1. ROOT-FINDING METHODS

Motivation for developing various root-finding methods arises from the requirement to approximate the roots of different functions with wanted accuracy by using as few numerical operations as possible.

The roots of a function $f(x)$, also called the *zeros* of a function, are points on the abscissa where this function has the zero value, i.e. where $f(x) = 0$. To find a particular root, it is required to know

- the first approximation of the root, hopefully close to the root (Fig. 1.1a), or
- ideally, points that bracket the root (Fig. 1.1b).

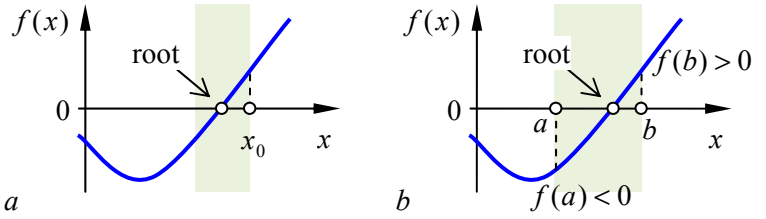

Fig. 1.1. *Basic requirements*: *a – first approximation*, *b – points that bracket the root*

In an interval close to the root and to its first approximation x_0, or in the interval $[a,b]$ that brackets the root, a function needs to be *well-behaved*[1].

[1] The term „well-behaved“ has no fixed formal definition, and it depends on the subject.

Functions that are not well-behaved in an interval of interest have, for example,

- singularity (Fig. 1.2a),
- an infinite number of roots (e.g. $\sin(1/x)$, Fig. 1.2b) , or
- an extreme at the root (very difficult to find, Fig. 1.2c).

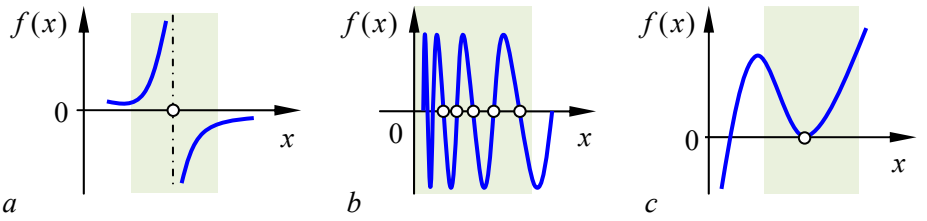

Fig. 1.2. *Functions with*: *a* – singularity, *b* – *infinite number of roots*, *c* – *extreme at the root*

During the history various root-finding methods have been developed. The oldest one is probably the Babylonian method that was used in Mesopotamia about 1800 BC. That has been the most efficient iterative method[1] for finding the square root known until today.

The ancient Babylonians had a nice method of computing square roots by using only simple arithmetic operations. For example, let the rational approximation for the square root of the number a begin with an initial estimate x_0. If the k^{th} estimate x_k is lesser than $\sqrt{a}$, then a/x_k is greater than $\sqrt{a}$, or, vice versa, if x_k is greater than $\sqrt{a}$, then a/x_k is less than $\sqrt{a}$. It follows that the average of these two numbers, x_k and a/x_k, gives a better estimate

$$x_{k+1} = \frac{x_k + a/x_k}{2}. \tag{1.1}$$

This formula[2] is sometimes attributed to Heron of Alexandria (about 10-70 AC) because he described it in his *Metrica*, but it was undoubtedly known to the Babylonians much earlier. For that reason, in this book the method is referred to as the Babylonian/Heron method.

---

[1] The *iteration* means **1.** the act of repeating, a repetition; **2.** (*Mathematic*) a problem-solving or computational method in which a succession of approximations, each building on the one preceding, is used to achieve desired degree of accuracy, or **3.** an instance of the use of this method; **4.** (*Computers*) a repetition of a statement or statements in a program.

[2] The same formula is developed in Chapter 1.5.1, by specialisation of *Newton's* method.

The approximation x_k converges very rapidly to the square root of a, as shown in Example 1.1. When the approximation is close to the solution, the number of accurate digits doubles with each iteration. Such convergence is called the *quadratic convergence*.

Example 1.1. Finding a square root with the Babylonian/Heron method.

Successive approximations in finding the square root of $a = 2$ with the Babylonian/Heron, starting with $x_1 = a$ as the first estimate, are listed in Tab. 1.1. The recursion (1.1) is used.

