Roots of Equations

Solution of Algebraic and Transcendental equations

Years ago, we learned to use the quadratic formula

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

to solve

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

The values calculated with Eq. are called the "roots" of Eq.

They represent the values of x that make Eq. equal to zero.

An example: A model is derived from Newton's second law, for the parachutist's velocity:

$$v = \frac{gm}{c}\left(1 - e^{-\left(\frac{c}{m}\right)t}\right)$$

where velocity v = the dependent variable, time t = the independent variable, the gravitational constant g = the forcing function, and the drag coefficient c and mass m = parameters.

If the parameters are known, Eq. can be used to predict the parachutist's velocity as a function of time. Such computations can be performed directly because v is expressed *explicitly* as a function of time.

Suppose we want to determine the drag coefficient for a parachutist of a given mass to attain a prescribed velocity in a set time period.

Equation provides a mathematical representation of the interrelationship among the model variables and parameters, it cannot be solved explicitly for the drag coefficient. Try it.

There is no way to rearrange the equation so that c is isolated on one side of the equal sign. In such cases, c is said to be implicit.

Bracketing Methods

This 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.

This method systematically reduce the width of the bracket and, reach to the correct answer.

Graphical Methods

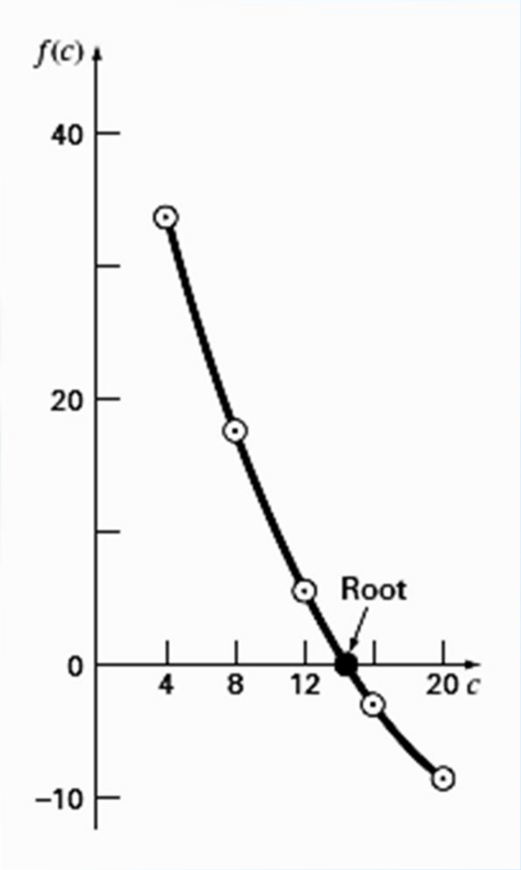
A simple method for obtaining an estimate of the root of the equation f(x) = 0 is to make a plot of the function and observe where it crosses the x axis. This point, which represents the x value for which f(x) = 0, provides a rough approximation of the root.

Example:

Use the graphical approach 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 9.81 m/s^2.

c	f(c)
4	34.190
8	17.712
12	6.114
16	-2.230
20	-8.368

The resulting curve crosses the c axis between 12 and 16. Visual inspection of the plot provides a rough estimate of the root of 14.75.



Bisection Method

When applying the graphical technique It is observed from Figure that f(x) changed sign on opposite sides of the root.

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, that is,

$$f(x_l) \cdot f(x_u) < 0$$

Then there is at least one real root between x_l and x_u.

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 $f(x_l) \cdot f(x_u) < 0$.

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

$$x_r = (x_l + x_u)/2$$

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

(a) If $f(x_l) \cdot f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.

(b) If $f(x_l) \cdot f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.

(c) If $f(x_l) \cdot f(x_r) = 0$, the root equals x_r; terminate the computation.

Termination Criteria and Error Estimates

An initial suggestion might be to end the calculation when the true error falls below some pre specified level.

$$e_t = \left|\frac{x_t - x_r^{new}}{x_t}\right| 100\%$$

However, we don't know the root of the equation therefore, we require an error estimate that is not depending on foreknowledge of the root.

