75	141.4063	1.66	0.93
7	141.4063	143.75	142.5781	0.82	0.11
8	142.5781	143.75	143.1641	0.41	0.30

Example 5: Bisection Method

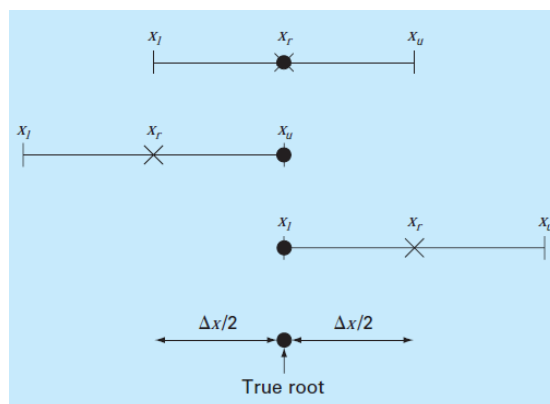


Termination Criteria and Error Estimates

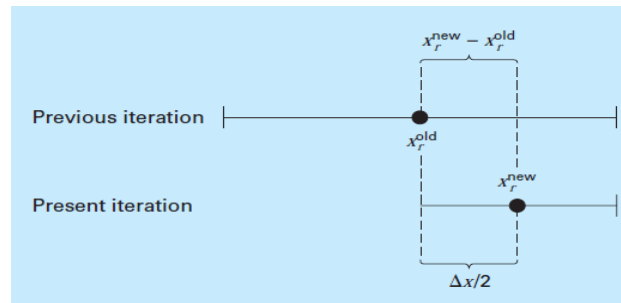


- It can be demonstrated that $|\epsilon_a|$ will always be greater than $|\epsilon_t|$ for bisection.
- This is due to the fact that each time an approximate root is located using bisection as $x_r = (x_l + x_u)/2$, we know that the true root lies somewhere within an interval of $\Delta x = x_u - x_l$.
- Therefore, the root must lie within $\pm\Delta x/2$ of our estimate.

Termination Criteria and Error Estimates



Termination Criteria and Error Estimates



$$\varepsilon_a = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}} \right| 100\%$$

Provides an exact upper bound on the true error

Termination Criteria and Error Estimates



- It is also possible to calculate an error estimate on the basis of our initial guesses—that is, on our first iteration.

$$x_r^{\text{new}} - x_r^{\text{old}} = \frac{x_u - x_l}{2}$$

$$x_r^{\text{new}} = \frac{x_l + x_u}{2}$$

$$\varepsilon_a = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}} \right| 100\%$$

$$\varepsilon_a = \left| \frac{x_u - x_l}{x_u + x_l} \right| 100\%$$

Termination Criteria and Error Estimates



- Another benefit of the bisection method is that the **number of iterations required to attain an absolute error** can be computed *a priori*—i.e., before starting the computation.

$$E_a^0 = x_u^0 - x_l^0 = \Delta x^0 \quad E_a^1 = \frac{\Delta x^0}{2} \quad E_a^n = \frac{\Delta x^0}{2^n}$$

$$n = \frac{\log(\Delta x^0 / E_{a,d})}{\log 2} = \log_2 \left(\frac{\Delta x^0}{E_{a,d}} \right) \quad E_{a,d} \text{ is the desired error}$$

$$\log_b(x) = \log(x) / \log(b).$$

Example 5: Bisection Method (Cont'd)