Tab. 1.1. Successive approximations of the square root of $a = 2$

k	x_k	Accuracy digits
0	2,000 000 000 000 000 000 000 000 000 000	
1	1,500 000 000 000 000 000 000 000 000 000	1,5
2	1,416 666 666 666 666 666 666 666 666 667	3
3	1,414 215 686 274 509 803 921 568 627 451	6
4	1,414 213 562 374 689 910 626 295 578 890	12
5	1,414 213 562 373 095 048 801 689 623 503	24
6	1,414 213 562 373 095 048 801 688 724 210	(48)

The number of accurate digits was doubled with the each iteration so that the method manifests the quadratic convergence.

1.1 CLASSIFICATION OF ROOT-FINDING METHODS

The root-finding methods can be classified either as

- General root-finding methods, or
- Specialised (problem specific) root-finding methods.

General root-finding methods are iterative (use recursive formulas or/and recursive algorithms). Efficiency, applicability and reliability of a particular general root-finding method depend on a function, the roots of which are to be found. For that reason, it is not possible to proclaim the best one.

General root-finding methods can be classified as

- Open root-finding methods,
- Bracketing root-finding methods or
- Multi-dimensional root-finding methods.

The *open* root-finding methods have not a predefined initial interval that brackets the root (Fig. 1.1a). Unlike the open methods, the *bracketing methods* are characterised by the initial interval containing the root (Fig. 1.1b). In each iteration, the interval is divided into two smaller intervals. The one that contains the root is used in the next iteration.

Unlike the general root-finding methods, which are all iterative, the specialised root-finding methods can be either direct or iterative. Specialised iterative methods are usually developed by specialisation of general root finding methods. However, any problem can be solved by several methods, but the efficiency and reliability of the particular specialised method must be separately found out.

The best-known root-finding methods described in this chapter are:

OPEN ROOT-FINDING METHODS (CHAPTER 1.2)

1. *Fixed point iteration*, repetitive approximation of the root of a function $g(x) = x$, by using the recursion formula $x_{n+1} = g(x_n)$, which is expected to converge to the root.
2. *Newton's method*, also known as the *Newton-Rapson* method, consists in successive approximation of the root by using the tangent of a function to find the next approximation. The method is reliable and fast with quadratic convergence if applied to the single root of a continuous function that does not change the sign of the first derivative in the interval between the first estimate and the exact root.
3. *Halley's method*, is similar to Newton's method, but it uses the first two derivatives of a function. The convergence is cubic, but each iteration requires computing a function as well as its first and second derivatives.
4. *Householder's methods*, the class of root-finding algorithms using derivatives up to order $d+1$. If $d=1$, the method is equal to *Newton's method*, whereas if $d=2$, the method is equal to *Halley's method*. The rate of convergence is $d+1$.

5. *Secant method* is similar to Newton's method, but it uses the secant through the last two computed points instead of the tangent in the single point. The convergence is superlinear (with order 1,62).
6. *Steffensen's method* is similar to Newton's method. It also achieves quadratic convergence, but it does not use the derivative. Instead of the first derivative, a *slope function* is used. The main advantage of Steffensen's method is that it has quadratic convergence like Newton's method, but does not require any development of the derivative.
7. *Müller's method*, also known as the *Müller-Traub method*, uses the last three points to determine the coefficients of the parabola $y = ax^2 + b$, and takes the intersection of the parabola and the *x*-axis to be the next approximation. The order of convergence is approximately 1,84.
8. *Inverse quadratic interpolation* constructs an "inverse" parabola through the last three points. Instead of finding the coefficients of the related quadratic equation $x = af^2 + b$, the method uses inverse quadratic interpolation to find the next approximation of the root.
9. *Modified secant method* uses the last three points to construct a rational function and takes the intersection of its graph and the *x*-axis as the next approximation of the root.

BRACKETING ROOT-FINDING METHODS (CHAPTER 1.3)

10. *Bisection method*, repetitive bisection of the interval containing the root, is the most robust and reliable root-finding method with predictable number of numerical operations, applicable to any continuous function when the minimisation of numerical operations is not a main priority.
11. *Dekker's method* is a combination of the bisection method and the linear interpolation or extrapolation. The bisection provides reliability, while the interpolation increases the speed.
12. *Brent's method* is a complex root-finding algorithm that combines the bisection method with the inverse quadratic interpolation. It has not only the reliability of the bisection method but it can be also as quick as some less reliable methods.
13. *False position method*, also known as *Regula falsi method*, consists in repetitive splitting the interval containing the root by using linear interpolation on the actual interval. In comparison with the bisection method, in some cases it is faster, but in some cases it is slower and not so reliable and predictable. The used iteration formula is formally equal

to those used in the secant method, but with different rules for retaining the two points for the next iteration.

