Lecture 2
Numerical Methods for Nonlinear Equations

1 Nonlinear Equations

We want to find the numbers x for which

$$f(x) = 0$$

with $f : [a, b] \to \mathbb{R}$ a given real-valued function. Here, we call x a **root** of the equation, or a **zero** of the function f, and we use the Greek letter α to denote it. So, we are looking for a real number $\alpha \in [a, b]$ such that:

$$f(\alpha) = 0.$$

Note that the roots of the equation are exactly x-intercepts of the graph of function f.

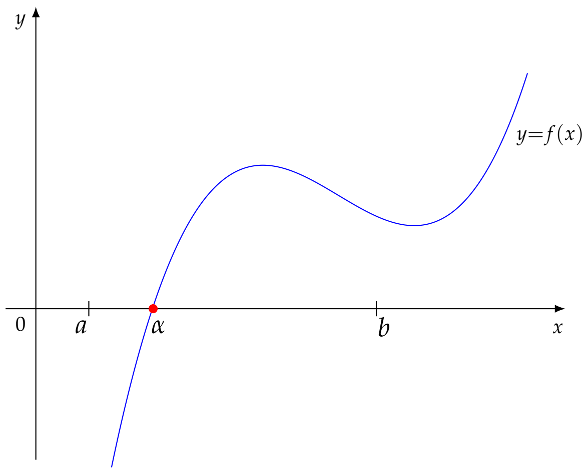

Figure 1: Nonlinear equation with one root

Root-finding 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 computational problem.

A linear equation will have always either exactly one root (slope is nonzero) or no roots at all (slope is zero), since its graph (a straight line) will either intersect the x-axis in only one point or will be parallel with it. The existence and uniqueness of roots for a nonlinear equation is often difficult to determine, and a much wider variety of behavior is possible for graphs of nonlinear functions (generally curved lines). Therefore nonlinear equations can have any number of roots including none.

For example, equation $\ln x - x = 0$ has no solutions (roots), equation $\sqrt{x} - x + 3 = 0$ has one solution, equation $e^{-x} + x^2 - 2$ has two solutions, equation $x^3 - 2x^2 - 5x + 6$ has three solutions and equation $\cos \frac{\pi x}{3} = 0$ has infinitely many solutions.

Finding solutions analytically is often a complicated task. We will focus on calculation of numerical approximations $\widetilde{\alpha}$ to the true solution α, in other words either

$$f(\widetilde{\alpha}) \approx 0 \quad \text{or} \quad |\alpha - \widetilde{\alpha}| \approx 0.$$

These two criteria are not necessarily small simultaneously. We prefer the second criterion, since the first one depends more on the function f (see "noise in function evaluation") example.

Most methods for solving approximately $f(x) = 0$ are **iterative** methods. This means that such a method, given an initial guess x_0 (i.e. starting point), will provide us with a sequence of consecutively computed solutions

$$x_1, x_2, x_3, \ldots, x_n, \ldots \quad \text{such that} \quad x_n \to \alpha \ \text{ as } \ n \to \infty.$$

In other words,

$$\lim_{n\to\infty} x_n = \alpha.$$

Clearly, we must terminate the iteration when x_n is "sufficiently" close to root α. Thus, we need to provide each method with a termination criterion and an error tolerance ε. For example,

$$\text{Stop at iteration } n \text{ when} \quad |\alpha - x_n| < \varepsilon.$$

But, generally, α is not known a priori, therefore, for practical implementation we also need an estimate for the error.

2 Bisection method

We begin with the simplest of such methods, one which most people use for some time.

Suppose that we are given a function f and we assume that we have an interval $[a, b]$ containing the root α. Also, assume that function f is continuous on interval $[a, b]$. We also assume that we are given an error tolerance $\varepsilon > 0$, and we want an approximate root $\widetilde{\alpha} \in [a, b]$ for which

$$|\alpha - \widetilde{\alpha}| < \varepsilon.$$

Bisection method is based on the following theorem from calculus:

Theorem 1 *If $f : [a, b] \to \mathbb{R}$ is a continuous function on closed and bounded interval $[a, b]$ and*

$$f(a) \cdot f(b) < 0$$

then there exists $\alpha \in [a, b]$ such that $f(\alpha) = 0$.

Therefore, further assume that the function f changes sign on interval $[a, b]$. Bisection method begins with initial interval $[a, b]$ and at each iteration, the function is evaluated at the midpoint of the current interval, and half of the interval can then be discarded, keeping the half of the interval on which the function still changes sign.

General scheme for Bisection method

Inputs: function f, endpoints a and b, and error tolerance ε.

Step 1: Compute

$$c = \tfrac{a+b}{2}.$$

Step 2: If $b - c < \varepsilon$, accept c as our root, and then stop.

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

$$sign(f(a)) \cdot sign(f(c)) < 0$$

then replace b with c; Otherwise, replace a with c.

Step 4: Return to Step 1.

Warning!

Note that, it is preferable to check the sign using condition $sign(f(a)) \cdot sign(f(b)) < 0$ instead of using $sign(f(a) \cdot f(b)) < 0$.

In the figure below we present one iteration of the Bisection method. Start with $[a,b] = [a_1, b_1]$, and then compute $c_1 = \frac{a_1+b_1}{2}$. Choose the sub-interval where the function changes its sign (in our case $[a_1, c_1]$). Let $a_2 = a_1$ and $b_2 = c_1$, and repeat. Thus, we will generate a sequence of $c_1, c_2, c_3, \ldots$ such that hopefully $c_n \to \alpha$ as $n \to \infty$.

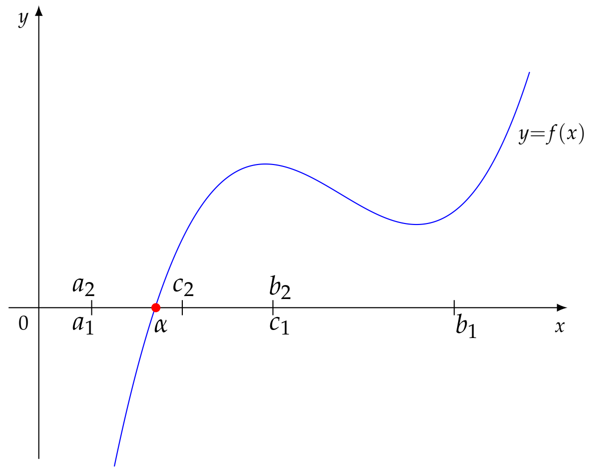

Figure 2: One iteration of bisection method

Example 1 *Consider the function*

$$f(x) = x^6 - x - 1.$$

We want to find the largest root with accuracy of $\varepsilon = 0.001$. *It can be seen from the graph of this function that the root is located in interval* $[1,2]$. *Also, note that the function is continuous (since it is a polynomial).*
Let $a = 1$ *and* $b = 2$.

Since $f(a) = -1$ *and* $f(b) = 61$, *the given function changes its sign and thus all conditions for bisection method are being satisfied. Numerical results are presented in the table below.*

n	a_n	b_n	c_n	$f(c_n)$	$b_n - c_n$
1	1.00000	2.00000	1.50000	$8.891e+00$	$5.000e-01$
2	1.00000	1.50000	1.25000	$1.565e+00$	$2.500e-01$
3	1.00000	1.25000	1.12500	$-9.771e-02$	$1.250e-01$
4	1.12500	1.25000	1.18750	$6.167e-01$	$6.250e-02$
5	1.12500	1.18750	1.15625	$2.333e-01$	$3.125e-02$
6	1.12500	1.15625	1.14063	$6.158e-02$	$1.563e-02$
7	1.12500	1.14063	1.13281	$-1.958e-02$	$7.813e-03$
8	1.13281	1.14063	1.13672	$2.062e-02$	$3.906e-03$
9	1.13281	1.13672	1.13477	$4.268e-04$	$1.953e-03$
10	1.13281	1.13477	1.13379	$-9.598e-03$	$9.766e-04$

Table 1: Numerical results for bisection method. Example 1

We needed 10 iterations of Bisection method to reach the desired accuracy of 10^{-3}. The approximate solution is $\widetilde{\alpha} \equiv c_{10} \approx 1.12279$. Also note that $f(c_n)$ gets closer to 0 as n increases.

2.1 Error Analysis for Bisection Method

Let a_n, b_n and c_n be the values provided by Bisection method at iteration n. Evidently,

$$b_{n+1} - a_{n+1} = \frac{1}{2}(b_n - a_n).$$

Using induction we get

$$\begin{aligned} b_n - a_n &= \frac{1}{2}(b_{n-1} - a_{n-1}), \\ &= \frac{1}{2^2}(b_{n-2} - a_{n-2}), \\ &= \dots \\ &= \frac{1}{2^{n-1}}(b-a). \end{aligned}$$

Since either $\alpha \in [a_n, c_n]$ or $\alpha \in [c_n, b_n]$, we have

$$|\alpha - c_n| \leqslant c_n - a_n = b_n - c_n = \frac{1}{2}(b_n - a_n) = \frac{1}{2^n}(b-a).$$

Therefore, if $\alpha \in [a,b]$ is the root, and c_n is the approximation provided by Bisection method at step n, we have

$$|\alpha - c_n| \leqslant \frac{1}{2^n}(b-a).$$

This relation provides us with a stopping criterion for Bisection method. Moreover, it follows that $c_n \to \alpha$ as $n \to \infty$.