An approximate percent relative error e_a can be calculated, as in

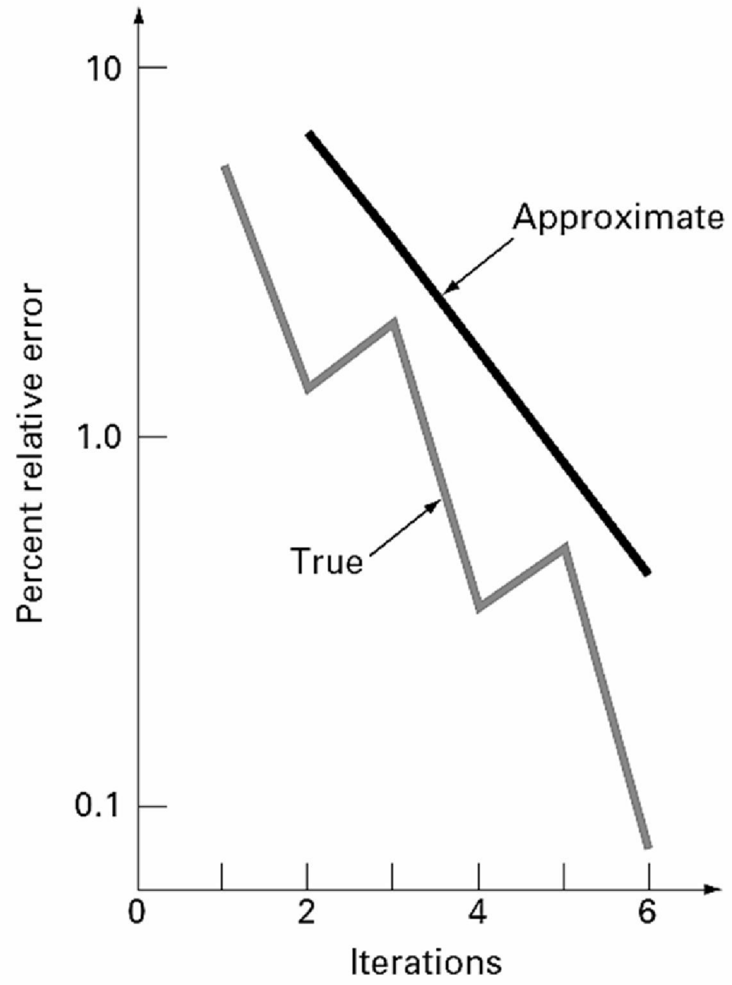
$$e_a = \left|\frac{x_r^{new} - x_r^{old}}{x_r^{new}}\right| 100\%$$

where x_r^{new} is the root for the present iteration and x_r^{old} is the root from the previous iteration.

Although the approximate error does not provide an exact estimate of the true error.

Fig. suggests that e_a captures the general downward trend of e_t.

In addition, the plot exhibits the extremely attractive characteristic that e_a is always greater than e_t.



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 iterations.

the absolute error before starting the technique.

$$E_a^0 = x_u^0 - x_l^0 = \Delta x^0$$

where the superscript designates the iteration. Hence, before starting the method, we are at the "zero iteration." After the first iteration, the error becomes

$$E_a^1 = \frac{\Delta x^0}{2}$$

Because each succeeding iteration halves the error, a general formula relating the error and the number of iterations n is

$$E_a^n = \frac{\Delta x^0}{2^n}$$

If $E_{a,d}$ is the desired error, this equation can be solved for

$$n = \frac{\log(\Delta x^0 / E_{a,d})}{\log 2} = \log_2\left(\frac{\Delta x^0}{E_{a,d}}\right)$$

False-position Method or in Latin, ***regula falsi***.