14. *Ridder's method* is a modification of the false position method based on the use of exponential function instead of linear interpolation. This method is extraordinary robust like the bisection method. The rate of convergence is 1,41 per each function evaluation.
15. *Modified false-position method* uses three points to construct a rational function at each iteration, and takes the intersection of its graph and the *x*-axis to be the next approximation.
16. *Illinois algorithm* is the improved "modified false position method" that prevents the retention of the old points. This is done by determining the optimal downscaling coefficient. The algorithm is simple and reliable with the rate of convergence 1,44 per each function evaluation.

SPECIALISED ITERATIVE METHODS (CHAPTERS 1.4 TO 1.7)

17. *Newton-Raphson division* is the application of Newton's method to the division (Chapter 1.4).
18. Square-root methods (Chapter 1.5)
    a. *The Babylonian/Heron method*, the oldest and possibly the most efficient iterative method for finding the square root of a real number. It can be developed from Newton's recursion formula.
    b. Methods based on Halley's and Householder's recursion formulas.
19. Inverse square-root methods (Chapter 1.6), the methods for computing the inverse of the square-root based on Newton's, Halley's and Householder's recursion formulas.
20. *N*-th root algorithms (Chapter 1.7), the methods for computing n^{th} root based on Newton's and Halley's iteration formulas.

MULTI-DIMENSIONAL ROOT-FINDING METHODS (CHAPTER 1.8)

21. Multi-dimensional fixed point iteration; the generalisation of the one--dimensional fixed point iteration.
22. Multi-dimensional *Newton's method*, the generalisation of the one-dimensional Newton's method.
23. *The Gauss-Newton method* combines Newton's iteration formula and the Gaussian method of least squares.

1.2 OPEN ROOT-FINDING METHODS

The open root-finding methods are iterative methods. With the exception of the fixed-point iteration, the common property of open methods is that the next guess of the root is computed by extrapolation. The extrapolation can be linear or higher order, depending on the number of sampling points in which a function and possibly its derivatives are calculated (Tab. 1.2).

Tab. 1.2. Open methods classified by order of extrapolation

Extrapolation	Points	Derivatives	Method	Chapter
-	1	(none)	Fixed-point	1.2.1
Linear	1	$f'(x)$	Newton's	1.2.2
	2	(none)	Secant	1.2.5
			Steffenson's	1.2.6
Quadratic	1	$f'(x)$, $f''(x)$	Halley's	1.2.3
	3	(none)	Müller's	1.2.7
			Inverse quadratic	1.2.8
Cubic	1	$f'(x)$, $f''(x)$, $f'''(x)$	Householder's 3rd order	1.2.4
High order	1	$f'(x) \dots f^{(n)}(x)$	Householder's n^{th} order	
Rational function	3	(none)	Modified secant	1.2.9

Open methods have some advantages and disadvantages (Tab. 1.3). Their common disadvantage is that the first guess must be sufficiently close to the root. As the first guess is closer to the root, the convergence will be faster. But even then, the convergence is not always guaranteed. The possibility that the particular method will fail usually depends not only on the choice of the first guess, but also on functions and the multiplicity of their roots. However, in some cases, as in the case of the Babylonian method (Chapter 1.5.1), which can be derived by a specialisation of Newton's method, the convergence is always guaranteed.

Tab. 1.3. Main advantages and disadvantages of the open root-finding methods

Main advantages	Main disadvantages
1. *Fixed point iteration* CHAPTER 1.2.1	
The simplest method.	If convergence exists, it can be slow.
2. *Newton's method* CHAPTER 1.2.2	
Reliable and fast method with the quadratic convergence.	The first derivative of a function must be developed.
3. *Halley's method* CHAPTER 1.2.3	
Reliable and fast method with the cubic convergence.	The first and second derivative of a function must be developed.
4. *Householder's methods* CHAPTER 1.2.4	
Reliable and fast methods with the convergence order $d+1$.	The first d derivatives of a function must be developed.
5. *The Secant method* CHAPTER 1.2.5	
Doesn't require the first derivative. Superlinear convergence (order 1,62).	Requires two points to define secant. Slower convergence.
6. *Steffensen's method* CHAPTER 1.2.6	
Doesn't require the first derivative. Quadratic convergence.	Double function evaluation in each iteration.
7. *Müller's method* CHAPTER 1.2.7	
Doesn't require derivatives. The order of convergence is 1,84.	Computes three points to define the interpolation parabola.
8. *Inverse quadratic interpolation* CHAPTER 1.2.8	
Doesn't require derivatives.	Requires three points for inverse interpolation with the parabola.
9. *Modified secant method* CHAPTER 1.2.9	
Doesn't require derivatives.	Requires three points for the extrapolation with rational function.