Suppose we want to estimate the number of iterations in Bisection method necessary to find the root with an error tolerance ε,

$$|\alpha - c_n| \leqslant \varepsilon.$$

This will happen if

$$\frac{1}{2^n}(b-a) \leqslant \varepsilon.$$

Solve for n to get

$$n \geqslant \frac{\ln\left(\frac{b-a}{\varepsilon}\right)}{\ln 2}.$$

Thus, using the last inequality, in the case of **Example 1** we get

$$n \geqslant \frac{\ln\left(\frac{1}{0.001}\right)}{\ln 2} \approx 9.97,$$

which is in accordance with numerical results from Table 1.

Advantages and Disadvantages of Bisection Method

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 1/2 with each iteration.

Disadvantages:

1. It is relatively slow when compared with other root-finding methods we will study, especially when the function f (x) has several continuous derivatives about the root α.
2. The algorithm has no check to see whether the ε is too small for the computer arithmetic being used.

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.

3 Newton's method

In a root-finding problem we want to find the root α of a given equation $f(x) = 0$. Thus, we want to find the x-intercept of the graph of function f, in other words the point x at which the graph of $y = f(x)$ intersects the x-axis.

One of the principles of numerical analysis is the following:

Numerical Analysis Principle

If you cannot solve the given problem, then solve a "nearby" problem, whose solution will be close enough to the solution of initial problem.

How do we obtain a nearby problem for equation $f(x) = 0$? Begin first by asking for types of problems which we can solve easily. At the top of the list should be that of finding where a straight line intersects the x-axis.

Thus, we seek to replace equation $f(x) = 0$ by that of solving $p(x) = 0$ for some linear polynomial p that approximates function f in the vicinity of the root α.

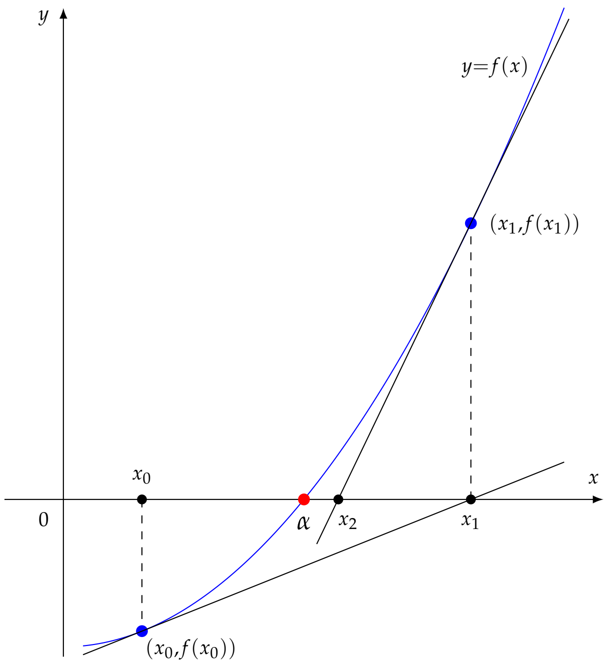

Figure 3: Two iterations of Newton's method

Let x_0 be an initial guess, sufficiently closed to the root α. Consider the tangent line to the graph of f in point $(x_0, f(x_0))$. Tangent line intersects x-axis at point x_1, which is a closer point to α than x_0. Tangent has the following equation (intercept-slope form)

$$p_1(x) = f(x_0) + f'(x_0)(x - x_0).$$

Since $p_1(x_1) = 0$ (look for x-intercept of the tangent), we get

$$f(x_0) + f'(x_0)(x_1 - x_0) = 0.$$

Solve this equation for x_1 to obtain

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

In a similar manner, we could get x_2, by considering as a new guess the point x_1:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}.$$

Repeat this process to obtain in the end a sequence of successive values $x_1, x_2, x_3, \ldots$ that hopefully will converge to root α.

General scheme for Newton's method

Inputs: functions f, f' and initial guess x_0.
Starting with initial guess x_0, compute iterated sequence:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \ldots$$

Iterate until stopping criterion is satisfied.

As a stopping criterion Newton's iterations we can use the condition

$$|x_{n+1} - x_n| < \varepsilon,$$

where ε is the desired error tolerance. The reasons why the above expression may be used as a stopping criterion will be provided later.

Example 2 *Apply Newton's method to the same equation as in* ***Example 1****: $f(x) = x^6 - x - 1 = 0$. Substitute f and its derivative $f'(x) = 6x^5 - 1$ in Newton's method formula to get*

$$x_{n+1} = x_n - \frac{x_n^6 - x_n - 1}{6x_n^5 - 1}, \quad \forall n \in \mathbb{N}.$$

Use initial guess $x_0 = 1.5$. Numerical results are presented below.

n	x_n	$f(x_n)$	$x_n - x_{n-1}$	$\alpha - x_n$
0	1.50000000	$8.89e+01$		
1	1.30049088	$2.54e+01$	$-2.00e-01$	$-3.65e-01$
2	1.18148042	$5.38e-01$	$-1.19e-01$	$-1.66e-01$
3	1.13945559	$4.92e-02$	$-4.20e-02$	$-4.68e-02$
4	1.13477763	$5.50e-04$	$-4.68e-03$	$-4.73e-03$
5	1.13472415	$7.11e-08$	$-5.35e-05$	$-5.35e-05$
6	1.13472414	$1.55e-15$	$-6.91e-09$	$-6.91e-09$

Table 2: Numerical results for Newton's method. Example 2

True solution is $\alpha = 1.134724138$. Observe that we reached the level of accuracy of 10^{-8}. Notice also that

$$x_n - x_{n-1} \approx \alpha - x_n.$$

Thus, $x_n - x_{n-1}$ can be used as an estimate of the true error $\alpha - x_n$ and therefore serve as a stopping criterion in Newton's iterations.

3.1 Division implemented by using Newton's method

Here we consider a division algorithm (based on Newton's method) implemented in some computers in the past. Say, we are interested in computing

$$\frac{a}{b} = a \cdot \frac{1}{b},$$

where $\frac{1}{b}$ is computed using Newton's method. Consider equation

$$f(x) \equiv b - \frac{1}{x} = 0,$$

with b positive. The root of this equation is: $\alpha = \frac{1}{b}$. Let's apply Newton's method to the above equation. Compute derivative of f:

$$f'(x) = \frac{1}{x^2}$$

and Newton's method for this problem becomes

$$x_{n+1} = x_n - \frac{b - \frac{1}{x_n}}{\frac{1}{x_n^2}}.$$

Simplifying the above expression we get

$$x_{n+1} = x_n(2 - bx_n), \quad n \geqslant 0.$$

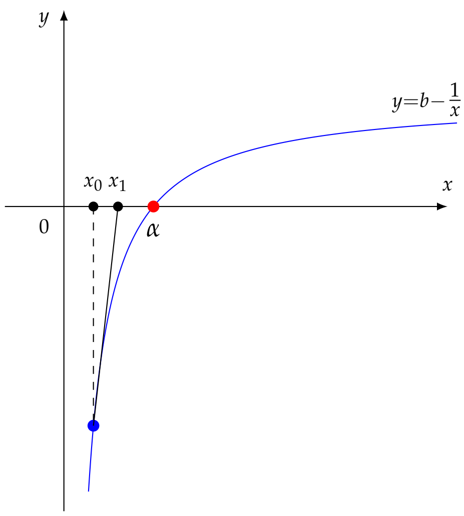

Figure 4: Division example

Initial guess x_0 must be close enough to the true solution and of course $x_0 > 0$. Consider the error

$$\begin{aligned}\alpha - x_{n+1} &= \frac{1}{b} - x_{n+1} \\ &= \frac{1 - bx_{n+1}}{b} \\ &= \frac{1 - bx_n(2 - bx_n)}{b} \\ &= \frac{(1 - bx_n)^2}{b}\end{aligned}$$

and immediately we get

$$\begin{aligned}\mathrm{Rel}(x_{n+1}) &= \frac{\alpha - x_{n+1}}{\alpha} \\ &= \frac{\frac{(1-bx_n)^2}{b}}{\frac{1}{b}} \\ &= (1 - bx_n)^2.\end{aligned}$$

On the other hand,

$$\begin{aligned}\mathrm{Rel}(x_{n+1}) &= \frac{\alpha - x_{n+1}}{\alpha}\\ &= \frac{\frac{1}{b} - x_{n+1}}{\frac{1}{b}}\\ &= 1 - bx_{n+1}.\end{aligned}$$

Thus, we obtained that

$$\mathrm{Rel}(x_{n+1}) = \left(\mathrm{Rel}(x_n)\right)^2.$$

It is an important relation, since it tells us that with every new iteration the error will be squared, i.e. will decrease quadratically.