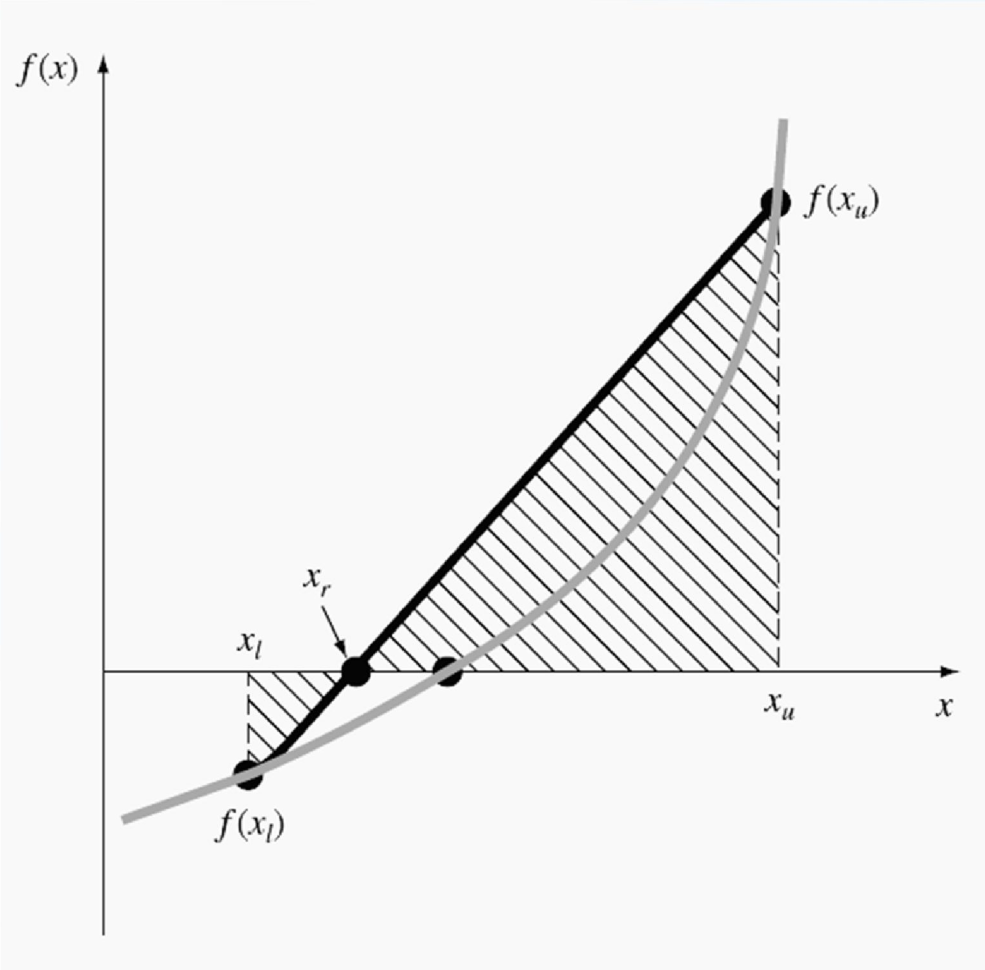
It is also called the *linear interpolation method*.

Bisection is a perfectly valid technique for determining roots, its "brute-force" approach is relatively inefficient.

As well as no account is taken of the magnitudes of $f(x_l)$ and $f(x_u)$.

False position is an alternative based on a graphical insight.

In this method first find the value of $f(x_l)$ and $f(x_u)$ then represent these value in graph. Connect the point with straight line.



The intersection of the straight line with the x axis can be estimated as

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

which can be solved for

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

The value of x_r computed from Eq. replaces one of the initial guess form the two initial guesses, x_l or x_u, which yields a function value with the same sign as $f(x_r)$.
In this way, the values of x_l and x_u always bracket the true root. The process is repeated until the root is estimated adequately.

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 $f(x_l) \cdot f(x_u) < 0$.