1.2.1 Fixed point iteration

The fixed point iteration is probably the simplest root finding method [1]. It consists in repetitive approximation for a root of a function $f(x)$ by using the recursion formula

$$x_{k+1} = g(x_k), \tag{1.3}$$

where auxiliary function

$$g(x) = f(x) + x. \tag{1.2}$$

Two examples of fixed point iteration are given in Fig. 1.3. The root is determined by an intersection of the curve $y = g(x)$ and the line $y = x$. The example *a* illustrates a convergence, while the example *b* illustrates a divergence.

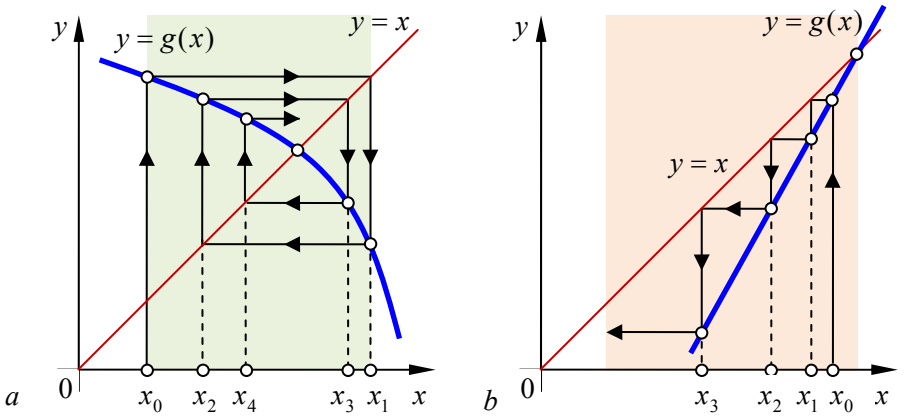

Fig. 1.3. *Fixed point iteration, a – convergent, b – divergent*

The main disadvantage of the method is that the convergence is not always guaranteed. It will be guaranteed only if $|g'(x_k)| < 1$.

Example 1.2. Application of fixed-point iteration to the function

$$f(x) = \ln x + x - 2{,}4. \tag{1.4}$$

The iteration formula is

$$x_{k+1} = 2{,}4 - \ln x_k. \tag{1.5}$$

Starting with $x_0 = 2$, the values in Tab. 1.4 are obtained. In the given example, the iteration is very slow although the first guess is close to the root. The calculated values oscillate around the true value. The accuracy of 10 digits is obtained after 34 steps – that is roughly one bit per step.

Tab. 1.4. Application of fixed-point iteration to the function (1.4)

k	x_k	k	x_k	k	x_k
0	2,000 000 000 00	7	1,804 911 890 15	14	1,807 904 231 22
1	1,706 852 819 44	8	1,809 488 223 73	15	1,807 831 708 89
2	1,865 348 781 67	9	1,806 955 944 02	16	1,807 871 823 73
3	1,776 551 950 09	10	1,808 356 369 42	17	1,807 849 634 50
4	1,825 325 621 01	11	1,807 581 650 48	18	1,807 861 908 25
5	1,798 241 606 41	12	1,808 010 152 82	19	1,807 855 119 13
6	1,813 190 697 89	13	1,807 773 122 55		1,807 857 537

1.2.2 Newton's method

Newton's method[1] is often referred to as one of the best-known methods for successively finding better approximations to the roots of real functions. The algorithm is the first in the class of Householder's methods (Chapter 1.2.4). It often converges quickly, especially if the first iteration begins sufficiently near the root. Unfortunately, when the first iteration begins far from the root, Newton's method can fail to converge.

Newton's method requires a successive calculation of a function $f_k = f(x_k)$ and its first derivative $f'_k = f'(x_k)$ at sampling points x_k. The first iteration begins with the calculation of $f_0 = f(x_0)$ and $f'_0 = f'(x_0)$ in an arbitrary initial value x_0, which is reasonably close to the root.

Starting with an initial value x_0, a tangent through the point $(x_0, f(x_0))$ can be constructed, as it is illustrated in Fig. 1.4. The tangent has the equation

$$y(x) = f'(x_0)(x - x_0) + f(x_0). \tag{1.6}$$

The point $(x_1,0)$ in which the tangent intersects the x-axis, i.e. such a value x_1 that $y(x_1) = 0$, can be found by solving the following equation for x_1:

$$f'(x_0)(x_1 - x_0) + f(x_0) = 0. \tag{1.7}$$

[1] Newton's method is also known as the Newton-Rapson method, named after Isac Newton and Joseph Rapson