It can be shown by induction that

$$\mathrm{Rel}(x_{n+1}) = \left(\mathrm{Rel}(x_{n-1})\right)^4 = \ldots = \left(\mathrm{Rel}(x_0)\right)^{2^{n+1}}.$$

Recall that the geometric sequence $\{q^n\}_{n=0}^{\infty}$ will converge to 0 as $n \to \infty$ if and only if $|q| < 1$. In order to guarantee convergence $x_n \to \alpha$ as $n \to \infty$, we ask

$$|\mathrm{Rel}(x_0)| < 1$$

or equivalently

$$0 < x_0 < \frac{2}{b}.$$

In other words, if above condition is satisfied, then the sequence generated by Newton's method for division example will converge. And the "speed" of convergence will be high. For example, suppose that x_0 was chosen such that

$$|\mathrm{Rel}(x_0)| = 0.1.$$

Then

$$\begin{aligned}\mathrm{Rel}(x_1) &= 10^{-2}, \qquad & \mathrm{Rel}(x_2) &= 10^{-4}\\ \mathrm{Rel}(x_3) &= 10^{-8}, \qquad & \mathrm{Rel}(x_4) &= 10^{-16}.\end{aligned}$$

Observe that convergence in this case is really fast (will be called later quadratic convergence).

3.2 Error analysis for Newton's method

Let $f \in C^2[a,b]$ and $\alpha \in [a,b]$ be the root of equation $f(x) = 0$, in other words $f(\alpha) = 0$. Also let $f'(\alpha) \neq 0$.

Consider Taylor formula for function $f(x)$ about point x_n:

$$f(x) = f(x_n) + (x - x_n)f'(x_n) + \frac{(x - x_n)^2}{2} f''(\xi_n),$$

where ξ_n is some (generally unknown) point between x and x_n. Take $x = \alpha$ to get:

$$f(\alpha) = f(x_n) + (\alpha - x_n)f'(x_n) + \frac{(\alpha - x_n)^2}{2} f''(\xi_n).$$

Since $f(\alpha) = 0$, we have

$$0 = \frac{f(x_n)}{f'(x_n)} + (\alpha - x_n) + (\alpha - x_n)^2 \frac{f''(\xi_n)}{2f'(x_n)}.$$

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

$$\alpha - x_{n+1} = (\alpha - x_n)^2 \left[\frac{-f''(\xi_n)}{2f'(x_n)}\right].$$

For previous example, $f''(x) = 30x^4$, and we get

$$\frac{-f''(\xi_n)}{2f'(x_n)} \approx \frac{-f''(\alpha)}{2f'(\alpha)} = \frac{-30\alpha^4}{2(6\alpha^5 - 1)} \approx -2.42.$$

Therefore, we could write that

$$\alpha - x_{n+1} \approx -2.42(\alpha - x_n)^2.$$

For example, if $n = 3$, we get $\alpha - x_3 \approx -4.73e - 03$ and using the above estimation we obtain

$$\alpha - x_4 \approx -2.42(\alpha - x_3)^2 \approx -5.42e - 05,$$

a result which is in accordance with the result from the table:

$$\alpha - x_4 \approx -5.35e - 05.$$

If iteration x_n is close to α, we could denote

$$\frac{-f''(\xi_n)}{2f'(x_n)} \approx \frac{-f''(\alpha)}{2f'(\alpha)} \equiv M$$

and get an important formula that describes the behavior of the error for Newton's method

$$\boxed{\alpha - x_{n+1} \approx M(\alpha - x_n)^2.}$$

Multiplying both sides by M we obtain

$$M(\alpha - x_{n+1}) \approx (M(\alpha - x_n))^2.$$

Inductively,

$$M(\alpha - x_{n+1}) \approx (M(\alpha - x_0))^{2^n}, \quad n = 0, 1, 2, \ldots.$$

In other words, in order to guarantee the convergence of Newton's method we should have

$$|M(\alpha - x_0)| < 1,$$

from which we get

$$|\alpha - x_0| < \frac{1}{|M|} = \left|\frac{2f'(\alpha)}{f''(\alpha)}\right|.$$

Thus, in order for the Newton's method to converge, the initial guess x_0 should be chosen close enough to the root α, such that it satisfies the above inequality.

Advantages and Disadvantages of Newton's Method

Advantages:

1. Fast convergence (usually quadratic convergence).
2. There is a stopping criterion.

Disadvantages:

1. It may fail to converge (for example, when initial guess is not close enough).
2. It needs also the derivative of the function.
3. There is no guaranteed error bound for the computed iterates.
4. It is likely to have difficulty if $f'(\alpha) = 0$. This means the $x-$axis is tangent to the graph of $y = f(x)$ at $x = \alpha$.

4 Secant method

Newton's method was based on using the line tangent to the curve of $y = f(x)$ at point $(x_0, f(x_0))$. When $x_0 \approx \alpha$, the graph of the tangent line is approximately the same as the graph of $y = f(x)$ around $x = \alpha$. Then the root of the tangent line is used to approximate α.

Consider using an approximating line based on "interpolation". Assume there are two estimates of the root α, say x_0 and x_1. Then, produce a linear function

$$q(x) = a_0 + a_1 x, \text{ such that}$$
$$q(x_0) = f(x_0), \qquad q(x_1) = f(x_1).$$

This line is called a **secant line** through points x_0 and x_1.

Equation of secant line is given by

$$q(x) = \frac{(x_1 - x)f(x_0) + (x - x_0)f(x_1)}{x_1 - x_0}.$$

This equation is linear in x. Solve the equation $q(x) = 0$, denoting its root by x_2:

$$x_2 = x_1 - f(x_1) : \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

We can now repeat the whole process: use x_1 and x_2 to produce another secant line, and then use its x-intercept, x_3, to approximate α. This yields the general iteration formula:

$$x_{n+1} = x_n - f(x_n) : \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}, \quad n = 1, 2, 3, \ldots.$$

This is called the **Secant method** for solving equation $f(x) = 0$.

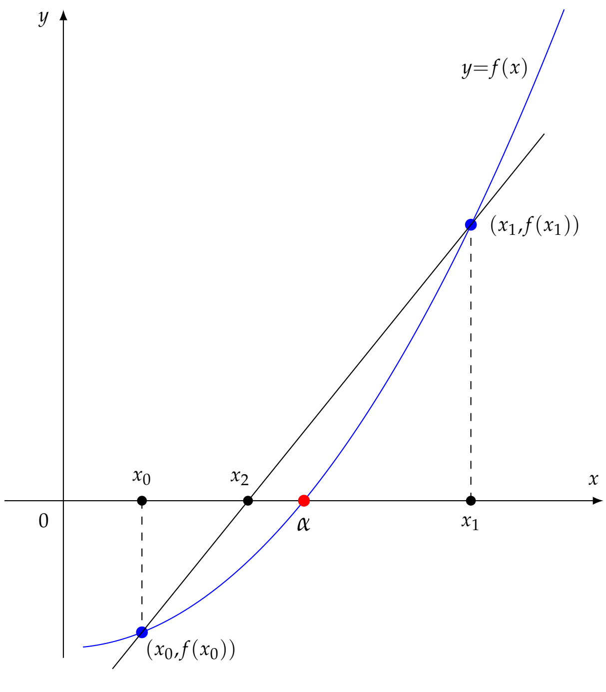

Figure 5: One iteration of the Secant method

General scheme for Secant method

Inputs: function f, initial guesses x_0 and x_1.
Starting with initial guesses x_0 and x_1, compute iterated sequence:

$$x_{n+1} = x_n - f(x_n) : \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}, \quad n = 1, 2, 3, \ldots$$

Iterate until stopping criterion is satisfied.

As a stopping criterion we can use the same one as in Newton's method i.e.

$$\text{Stop at iteration } n \text{ when} \quad |x_{n+1} - x_n| < \varepsilon.$$

Example 3 *Apply Secant method to find the zero of the function from Examples 1 and 2, $f(x) \equiv x^6 - x - 1 = 0$ with initial guesses $x_0 = 2$ and $x_1 = 0$. Numerical results are presented in the table below.*

n	x_n	$f(x_n)$	$x_n - x_{n-1}$	$\alpha - x_n$
0	2.0	61.0		
1	1.0	-1.0	$-1.00e+00$	
2	1.01612903	$-9.15e-01$	$1.61e-02$	$1.35e-01$
3	1.19057777	$6.57e-01$	$1.74e-01$	$1.19e-01$
4	1.11765583	$-1.68e-01$	$-7.29e-02$	$-5.59e-02$
5	1.13253155	$-2.24e-02$	$1.49e-02$	$1.71e-02$
6	1.13481681	$9.54e-04$	$2.29e-03$	$2.19e-03$
7	1.13472414	$-5.07e-06$	$-9.32e-05$	$-9.72e-05$
8	1.13472414	$-1.13e-09$	$4.92e-07$	$4.29e-07$

Observe that in comparison with Newton's method we will need more iterations to achieve a desired level of accuracy, but less iterations will be required compared to Bisection method. Also, as in Newton's method example 2, we can see that

$$x_n - x_{n-1} \approx \alpha - x_n,$$

meaning that $x_n - x_{n-1}$ can be used as an estimate for the error and thus can serve as a stopping criterion for the iterative process.