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

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

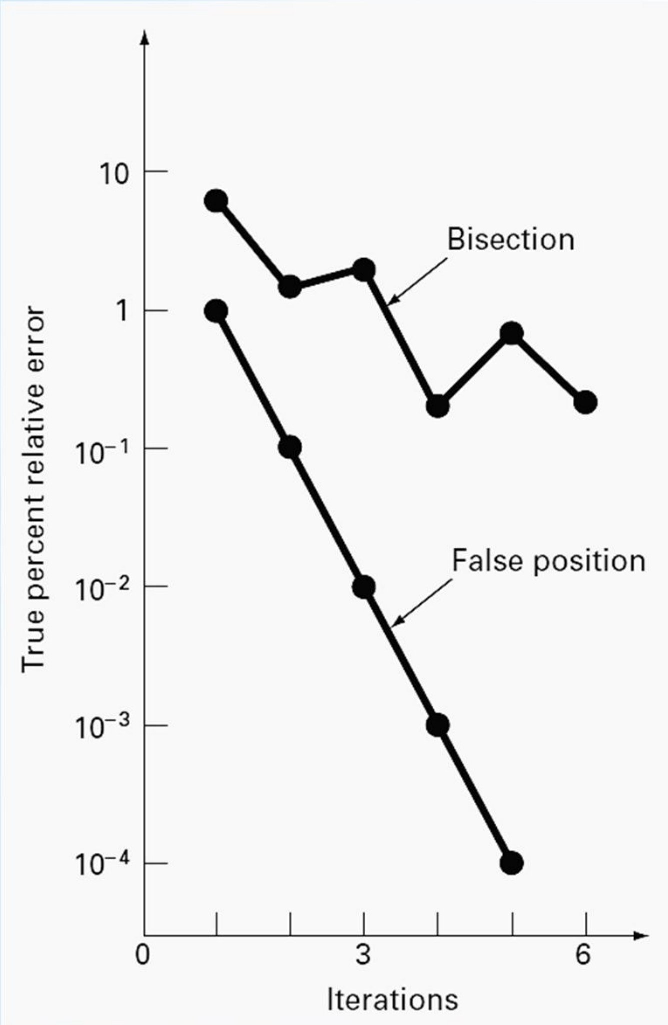
Step 3: Make the following evaluations to determine in which subinterval the root lies:

(a) If $f(x_l) \cdot f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.

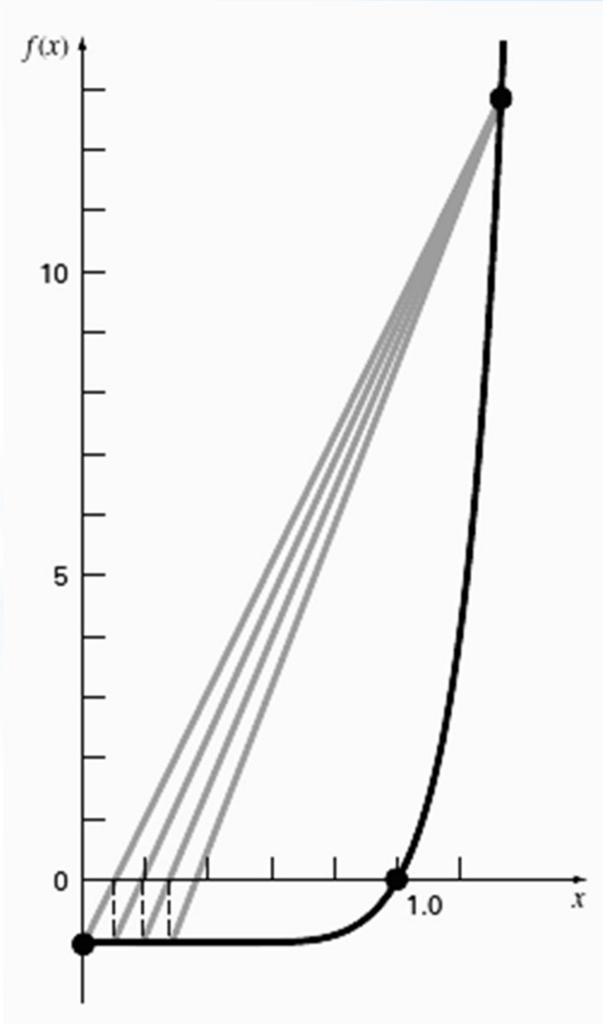
(b) If $f(x_l) \cdot f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.

(c) If $f(x_l) \cdot f(x_r) = 0$, the root equals x_r; terminate the computation.

Comparison of the relative errors of the bisection and the false-position methods.



Plot of $f(x) = x^{10} - 1$, illustrating slow convergence of the false-position method.



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.*

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.