Iteration	x_l	x_u	x_r	$ \varepsilon_a $ (%)	$ \varepsilon_r $ (%)
1	50	200	125		12.43
2	125	200	162.5	23.08	13.85
3	125	162.5	143.75	13.04	0.71
4	125	143.75	134.375	6.98	5.86
5	134.375	143.75	139.0625	3.37	2.58
6	139.0625	143.75	141.4063	1.66	0.93
7	141.4063	143.75	142.5781	0.82	0.11
8	142.5781	143.75	143.1641	0.41	0.30

$$\Delta x_0 = 200 - 50 = 150$$

$$E_a = \frac{|143.7500 - 142.5781|}{2} = 0.5859$$

$$n = \log_2(150/0.5859) = 8$$

Bisection Algorithm



```

FUNCTION Bisect(xl, xu, es, imax, xr, iter, ea)
  iter = 0
  DO
    xrold = xr
    xr = (xl + xu) / 2
    iter = iter + 1
    IF xr ≠ 0 THEN
      ea = ABS((xr - xrold) / xr) * 100
    END IF
    test = f(xl) * f(xr)
    IF test < 0 THEN
      xu = xr
    ELSE IF test > 0 THEN
      xl = xr
    ELSE
      ea = 0
    END IF
    IF ea < es OR iter ≥ imax EXIT
  END DO
  Bisect = xr
END Bisect

```

Minimizing Function Evaluations



```

FUNCTION Bisect(xl, xu, es, imax, xr, iter, ea)
  iter = 0
  DO
    xrold = xr
    xr = (xl + xu) / 2
    iter = iter + 1
    IF xr ≠ 0 THEN
      ea = ABS((xr - xrold) / xr) * 100
    END IF
    test = f(xl) * f(xr)
    IF test < 0 THEN
      xu = xr
    ELSE IF test > 0 THEN
      xl = xr
    ELSE
      ea = 0
    END IF
    IF ea < es OR iter ≥ imax EXIT
  END DO
  Bisect = xr
END Bisect

```

```

FUNCTION Bisect(xl, xu, es, imax, xr, iter, ea)
  iter = 0
  f1 = f(xl)
  DO
    xrold = xr
    xr = (xl + xu) / 2
    fr = f(xr)
    iter = iter + 1
    IF xr ≠ 0 THEN
      ea = ABS((xr - xrold) / xr) * 100
    END IF
    test = f1 * fr
    IF test < 0 THEN
      xu = xr
    ELSE IF test > 0 THEN
      xl = xr
      f1 = fr
    ELSE
      ea = 0
    END IF
    IF ea < es OR iter ≥ imax EXIT
  END DO
  Bisect = xr
END Bisect

```

**(n+1) function
evaluations are
performed, rather
than 2n**

False Position Method

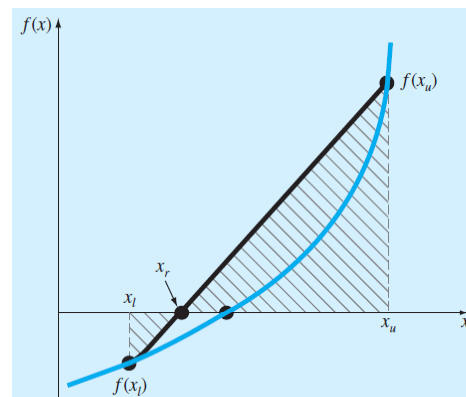


- False position is also as the linear interpolation method.
- It is very similar to bisection with the exception that it uses a different strategy to come up with its new root estimate.
- False position is an alternative based on a graphical insight.
- A shortcoming of the bisection method is that, in dividing the interval from x_l to x_u into equal halves, no account is taken of the magnitudes of $f(x_l)$ and $f(x_u)$.

False Position Method



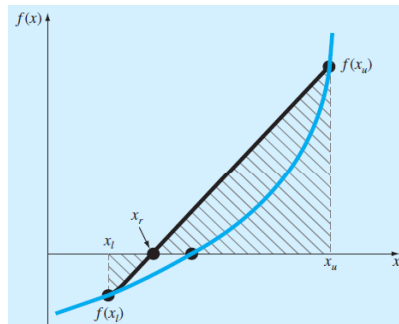
- Rather than bisecting the interval, it locates the root by joining $f(x_l)$ and $f(x_u)$ with a straight line.
- The intersection of this line with the x axis represents an improved estimate of the root.
- Thus, the shape of the function influences the new root estimate.