Note as well that the Secant method does not require a knowledge of $f'(x)$, whereas Newton's method requires both $f(x)$ and $f'(x)$.

Note also that the Secant method

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

can be considered as an approximation of the Newton's method:

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

by using the approximation of the derivative:

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

4.1 Convergence Analysis for Secant method

It can be shown that the Secant method error satisfies the following relation

$$\alpha - x_{n+1} = (\alpha - x_n)(\alpha - x_{n-1}) \cdot \left[-\frac{f''(\xi_n)}{2f'(\zeta_n)} \right]$$

with ξ_n and ζ_n unknown points located between the minimum and maximum of $\{x_{n-1}, x_n, \alpha\}$.
Recall that the Newton's method iterations satisfied

$$\alpha - x_{n+1} = (\alpha - x_n)^2 \cdot \left[-\frac{f''(\xi_n)}{2f'(x_n)} \right]$$

From the error formula for the Secant method it follows that $x_n \to \alpha$ as $n \to \infty$ assuming that x_0 and x_1 are chosen sufficiently close to α. Moreover, it can be proved (assuming that $f \in C^2([a,b])$) that

$$\lim_{n\to\infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|^r} = \left|\frac{f''(\alpha)}{2f'(\alpha)}\right|^{r-1},$$

where $r = \frac{1+\sqrt{5}}{2} \approx 1.62$. The last formula says that when we are close to α, then

$$\boxed{|\alpha - x_{n+1}| \approx C|\alpha - x_n|^r}$$

with

$$C = \left|\frac{f''(\alpha)}{2f'(\alpha)}\right|^{r-1} \text{ and } r \approx 1.62.$$

This looks very much like the Newton's method result:

$$|\alpha - x_{n+1}| \approx M|\alpha - x_n|^2$$

with

$$M = \left|\frac{f''(\alpha)}{2f'(\alpha)}\right|.$$

Obviously, the Secant method converge slower than Newton's method ($r = 1.62$ versus $r = 2$). Both the secant and Newton's methods converge at faster than a linear rate (as Bisection method), and thus they are called **super-linear** methods.

It can be proved that for Secant method we have

$$\lim_{n\to\infty} \frac{|x_{n+1} - x_n|}{|\alpha - x_n|} = 1$$

and therefore,

$$|\alpha - x_n| \approx |x_{n+1} - x_n|$$

is a good error estimator for the true error.

Warning!

Do not combine the secant method formula and write it in the form

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

This has enormous loss of significance error as compared with the earlier formulation:

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

4.2 Costs of Secant and Newton's methods

Observe that the Newton's method

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

requires two function evaluations per iteration: one for the function itself and another one for its derivative. Whilst the Secant method

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

requires only one function evaluation per iteration, following the initial step.

For this reason, the Secant method is often faster in time, even though more iterates are needed with it than with Newton's method to attain a similar accuracy. In what follows we will answer the question when the Secant method will be faster than Newton's method.

Let T_N and T_S be CPU times used to compute the root α with a given accuracy using Newton's method and Secant method, respectively. If

$$\frac{T_S}{T_N} < 1,$$

then clearly Secant method is faster.

Let t_f be the time needed to evaluate function f. Suppose that time needed to evaluate derivative of f is $(1+s) \cdot t_f$, where s the relative increase in computing time for evaluating the derivative. It can be shown that

$$\frac{T_S}{T_N} = \frac{t_f}{t_f + s \cdot t_f} \cdot \frac{\ln 2}{\ln r},$$
$$\frac{T_S}{T_N} = \frac{1}{1+s} \cdot \frac{\ln 2}{\ln r},$$

where $r \approx 1.62$. From the last relation we get that Secant method will be faster if

$$\frac{T_S}{T_N} = \frac{1}{1+s} \cdot \frac{\ln 2}{\ln r} < 1.$$

Solving for s we obtain

$$s > \frac{\ln 2}{\ln r} - 1 \approx 0.44.$$

In other words, if CPU time needed to evaluate $f'(x)$ is bigger than 44% from the time needed to evaluate $f(x)$, then we should prefer the Secant method, and not Newton's.

Advantages and Disadvantages of Secant Method

Advantages:

1. Secant method converges at faster than a linear rate, so that it is more rapidly convergent than the bisection method.
2. It does not require use of the derivative of the function, something that is not available in a number of applications.
3. It requires only one function evaluation per iteration, as compared with Newton's method which requires two.

Disadvantages:

1. Secant method may not converge.
2. There is no guaranteed error bound for the computed iterates.
3. It is likely to have difficulty if $f'(\alpha) = 0$. This means the $x-$axis is tangent to the graph of $y = f(x)$ at $x = \alpha$.
4. Newton's method generalizes more easily to new methods for solving simultaneous systems of nonlinear equations.

5 Brent's method

Richard Brent devised a method combining the advantages of the bisection method and the secant method:

- It is guaranteed to converge.
- It has an error bound which will converge to zero in practice.
- For most problems $f(x) = 0$, with $f(x)$ differentiable about the root α, the method behaves like the secant method.
- In the worst case, it is not too much worse in its convergence than the bisection method.

In Matlab, it is implemented as $fzero$; and it is presented in most Fortran numerical analysis libraries.

6 Order of convergence

Besides the convergence of iterations generated by a numerical method, often we would like to know how fast is the sequence x_n converging to a solution α. In other words, we want to know how fast the error $\alpha - x_n$ is decreasing toward 0.

Definition 1 *We say that sequence $\{x_n\}_{n=0}^{\infty}$ converges to α with order of convergence $p \geqslant 1$, if*

$$|\alpha - x_{n+1}| \leqslant C|\alpha - x_n|^p, \quad n = 0, 1, 2, 3, \dots,$$

*where $C \geqslant 0$ is a constant. Cases $p = 1, 2, 3$ are called **linear**, **quadratic** and **cubic** convergences. In case of linear convergence, constant C is called the **rate of linear convergence**.*

Thus, the Bisection method is linearly convergent with rate 0.5, since it was shown in section 2.1 that

$$|\alpha - x_{n+1}| \leqslant \tfrac{1}{2}|\alpha - x_n|,$$

Newton's method is quadratically convergent since from section 3.2 we have,

$$|\alpha - x_{n+1}| \approx M|\alpha - x_n|^2,$$

and the Secant method has order of convergence $p = \frac{1+\sqrt{5}}{2} \approx 1.62$, as it was shown in section 4.1

$$|\alpha - x_{n+1}| \approx C|\alpha - x_n|^{1.62}.$$

7 Fixed point iteration

Another approach in deriving root-finding methods is based on so-called **fixed point iterations**. Let's start with some examples.

Example 4 *Consider solving the following two equations with solution α*

$$\begin{aligned} E1: &\quad x = 1 + 0.5\sin x, \quad \alpha = 1.49870113351785, \\ E2: &\quad x = 3 + 2\sin x, \quad \alpha = 3.09438341304928. \end{aligned}$$

In order to solve these equations, we are going to use a numerical scheme called "fixed point iteration". It amounts to making an initial guess of x_0 and substituting this into the right side of the equation.

$$\begin{aligned} x_1 &= 1 + 0.5\sin x_0, \\ x_2 &= 1 + 0.5\sin x_1, \\ &\dots \\ x_n &= 1 + 0.5\sin x_{n-1}, \\ &\dots \end{aligned}$$