A partial remedy for such cases is:

Compute the first derivative of the function $f'(x)$ at the beginning and the end of each interval. If the derivative changes sign, it suggests that a minimum or maximum may have occurred and that the interval should be examined more closely for the existence of a possible root.

Open Methods

In the bracketing methods the root is located within an interval prescribed by a lower and an upper bound.

Repeated application of these methods always results in closer estimates of the true value of the root. Such methods are said to be *convergent* because they move closer to the truth as the computation progresses.

In contrast, the *open methods* are based on formulas that require only a single starting value of x or two starting values that do not necessarily bracket the root.

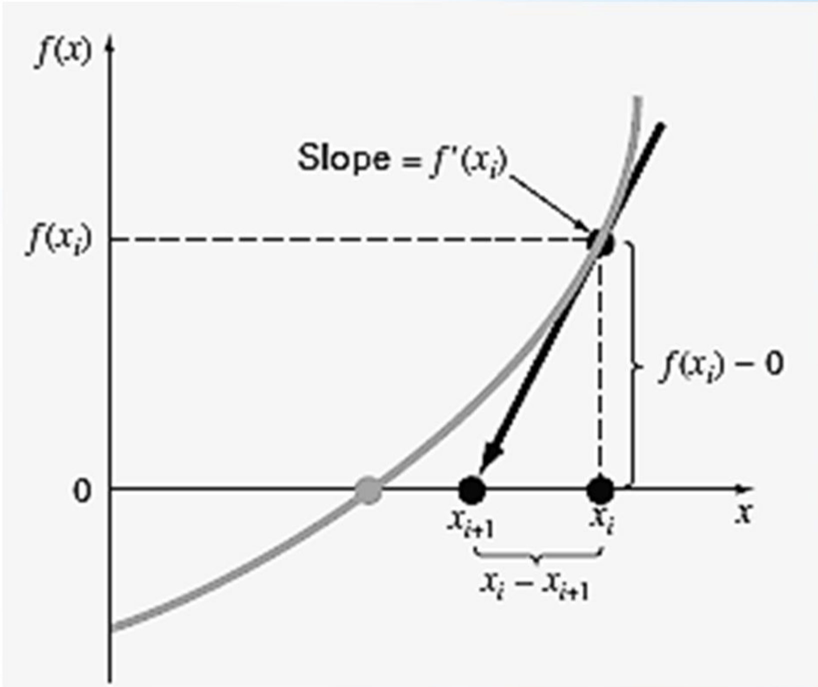
As such, they sometimes *diverge* or move away from the true root as the computation progresses.

However, when the open methods converge, they usually do so much more quickly than the bracketing methods.

Newton-Raphson Method

The most widely used of all root-locating formulas is the Newton-Raphson equation.

If the initial guess at the root is x_i, a tangent can be extended from the point $[x_i, f(x_i)]$. The point where this tangent crosses the x axis usually represents an improved estimate of the root.



The Newton-Raphson method can be derived on the basis of this geometrical interpretation. As in Fig. the first derivative at *x* is equivalent to the slope:

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

Which can be rearranged to yield

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Which is called the Newton-Raphson formula.

Convergence properties of Newton's method for a single equation

When the initial guess is not very close to a solution, Newton's method behaves erratically.

Let us assume now that the current estimate of the solution is indeed near a solution. Then, can we say anything about the rate at which successive Newton iterations converge upon the true solution value?

Suppose X_n differs from the root α by a small quantity ε_n and

$$X_n = \alpha + \varepsilon_n \qquad\qquad X_{n+1} = \alpha + \varepsilon_{n+1}$$

Put these value in Newton Raphson formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$\alpha + \varepsilon_{n+1} = \alpha + \varepsilon_n - \frac{f(\alpha + \varepsilon_n)}{f'(\alpha + \varepsilon_n)}$$