False Position Method



- The fact that the replacement of the curve by a straight line gives a “false position” of the root is the origin of the name, method of **false position**, or in Latin, *regula falsi*.



$$\frac{f(x_l)}{x_r - x_l} = \frac{f(x_u)}{x_r - x_u}$$

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

Derivation of the Method of False Position



$$\frac{f(x_l)}{x_r - x_l} = \frac{f(x_u)}{x_r - x_u}$$

$$x_r = \frac{x_u f(x_l)}{f(x_l) - f(x_u)} - \frac{x_l f(x_u)}{f(x_l) - f(x_u)}$$

$$f(x_l)(x_r - x_u) = f(x_u)(x_r - x_l)$$

$$x_r = x_u + \frac{x_u f(x_l)}{f(x_l) - f(x_u)} - x_u - \frac{x_l f(x_u)}{f(x_l) - f(x_u)}$$

$$x_r [f(x_l) - f(x_u)] = x_u f(x_l) - x_l f(x_u)$$

$$x_r = x_u + \frac{x_u f(x_l)}{f(x_l) - f(x_u)} - \frac{x_l f(x_u)}{f(x_l) - f(x_u)}$$

$$x_r = \frac{x_u f(x_l) - x_l f(x_u)}{f(x_l) - f(x_u)}$$

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

- The later involves one less function evaluation and one less multiplication than the former.

Example 6: False-Position Method

Problem Statement. Use the false-position method to determine the drag coefficient c needed for a parachutist of mass $m = 68.1$ kg to have a velocity of 40 m/s after free-falling for time $t = 10$ s. *Note:* The acceleration due to gravity is 9.8 m/s².

Solution. As in Example 5.3, initiate the computation with guesses of $x_l = 12$ and $x_u = 16$.

First iteration:

$$\begin{aligned} x_l &= 12 & f(x_l) &= 6.0699 \\ x_u &= 16 & f(x_u) &= -2.2688 \\ x_r &= 16 - \frac{2.2688(12 - 16)}{6.0669 - (-2.2688)} = 14.9113 \end{aligned}$$

which has a true relative error of 0.89 percent.

Second iteration:

$$f(x_l) f(x_r) = -1.5426$$

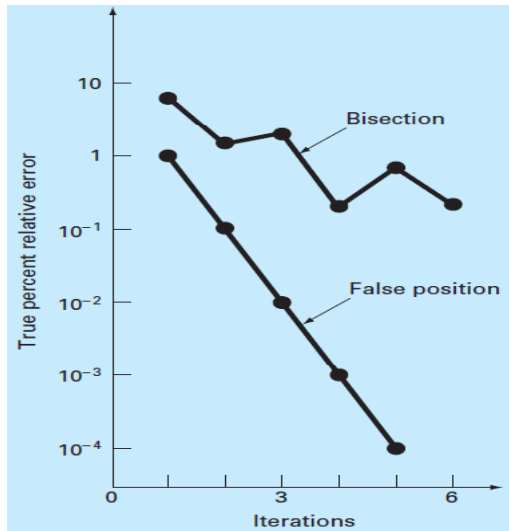
Example 6: False-Position Method

Therefore, the root lies in the first subinterval, and x_r becomes the upper limit for the next iteration, $x_u = 14.9113$:

$$\begin{aligned} x_l &= 12 & f(x_l) &= 6.0699 \\ x_u &= 14.9113 & f(x_u) &= -0.2543 \\ x_r &= 14.9113 - \frac{-0.2543(12 - 14.9113)}{6.0669 - (-0.2543)} = 14.7942 \end{aligned}$$

which has true and approximate relative errors of 0.09 and 0.79 percent. Additional iterations can be performed to refine the estimate of the roots.

Comparison of Relative Errors



- The error for false position **decreases much faster** than for bisection because of the more efficient scheme for root location in the false-position method.