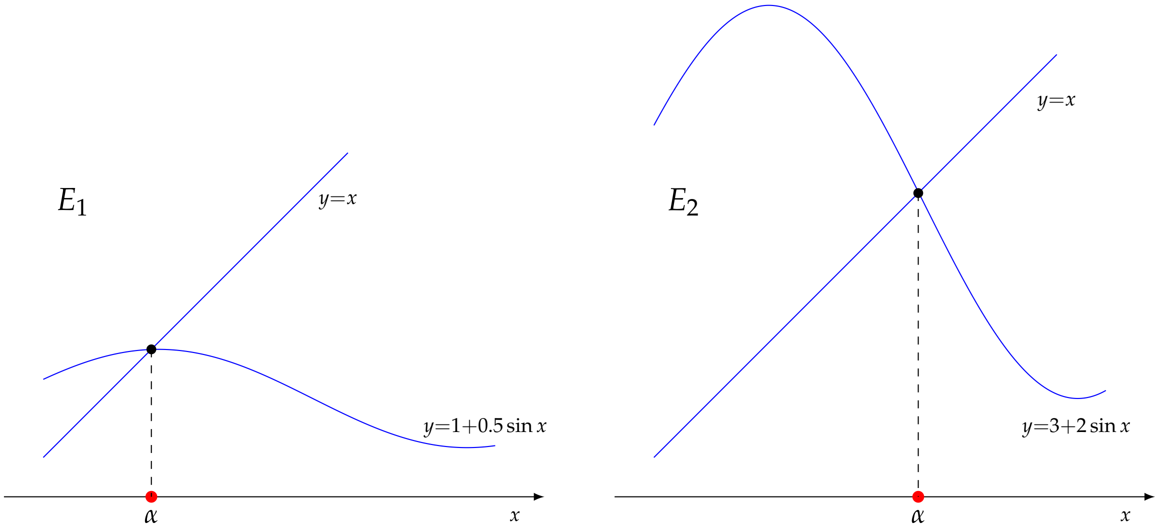

Figure 6: Fixed point Example 4

	$E1$	$E2$
n	x_n	x_n
0	0.00000000000000	3.00000000000000
1	1.00000000000000	3.28224001611973
2	1.42073549240395	2.71963177181556
3	1.49438099256432	3.81910025488514
4	1.49854088439917	1.74629389651652
5	1.49869535552190	4.96927957214762
6	1.49870092540704	1.06563065299216
7	1.49870112602244	4.75018861639465
8	1.49870113324789	1.00142864236516
9	1.49870113350813	4.68448404916097
10	1.49870113351750	1.00077863465869

Table 3: Fixed point iteration numerical results for Example 4

This is repeated until convergence occurs or until the iteration is terminated. The above fixed point iterations can be written symbolically as

$$
\begin{aligned}
E1: &\quad x_{n+1} = 1 + 0.5\sin x_n, \quad n = 0, 1, 2, \ldots \\
E2: &\quad x_{n+1} = 3 + 2\sin x_n, \quad n = 0, 1, 2, \ldots
\end{aligned}
$$

In Table 3 there are presented numerical results for 10 fixed point iterations.

Clearly convergence is occurring with $E1$, but not with $E2$. Why does one of these iterations converge, but not the other? The graphs show similar behavior, so why the difference? Before answering these questions, consider another example.

Example 5 *Suppose we are solving the equation* $x^2 - 5 = 0$ *with exact root* $\alpha = \sqrt{5} \approx 2.2361$ *using fixed point iterates of the form*

$$x_{n+1} = g(x_n).$$

Consider four different iterations:

$$\begin{aligned}
\mathbf{I}_1&: \quad x_{n+1} = 5 + x_n - x_n^2; \\
\mathbf{I}_2&: \quad x_{n+1} = \frac{5}{x_n}; \\
\mathbf{I}_3&: \quad x_{n+1} = 1 + x_n - \frac{1}{5}x_n^2; \\
\mathbf{I}_4&: \quad x_{n+1} = \frac{1}{2}\left(x_n + \frac{5}{x_n}\right).
\end{aligned}$$

It can be seen that if we suppose that $\lim_{n\to\infty} x_n = \alpha$ exists, then α must equal $\sqrt{5}$. Therefore, one might suspect that all four fixed point sequences will converge to $\sqrt{5} \approx 2.2361$.

Numerical results for all four fixed point iterations are presented in the Table 5 below:

	$\mathbf{I}_1$	$\mathbf{I}_2$	$\mathbf{I}_3$	$\mathbf{I}_4$
n	x_n	x_n	x_n	x_n
0	$1.0e+00$	1.0	1.0	1.0
1	$5.0000e+00$	5.0	1.8000	3.0000
2	$-1.5000e+01$	1.0	2.1520	2.3333
3	$-2.3500e+02$	5.0	2.2258	2.2381
4	$-5.5455e+04$	1.0	2.2350	2.2361
5	$-3.0753e+09$	5.0	2.2360	2.2361
6	$-9.4575e+18$	1.0	2.2361	2.2361
7	$-8.9445e+37$	5.0	2.2361	2.2361
8	$-8.0004e+75$	1.0	2.2361	2.2361

Table 4: Numerical results for Example 5.

Again, observe in some cases we have convergence, and in others the iterations failed to converge. Thus, iterations $\mathbf{I_1}$ and $\mathbf{I_2}$ aren't converging, but $\mathbf{I_3}$ and $\mathbf{I_4}$ are converging. Moreover, the last one converges faster than the third.

As another example of a fixed point iteration, note that the Newton's method itself

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

is also a fixed point iteration, for the equation

$$x = x - \frac{f(x)}{f'(x)} = g(x).$$

In general, we are interested in solving equations of the form

$$x = g(x)$$

by means of fixed point iterations:

$$x_{n+1} = g(x_n) \quad \text{with } n = 0, 1, 2, 3, \ldots$$

General scheme for fixed point iteration

Inputs: function g, initial guess x_0, tolerance ε.
Starting with initial guess x_0, compute iterated sequence:

$$x_{n+1} = g(x_n), \quad n = 0, 1, 2, \ldots$$

Iterate until stopping criterion is satisfied.

It is called "**fixed point iteration**" because the root α is a so-called fixed point for the function g, meaning that α is a real number such that

$$g(\alpha) = \alpha.$$

First let's start by asking under what conditions a function will have one or more fixed points, i.e. by asking whether there is a solution to equation

$$x = g(x).$$

For this to happen, the graphs of $y = x$ and $y = g(x)$ must intersect as in Figure 6. There are several lemmas and theorems from Calculus that give conditions under which we are guaranteed that there is a fixed point i.e. $\alpha = g(\alpha)$. One such Lemma is presented below.

Lemma 1 *Let $g \in C[a,b]$ (continuous function on $[a,b]$) and suppose it satisfies the property (called **contracting**):*

$$a \leqslant x \leqslant b \Rightarrow a \leqslant g(x) \leqslant b,$$
$$i.e. \quad g([a,b]) \subset [a,b].$$

Then, the equation $x = g(x)$ has at least one solution in the interval $[a,b]$.

The following theorem ensures the convergence of fixed point iterations.

Theorem 2 *Assume $g(x) \in C^1[a,b]$ and $g([a,b]) \subset [a,b]$ (i.e. contracting property holds). Define*

$$\lambda = \max_{a \leqslant x \leqslant b} |g'(x)|.$$

Further assume that $\lambda < 1$.Then the following statements hold

$$\begin{aligned}
&(S1) \qquad \text{there exists a unique } \alpha \in [a,b] \text{ such that } \alpha = g(\alpha);\\
&(S2) \qquad \text{for any } x_0 \in [a,b], \text{ the fixed point iteration } x_{n+1} = g(x_n) \to \alpha.\\
&(S3) \qquad |\alpha - x_n| \leqslant \frac{\lambda^n}{1-\lambda}|x_1 - x_0|, \quad n \geqslant 1;\\
&(S4) \qquad \lim_{n\to\infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|} = g'(\alpha).
\end{aligned}$$

The identity from the forth statement $S(4)$

$$\lim_{n\to\infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|} = g'(\alpha)$$

tells us that if x_n is sufficiently close to α then

$$|\alpha - x_{n+1}| \approx g'(\alpha)\,|\alpha - x_n|$$

in other words, the errors will decrease by a constant factor of $g'(\alpha)$. If this is negative, then the errors will oscillate between positive and negative values, and the iterates will be approaching α from both sides. When $g'(\alpha)$ is positive, the iterates will approach α from only one side.

Also we can see what happens when $g'(\alpha) > 1$. In this case the errors will increase in size as we approach the root rather than decrease. Figure 7 confirms these findings.

Consider first fixed point Example 4:

$$\begin{aligned}
&(E1) \qquad x = 1 + 0.5\sin x,\\
&(E2) \qquad x = 3 + 2\sin x.
\end{aligned}$$

For equation $(E1)$ we have $g(x) = 1 + 0.5\sin x$ and $\alpha \approx 1.4987$. Compute $g'(\alpha)$:

$$g'(x) = 0.5\cos x,$$
$$g'(\alpha) \leqslant 0.5$$

and therefore according to the above observation, fixed point iterations $x_{n+1} = 1 + 0.5\sin x_n$ will converge to α.