Expand the $f(\alpha+\varepsilon_n)$ and $f'(\alpha+\varepsilon_n)$ using Taylor series about the true value α,

$$\varepsilon_{n+1} = \varepsilon_n - \frac{f(\alpha) + \varepsilon_n f'(\alpha) + \frac{1}{2!}\varepsilon_n{}^2 f''(\alpha) + \cdots \ldots}{f'(\alpha) + \varepsilon_n f''(\alpha) + \frac{1}{2!}\varepsilon_n{}^2 f'''(\alpha) + \cdots \ldots} \qquad \therefore f(\alpha) = 0$$

$$\varepsilon_{n+1} = \varepsilon_n - \frac{\varepsilon_n f'(\alpha) + \frac{1}{2!}\varepsilon_n{}^2 f''(\alpha) + \cdots \ldots}{f'(\alpha) + \varepsilon_n f''(\alpha) + \frac{1}{2!}\varepsilon_n{}^2 f'''(\alpha) + \cdots \ldots}$$

Neglect higher order term and consider upto second order term

$$\varepsilon_{n+1} = \varepsilon_n - \frac{\varepsilon_n f'(\alpha) + \frac{1}{2!}\varepsilon_n{}^2 f''(\alpha)}{f'(\alpha) + \varepsilon_n f''(\alpha)}$$

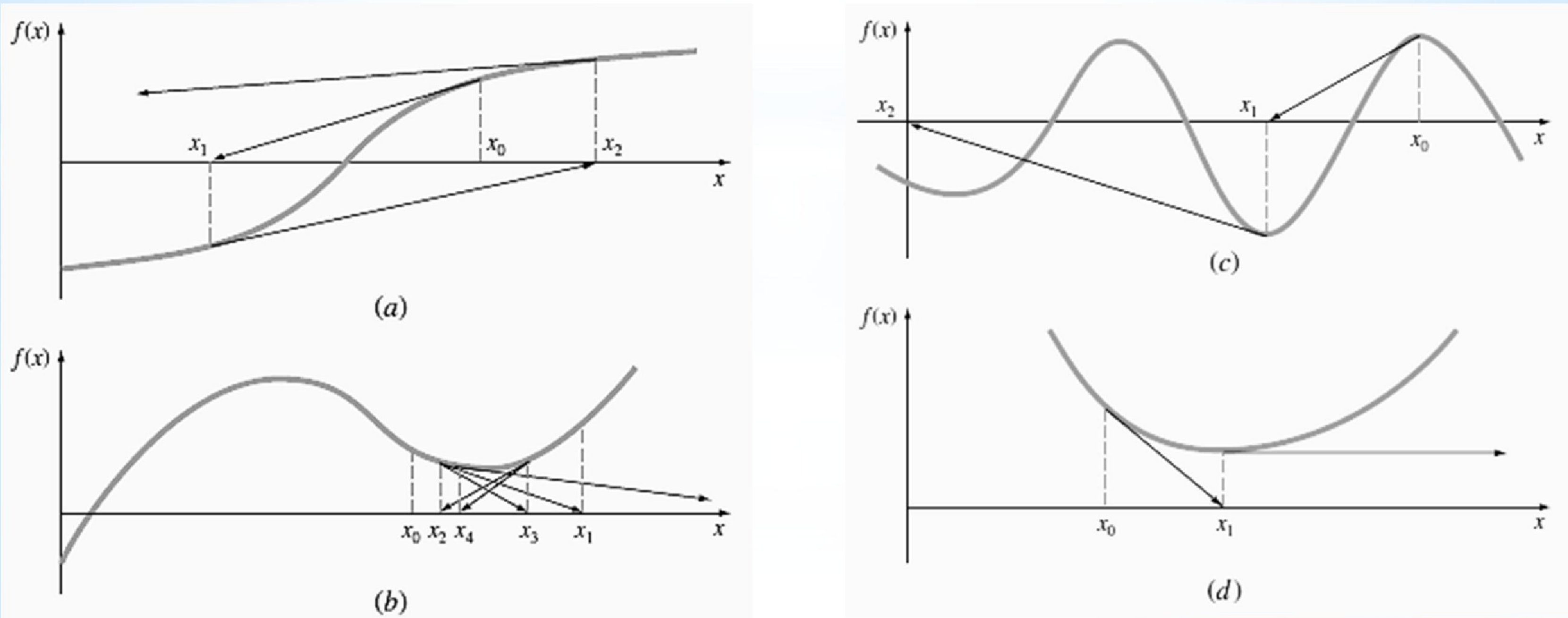
$$\varepsilon_{n+1} = \frac{\varepsilon_n f'(\alpha) + \varepsilon_n{}^2 f''(\alpha) - \varepsilon_n f'(\alpha) - \frac{1}{2!}\varepsilon_n{}^2 f''(\alpha)}{f'(\alpha) + \varepsilon_n f''(\alpha)}$$