Pitfalls of False-Position Method



- A major weakness of the false-position method: its one-sidedness.
- That is, as iterations are proceeding, one of the bracketing points will tend to stay fixed.
- This can lead to poor convergence, particularly for functions with significant curvature.

Example 7: A Case where Bisection is Preferable to False Position

Problem Statement. Use bisection and false position to locate the root of

$$f(x) = x^{10} - 1$$

between $x = 0$ and 1.3 .

Solution. Using bisection, the results can be summarized as

Iteration	x_l	x_u	x_r	ϵ_a (%)	ϵ_t (%)
1	0	1.3	0.65	100.0	35
2	0.65	1.3	0.975	33.3	2.5
3	0.975	1.3	1.1375	14.3	13.8
4	0.975	1.1375	1.05625	7.7	5.6
5	0.975	1.05625	1.015625	4.0	1.6

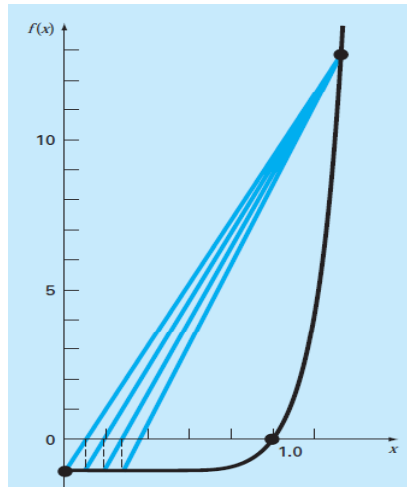
Example 7: A Case where Bisection is Preferable to False Position

Thus, after five iterations, the true error is reduced to less than 2 percent. For false position, a very different outcome is obtained:

Iteration	x_l	x_u	x_r	ϵ_a (%)	ϵ_t (%)
1	0	1.3	0.09430		90.6
2	0.09430	1.3	0.18176	48.1	81.8
3	0.18176	1.3	0.26287	30.9	73.7
4	0.26287	1.3	0.33811	22.3	66.2
5	0.33811	1.3	0.40788	17.1	59.2

- After five iterations, the true error has only been reduced to about 59 percent.
- In addition, note that $\epsilon_a < \epsilon_t \rightarrow$ the approximate error is misleading.

Example 7: A Case where Bisection is Preferable to False Position



- The curve violates the **premise** upon which false position was based—that is, if $f(x_l)$ is much closer to zero than $f(x_u)$, then the root is closer to x_l than to x_u .
- Because of the shape of the present function, the opposite is true.

Modified False Position

- One way to mitigate the “one-sided” nature of false position is to have the algorithm detect when one of the bounds is stuck.
- If this occurs, the function value at the stagnant bound can be divided in half.
- This is called the *modified false-position method*.

Modified False Position



```
FUNCTION ModFalsePos(xl, xu, es, imax, xr, iter, ea)
```

```
  iter = 0
  fl = f(xl)
  fu = f(xu)
  DO
    xrold = xr
    xr = xu - fu * (xl - xu) / (fl - fu)
    fr = f(xr)
    iter = iter + 1
    IF xr <> 0 THEN
      ea = Abs((xr - xrold) / xr) * 100
    END IF
    test = fl * fr
    IF test < 0 THEN
      xu = xr
      fu = f(xu)
      iu = 0
      i1 = i1 + 1
      If i1 ≥ 2 THEN fl = fl / 2
```

```
    ELSE IF test > 0 THEN
      xl = xr
      fl = f(xl)
      i1 = 0
      iu = iu + 1
      IF iu ≥ 2 THEN fu = fu / 2
    ELSE
      ea = 0
    END IF
    IF ea < es OR iter ≥ imax THEN EXIT
  END DO
  ModFalsePos = xr
END ModFalsePos
```

Assignment-05



- Problems 5.1, 5.4, 5.9, 5.15, 5.17.