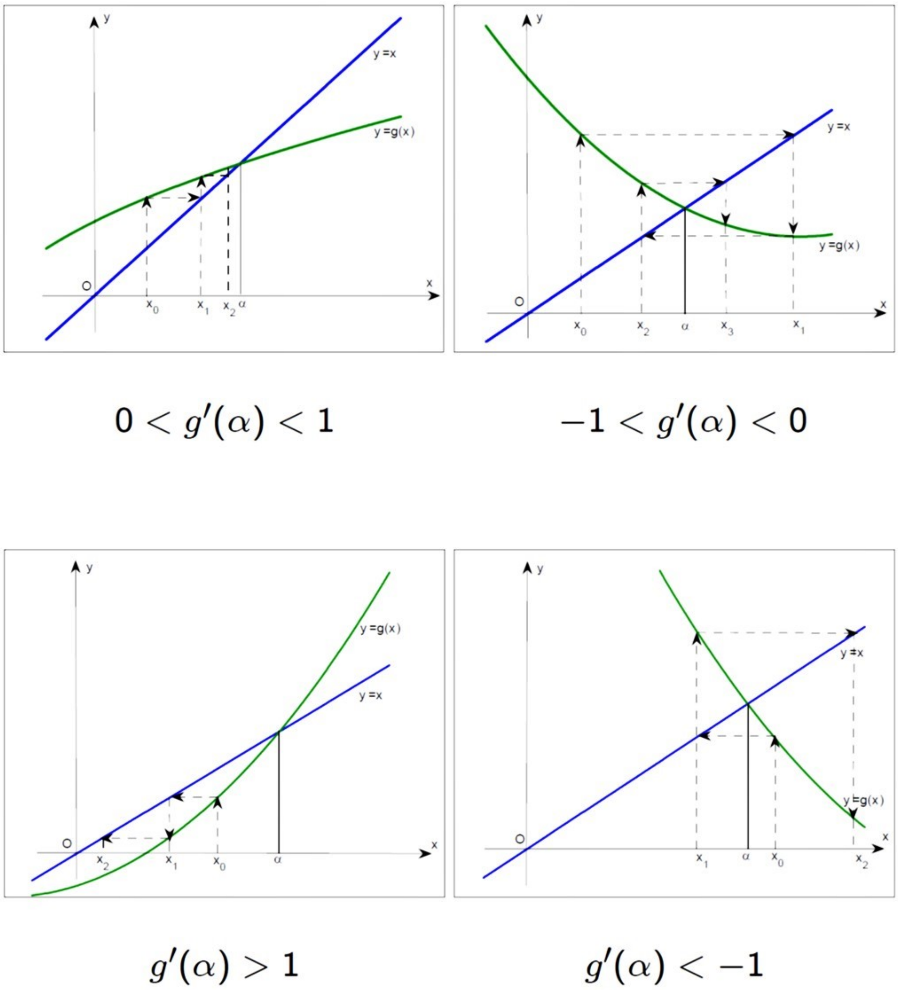

Figure 7: Fixed point iterations

For equation $E2$ we have $g(x) = 3 + 2\sin x$ and $\alpha \approx 3.0944$, and

$$\begin{aligned} g'(x) &= 2\cos x, \\ g'(\alpha) &= g'(3.09438341304928) \approx -1.998 \end{aligned}$$

and thus the fixed point iterations $x_{n+1} = 3 + 2\sin x_n$ will diverge.

Consider second fixed point Example 5: equation $x^2 = 5$ with solution $\alpha = \sqrt{5}$.

Case $(\mathbf{I}_1)$: $\quad g(x) = 5 + x - x^2$.

$$\begin{aligned} g'(x) &= 1 - 2x, \\ g'(\sqrt{5}) &= 1 - 2\sqrt{5} < -1. \end{aligned}$$

Thus, $x_n = g(x_{n-1})$ will not converge to $\sqrt{5}$.

Case $(\mathbf{I}_2)$: $\quad g(x) = \frac{5}{x}$.

$$\begin{aligned} g'(x) &= -\frac{5}{x^2}, \\ g'(\sqrt{5}) &= -1. \end{aligned}$$

So, $x_n = g(x_{n-1})$ may converge or diverge, but results show that it doesn't converge to $\sqrt{5}$.

Case ($\mathbf{I}_3$) : $\quad g(x) = 1 + x - \frac{1}{5}x^2$.

$$g'(x) = 1 - \frac{2}{5}x,$$
$$g'(\sqrt{5}) = 1 - \frac{2}{5}\sqrt{5} \approx 0.106.$$

Therefore, $x_n = g(x_{n-1}) \to \sqrt{5}$. Moreover,

$$\left|\sqrt{5} - x_{n+1}\right| \approx 0.106 \left|\sqrt{5} - x_n\right|,$$

This is a linear convergence with rate 0.106.

Case ($\mathbf{I}_4$) : $\quad g(x) = \frac{1}{2}\left(x + \frac{5}{x}\right)$.

$$g'(x) = \frac{1}{2}\left(1 - \frac{5}{x^2}\right),$$
$$g'(\sqrt{5}) = 0.$$

Thus, $x_n = g(x_{n-1}) \to \sqrt{5}$ is convergent with an order of convergence higher than 1. Also, it can be observed that this last fixed point iteration is nothing else but Newton's method applied to equation $x^2 - 5 = 0$.

Sometimes it is difficult to express initial equation $f(x) = 0$ in the fixed point form $x = g(x)$ such that the resulting iterates will converge. Such a process is presented in the following examples.

Example 6 *Let equation $x^4 - x - 1 = 0$ be rewritten as*

$$x = \sqrt[4]{1 + x},$$

which will provide us with iterations

$$x_0 = 1, \quad x_{n+1} = \sqrt[4]{1 + x_n}.$$

This sequence will converge to $\alpha \approx 1.2207$.

Example 7 *Let equation $x^3 + x - 1 = 0$ be rewritten as*

$$x = \frac{1}{1 + x^2}.$$

That will provide us with iterations

$$x_0 = 1, \quad x_{n+1} = \frac{1}{1 + x_n^2},$$

which will converge to $\alpha \approx 0.6823$.

7.1 Order of convergence of fixed point iterations

If $|g'(\alpha)| < 1$, then we have that iterations $x_n = g(x_{n-1})$ are at least linear convergent.
If additionally, $g'(\alpha) \neq 0$, then we have exactly linear convergence with rate $g'(\alpha)$.

Consider Newton's method to find the root α of equation $f(x) = 0$

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

as a fixed point method with

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Compute $g'(x)$:

$$g'(x) = -\frac{f(x)f''(x)}{\left(f'(x)\right)^2}.$$

Since α is a root, then $f(\alpha) = 0$ and therefore

$$g'(\alpha) = 0.$$

Thus, Newton's method will converge and it will be faster than linear convergence.

7.2 Error analysis of fixed point iterations

Consider the fixed point iteration $x_n = g(x_{n-1})$ and suppose that it is convergent to a fixed point $\alpha = g(\alpha)$. Recall the formula:

$$\lim_{n\to\infty} \frac{\alpha - x_n}{\alpha - x_{n-1}} = g'(\alpha) = \lambda.$$

Thus, if $|\lambda| < 1$, we can write

$$\alpha - x_n \approx \lambda(\alpha - x_{n-1}).$$

If λ is known, then we can solve for α the above equation to get

$$\begin{aligned}\alpha &\approx \frac{x_n - \lambda x_{n-1}}{1-\lambda}\\ &= \frac{x_n - \lambda x_n}{1-\lambda} + \frac{\lambda x_n - \lambda x_{n-1}}{1-\lambda}\\ &= x_n + \frac{\lambda}{1-\lambda}\Big[x_n - x_{n-1}\Big].\end{aligned}$$

Therefore, if λ is known, we can obtain a new approximation for α, given by the so-called **extrapolation** of x_{n-1} and x_n:

$$\boxed{\alpha \approx x_n + \frac{\lambda}{1-\lambda}\Big[x_n - x_{n-1}\Big].}$$

But how to compute λ? From formula

$$\alpha - x_n \approx \lambda(\alpha - x_{n-1}),$$

solving for λ we can get

$$\lambda \approx \frac{\alpha - x_n}{\alpha - x_{n-1}}.$$

Consider sequence of ratios:

$$\lambda_n = \frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}},$$

and since $x_n \to \alpha$, we easily prove that $\lambda_n \to \lambda$ as $n \to \infty$.

Thus, once x_{n-2}, x_{n-1} and x_n are known, by substituting in extrapolation formula λ with λ_n, we get the so-called **Aitken's extrapolation formula**:

$$\boxed{\hat{x}_n = x_n + \frac{\lambda_n}{1-\lambda_n}\Big[x_n - x_{n-1}\Big],}$$

where

$$\lambda_n = \frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}}.$$

Obviously, $\hat{x}_n$ is an approximation of α, much better than x_n. Moreover,

$$\alpha - x_n \approx \hat{x}_n - x_n = \frac{\lambda_n}{1-\lambda_n}\Big[x_n - x_{n-1}\Big],$$

which is known as **Aitken's error estimation formula** and can be used to estimate the error in fixed point iterations.