$$\varepsilon_{n+1} = \frac{\frac{1}{2}\varepsilon_n{}^2 f''(\alpha)}{f'(\alpha) + \varepsilon_n f''(\alpha)} \qquad \varepsilon_n << 1$$

$$\varepsilon_{n+1} = \frac{\frac{1}{2}\varepsilon_n{}^2 f''(\alpha)}{f'(\alpha)}$$

This shows that subsequent error at each step is proportional to the square of the previous error i.e. convergence is quadratic.

Four cases where the Newton-Raphson method exhibits poor



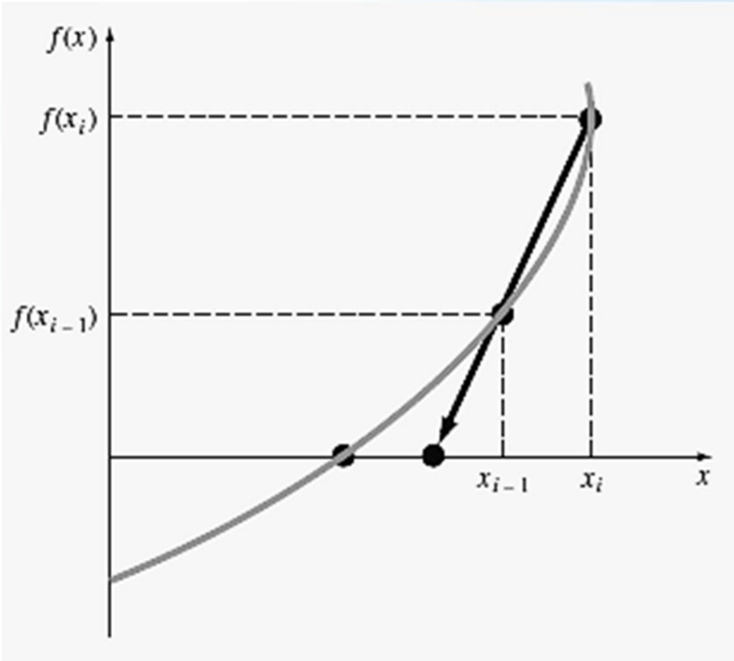
Secant Method

A potential problem in implementing the Newton-Raphson method is the evaluation of the derivative for polynomials and many other functions.

There are certain functions whose derivatives may be extremely difficult or inconvenient to evaluate.

For these cases, the derivative can be approximated by a backward finite divided difference