Example 8 *Consider solving for fixed point the equation $x = 2\pi + \sin x$ using fixed point iterations*

$$x_n = 2\pi + \sin x_{n-1}, \quad n = 0, 1, 2, \dots$$

with initial guess $x_0 = 6$. Numerical results are presented in the Table below.

n	x_n	λ_n	$\alpha - x_n$	Aitken Estimate
0	6.0000000		$1.55e-02$	
1	6.0005845		$1.49e-02$	
2	6.0011458	0.9603	$1.44e-02$	$1.36e-02$
3	6.0016848	0.9604	$1.38e-02$	$1.31e-02$
4	6.0022026	0.9606	$1.33e-02$	$1.26e-02$
5	6.0027001	0.9607	$1.28e-02$	$1.22e-02$
6	6.0031780	0.9609	$1.23e-02$	$1.17e-02$
7	6.0036374	0.9610	$1.18e-02$	$1.13e-02$

Table 5: Numerical results for Example

First, observe a very slow linear convergence $\alpha - x_n \to 0$.

Explanations lie in the third column. Note that sequence of λ_n converge slowly towards $\lambda \approx 0.97$. Actually,

$$\lambda \approx 0.9644. = g'(\alpha)$$

and thus

$$\alpha - x_n \approx 0.9644(\alpha - x_{n-1}),$$

which confirms the linear convergence with rate 0.9644. This rate (close to 1) tells us that with each iteration the error decreases only by 3.56%.

Second, observe that Aitken's estimate (last column in the table) approximates well the true error (the 4th column), and therefore it can be used to estimate the error in fixed point iteration methods.

$$\alpha - x_n \approx \hat{x}_n - x_n = \frac{\lambda_n}{1-\lambda_n}[x_n - x_{n-1}].$$

7.3 Aitken algorithm

Based on the Aitken extrapolation formula, the following improvement to fixed point iteration can be proposed, known as **Aitken's Algorithm**:

Aitken's Algorithm

Inputs: function g, initial guess x_0, tolerance ε.

1. Start with initial guess x_0.
2. Compute $x_1 = g(x_0)$ and $x_2 = g(x_1)$.
3. Compute Aitken's extrapolation of x_0, x_1 and x_2:
$$\lambda_2 = \frac{x_2 - x_1}{x_1 - x_0},$$
$$x_3 = x_2 + \frac{\lambda_2}{1+\lambda_2}[x_2 - x_1].$$
4. Compute $x_4 = g(x_3)$ and $x_5 = g(x_4)$.
5. Compute again x_6 as Aitken's extrapolation of x_3, x_4 and x_5.
6. And so on, until stopping criterion for fixed point iteration method is satisfied.

Example 9 *Consider again the fixed point iterations from Example 8.*

$$x_n = 2\pi + \sin x_{n-1}, \quad n = 0, 1, 2, \ldots$$

with initial guess $x_0 = 6$. Using the Aitken's Algorithm we will get

$$\alpha - x_3 = 7.98e-04, \qquad \alpha - x_6 = 2.27e-06.$$

On the other hand, using original fixed point iterations we got (see Table 5)

$$\alpha - x_3 = 1.38e-02, \qquad \alpha - x_6 = 1.23e-02.$$

Therefore, Aitken's Algorithm based on Aitken's extrapolation formula can greatly accelerate the convergence of linearly convergent fixed point iterations $x_n = g(x_{n-1})$.

8 Multiple roots

In some cases, the numerical methods for root-finding will not converge as predicted by theory. For example, Newton's method, instead of converging quadratically, will converge linearly, even worse than Bisection method (it will happen if the rate is bigger than 0.5). For example, this will happen if the root has a multiplicity greater than 1.

Definition 2 *We say that a root α of f has **multiplicity** $m \in \mathbb{N}$, if the function f can be factored as*

$$f(x) = (x-\alpha)^m h(x),$$

*where h is a continuous function such that $h(\alpha) \neq 0$. Roots of multiplicity 1 are called **simple** roots.*

Example 10 *Consider polynomial*

$$f(x) = (x+2.8)^2(x-1.1)^3(x-7).$$

This polynomial has root -2.8 of multiplicity 2, root 1.1 of multiplicity 3 and a simple root 7.

Example 11 *Let $f(x) = e^{x^2} - 1$. Obviously $\alpha = 0$ is a zero of this function.*
But it is not that obvious that root $\alpha = 0$ has multiplicity 2. Rewrite function f as

$$f(x) = x^2 \cdot \frac{e^{x^2}-1}{x^2} = x^2 h(x)$$

and let's show that $h(0) \neq 0$. Indeed,

$$\lim_{x\to 0} h(x) = \lim_{x\to 0} \frac{e^{x^2}-1}{x^2} = 1.$$

Therefore, root $\alpha = 0$ is a root of multiplicity 2.

For simple roots, the root-finding numerical methods considered up to this moment will have a behavior as discussed. Let's consider now the case of multiple roots.

Let α be a root of multiplicity m of function f. For simplicity, suppose it has multiplicity 3. Therefore,

$$f(x) = (x-\alpha)^3 h(x) \quad \text{and} \quad h(\alpha) \neq 0.$$

Compute derivative of f,

$$\begin{aligned} f'(x) &= 3(x-\alpha)^2 h(x) + (x-\alpha)^3 h'(x) \\ &= (x-\alpha)^2 \left[3h(x) + (x-\alpha)h'(x)\right] \\ &= (x-\alpha)^2 q(x). \end{aligned}$$

Substitute α in the above expression to get

$$\begin{aligned} f'(\alpha) &= 0, \\ q(\alpha) &= 3h(\alpha) \neq 0 \end{aligned}$$

and therefore, we conclude that α is a root of f' of multiplicity 2.

Compute the second derivative of f, to get

$$\begin{aligned} f''(x) &= \Big((x-\alpha)^2 q(x)\Big)', \\ &= (x-\alpha) r(x), \end{aligned}$$

where $r(\alpha \neq 0)$ and thus, we conclude that α is a simple root of f''.

Differentiating a third time, we can obtain that $f'''(\alpha) \neq 0$.

As conclusion, if we have for a root of function f of multiplicity 3, then

$$\begin{aligned} f(\alpha) &= 0, \\ f'(\alpha) &= 0, \\ f''(\alpha) &= 0, \\ f'''(\alpha) &\neq 0. \end{aligned}$$

We can prove the following generalizing theorem:

Theorem 3 *Number α is a root of function f of multiplicity m* ***if and only if***

$$\begin{aligned} & f(\alpha) = f'(\alpha) = f''(\alpha) = \ldots = f^{(m-1)}(\alpha) = 0, \\ & f^{(m)}(\alpha) \neq 0. \end{aligned}$$

This Theorem can be used as an alternative definition of a root with multiplicity m.

8.1 Difficulties in multiple roots calculation

In the case of multiple roots ($m > 1$), we distinguish two main difficulties in applying numerical methods for root-finding:

1. Newton's and Secant methods will converge more slowly than for the case of a simple root.
2. Due to the noise in function evaluation there will be a large interval of uncertainty for the location of a multiple root.

We know that Newton's method can be viewed as a fixed point iteration method:

$$x_n = g(x_{n-1}) \quad \text{with} \quad g(x) = x - \frac{f(x)}{f'(x)}.$$

On the other hand, since α is a root of multiplicity m, we have $f(x) = (x-\alpha)^m h(x)$. Substitute this expression in the formula for g to get

$$\begin{aligned} g(x) &= x - \frac{(x-\alpha)^m h(x)}{m(x-\alpha)^{m-1} h(x) + (x-\alpha)^m h'(x)} \\ &= x - \frac{(x-\alpha) h(x)}{m h(x) + (x-\alpha) h'(x)}. \end{aligned}$$

Differentiate $g(x)$ and evaluate it at α to obtain

$$g'(\alpha) = \frac{m-1}{m},$$

where m is the multiplicity of α.

Observe that in this case $\lambda = g'(\alpha) \neq 0$ and therefore Newton's method will be only linear convergent:

$$\alpha - x_n \approx \lambda(\alpha - x_{n-1})$$

with linear rate of convergence $\lambda = \dfrac{m-1}{m}$. For example, if root α has multiplicity 3, then Newton's method will converge linearly with rate $\frac{2}{3} \approx 0.67$, worse than Bisection method (which has rate 0.5). Same result can be proven for secant method as well.

Is it possible to accelerate the convergence of Newton's and secant method in case of multiple roots?

For example, to ensure a quadratic convergence as in the case of simple roots?

Let α be a root of f of multiplicity $m > 1$. Then, there are two possibilities:

1. Instead of solving $f(x) = 0$, solve $f^{(m-1)}(x) = 0$ for which α will be a simple root (of multiplicity 1). And consequently, Newton's method will converge quadratically as expected.
2. Consider the fixed iteration method
$$x_n = g(x_{n-1}) \quad \text{with} \quad g(x) = x - m\frac{f(x)}{f'(x)}$$
for which $g'(\alpha) = 0$ and the iteration method will converge quadratically.

Both possibilities presume that multiplicity m is known in advance. But how to find out what is the multipliccity of the root that we need to approximate?

Recall that λ_n (that are computable numbers) converge to $\lambda = \frac{m-1}{m}$. Therefore, multiplicity m can be guessed from numerical computations. For example, if λ_n converge to $0.67 \approx \frac{2}{3}$, one can conclude that probably the multiplicity is 3. If $\lambda_n \to 0.75 \approx 34$, then probably $m = 4$. And so on.

9 Noise in function evaluation for multiple roots

Recall that in evaluating a function, due to rounding/chopping errors, its graph is rather a band with random noise rather than a smooth curve as depicted in figure below.

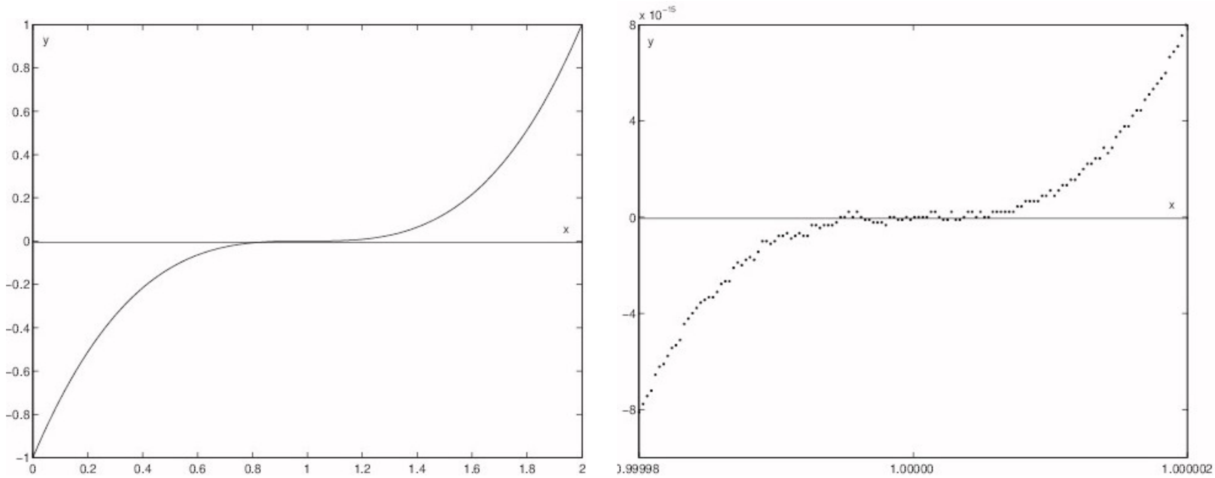

Figure 8: Noise in function evaluation

In the case of multiple roots, this will affect the accuracy of computation in a greater manner than in the case of simple roots because the graph of the function in case of a multiple root will be tangent to x-axis. This is shown in Figure 9, where two functions with the same level of uncertainty in their values have different uncertainties in the location of their roots.

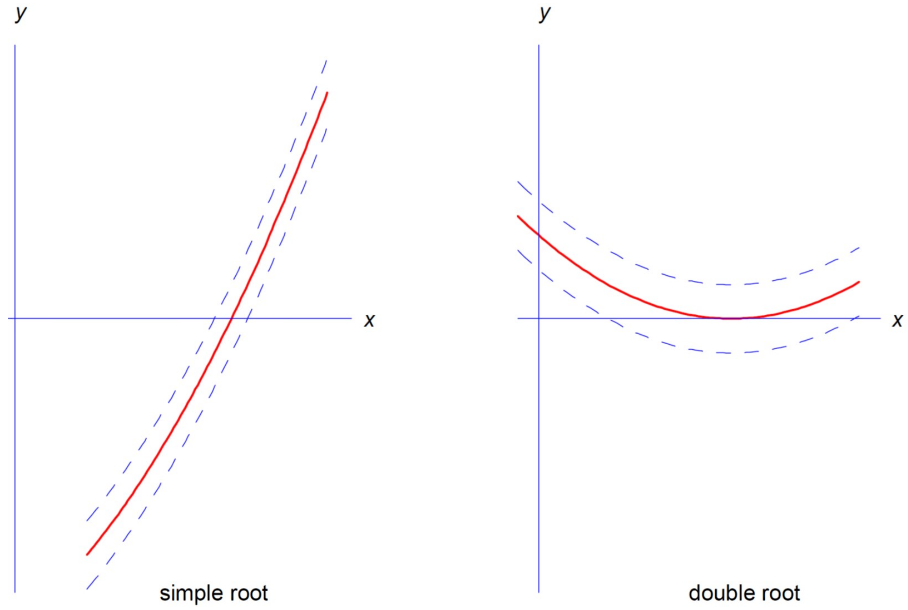

Figure 9: Intervals of uncertainty for simple and multiple roots

Example 12 *Consider equation*

$$f(x) = 2.7951 - 8.954x + 10.56x^2 - 5.4x^3 + x^4 = 0$$

and let α be its root on interval $[0, 2]$. Apply Newton's method with initial guess $x_0 = 0.8$. Numerical results are presented below.

n	x_n	$f(x_n)$	$x_n - x_{n-1}$	λ_n
0	0.800000	0.03510		
1	0.892857	0.01073	0.092857	
2	0.958176	0.00325	0.065319	0.7034
3	1.00344	0.00099	0.045264	0.6930
4	1.03486	0.00029	0.03142	0.6942
5	1.05581	0.00009	0.02095	0.6668
6	1.07028	0.00003	0.01447	0.6907
7	1.08092	0.0	0.01064	0.7353

Table 6: S

Observe that convergence is quite slow. Numerical results are in accordance with linear convergence, since $\lambda_n = \frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}}$ are in interval $[0.65, 0.75]$ and thus $\lambda = g'(\alpha) \neq 0$:

$$\alpha - x_n \approx \lambda(\alpha - x_{n-1}).$$

Also, observe that the convergence is even slower than bisection method, since for bisection method the linear rate is 0.5, whilst in this case the rate is approximately 0.7. The explanation for this behavior is probably the multiplicity $m > 1$ of the root. If this is the case, then since $\lambda \approx \frac{m-1}{m}$, we can make a reasonable guess from numerical results that $m = 3$. Next apply Newton's method to the second derivative equation $f''(x) = 21.12 - 32.4x + 12x^2 = 0$ with initial guess $x_0 = 1$ and it will lead to a much faster convergence to $\alpha = 1.1$.

10 Stability of the roots

Consider polynomial

$$\begin{aligned} f(x) &= x^7 - 28x^6 + 322x^5 - 1960x^4 + 6769x^3 - 13132x^2 + 13068x - 5040 \\ &= (x-1)(x-2)(x-3)(x-4)(x-5)(x-6)(x-7). \end{aligned}$$

This polynomial has exactly 7 simple roots: 1, 2, 3, 4, 5, 6, 7.
Now, consider **the perturbed** polynomial:

$$F(x) = x^7 - 28.002\,x^6 + 322x^5 - 1960x^4 + 6769x^3 - 13132x^2 + 13068x - 5040.$$

As it can be seen, this is a relatively small change in one coefficient with a relative error of

$$\mathrm{Rel}(28) = \frac{28 - 28.002}{28} = -\frac{0.002}{28} \approx 7.14e-05.$$

How this small error in one coefficient changed the roots of $f(x)$? Numerical results are presented in the table below:

Roots of $f(x)$	Roots of $F(x)$	Error
1	1.0000028	$-2.8e-06$
2	1.9989382	$+1.1e-03$
3	3.0331253	$-3.3e-02$
4	3.8195692	$+1.8e-01$
5	$5.4586758 + 0.54012578i$	$-0.46 - 0.54i$
6	$5.4586758 - 0.54012578i$	$-0.46 + 0.54i$
7	7.2330128	$+2.3e-01$

Table 7: Stability of the roots example

It can be seen that a small error (perturbation) (of 10^{-5}) in one coefficient lead to much bigger errors in the roots of perturbed polynomial. Moreover, two of the roots now are complex numbers.

This phenomena (small changes in input lead to big changes in output) is an example of so-called **unstable** or **ill-conditioned** root-finding problem.

Suppose that for a given function f there is perturbed function

$$F(x) = f(x) + \varepsilon g(x).$$

For example, for the previous case we have $g(x) = x^6$ and $\varepsilon = -0.002$.

Let α be a simple root of $f(x)$ and α_ε will be a root of $F(x)$ close to α. It can be shown that

$$\alpha - \alpha_\varepsilon \approx \varepsilon \frac{g(\alpha)}{f'(\alpha)} = \varepsilon M$$

Number M in this case is a condition number. This last approximation allows us to estimate the changes in the root $\alpha - \alpha_\varepsilon$, if a small perturbation(error) ε was introduced. Thus, it can be considered as an indicator of stability.

For our example, $\alpha = 3$ and $\varepsilon = -0.002$:

$$3 - \alpha_\varepsilon \approx 0.002 \cdot \frac{3^6}{48} \approx 0.0304$$

and we see that a small input error of 0.002 is multiplied by a factor of 15. Also, note that the estimated error of 0.0304 is close to the actual error of 0.033.