$$f'(x_i) \cong \frac{f(x_{i-1}) - f(x_i)}{x_{i-1} - x_i}$$

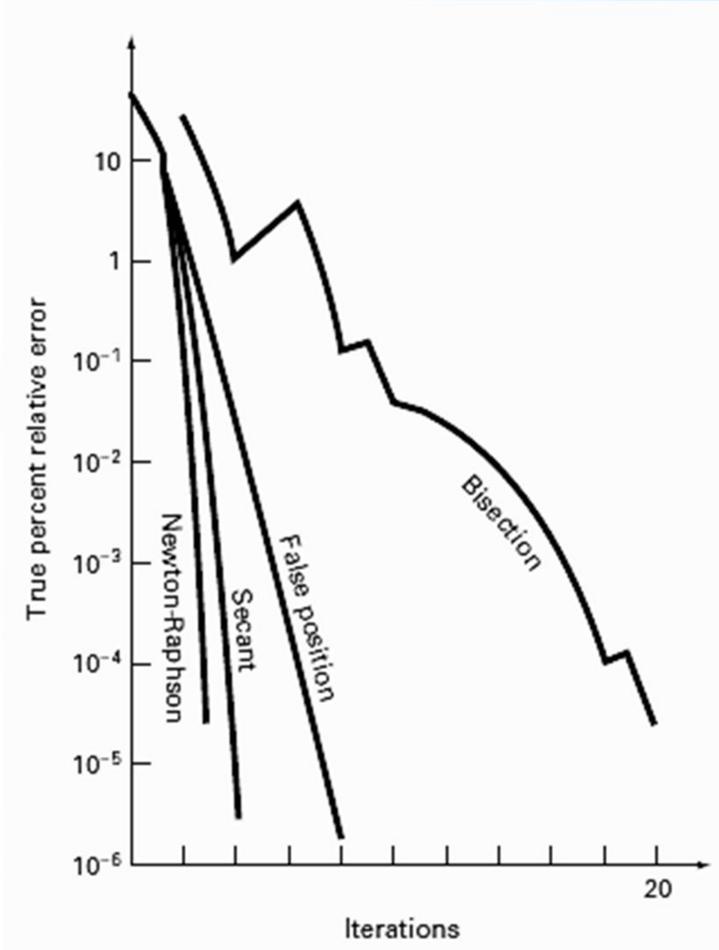


This approximation can be substituted into Eq. to yield the following iterative equation:

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$$

This is *secant method* and the approach requires two initial estimates of *x*.
However, *f*(*x*) is not required to change signs between the estimates, it is not classified as a bracketing method.

Comparison of the true percent relative errors e_t for the methods to determine the roots of
$f(x) = e^{-x} - x$.



Modified Secant Method

Rather than using two arbitrary values to estimate the derivative, an alternative approach involves a fractional perturbation of the independent variable to estimate $f'(x)$,

$$f'(x_i) \cong \frac{f(x_i + \delta x_i) - f(x_i)}{\delta x_i}$$

where δ = a small perturbation fraction.

This approximation can be substituted into Eq. to yield the following iterative equation:

$$x_{i+1} = x_i - \frac{\delta x_i f(x_i)}{f(x_i + \delta x_i) - f(x_i)}$$

The choice of δ value is important. If δ is too small, the method can be swamped by round-off error caused by subtractive cancellation in the denominator. If it is too big, the technique can become inefficient and even divergent.

However, if chosen correctly, it provides a nice alternative for cases where evaluating the derivative is difficult and developing two initial guesses is inconvenient.

Systems of Nonlinear Equations

Locate the roots of a set of simultaneous equations,

$$f_1(x_1, x_2, \ldots, x_n) = 0$$
$$f_2(x_1, x_2, \ldots, x_n) = 0$$
$$- - - - - - - - - - - - - -$$
$$- - - - - - - - - - - - - -$$
$$f_n(x_1, x_2, \ldots, x_n) = 0$$

The solution of this system consists of a set of x values that simultaneously result in all the equations equaling zero.

- **Newton-Raphson**

A multivariable Taylor series must be used to account more than one independent variable contributes to the determination of the root.

Consider equation

$f(x, y) = 0$ and $g(x, y) = 0$

Initial approximation (x_0, y_0)

Better approximation (x_1, y_1) and if this is root of the equations

$x_1 = x_0 + h$ $\quad$ $y_1 = y_0 + k$ $\quad$ so that

$f(x_0 + h, y_0 + k) = 0$ $\quad$ $g(x_0 + h, y_0 + k) = 0$

For the two-variable case, a first order Taylor series can be written

$$f(x_0+h, y_0+k) = f(x_0, y_0) + h\frac{\partial f(x_0, y_0)}{\partial x} + k\frac{\partial f(x_0, y_0)}{\partial y}$$

and

$$g(x_0+h, y_0+k) = g(x_0, y_0) + h\frac{\partial g(x_0, y_0)}{\partial x} + k\frac{\partial g(x_0, y_0)}{\partial y}$$

$$0 = f(x_0, y_0) + h\frac{\partial f(x_0, y_0)}{\partial x} + k\frac{\partial f(x_0, y_0)}{\partial y}$$

$$0 = g(x_0, y_0) + h\frac{\partial g(x_0, y_0)}{\partial x} + k\frac{\partial g(x_0, y_0)}{\partial y}$$

Solve these equation for h and k then we get new approximation

$x_1 = x_0 + h$ $\quad$ and $\quad$ $y_1 = y_0 + k$