Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Afrikaans
العربية
Български
Català
Čeština
Dansk
Deutsch
Ελληνικά
Español
فارسی
Français
한국어
हिन्दी
Bahasa Indonesia
Italiano
עברית
Magyar
Nederlands
日本語
Norsk bokmål
Polski
Português
Română
Русский
Simple English
Slovenščina
Suomi
Svenska
Українська
中文

Newton's method

From Wikipedia, the free encyclopedia

This article is about Newton's method for finding roots. For Newton's method for finding minima, see Newton's method in optimization.

This article includes a list of references, but **its sources remain unclear** because it has **insufficient inline citations**. Please help to improve this article by introducing more precise citations. *(February 2014)*

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. *(November 2013)*

In numerical analysis, **Newton's method** (also known as the **Newton–Raphson method**), named after Isaac Newton and Joseph Raphson, is a method for finding successively better approximations to the roots (or zeroes) of a real-valued function.

$$x : f(x) = 0\,.$$

The Newton–Raphson method in one variable is implemented as follows:

Given a function f defined over the reals x, and its derivative f', we begin with a first guess x_0 for a root of the function f. Provided the function satisfies all the assumptions made in the derivation of the formula, a better approximation x_1 is

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

Geometrically, $(x_1, 0)$ is the intersection with the x-axis of the tangent to the graph of f at $(x_0, f(x_0))$.

The process is repeated as

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

until a sufficiently accurate value is reached.

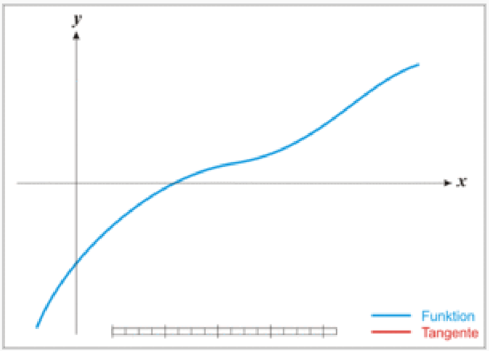
This algorithm is first in the class of Householder's methods, succeeded by Halley's method. The method can also be extended to complex functions and to systems of equations.

Contents [hide]

Edit links

Description [edit]

The idea of the method is as follows: one starts with an initial guess which is reasonably close to the true root, then the function is approximated by its tangent line (which can be computed using the tools of calculus), and one computes the *x*-intercept of this tangent line (which is easily done with elementary algebra). This *x*-intercept will typically be a better approximation to the function's root than the original guess, and the method can be iterated.



The function f is shown in blue and the tangent line is in red. We see that x_{n+1} is a better approximation than x_n for the root *x* of the function *f*.

Suppose $f : [a, b] \to \mathbf{R}$ is a differentiable function defined on the interval [*a*, *b*] with values in the real numbers **R**. The formula for converging on the root can be easily derived. Suppose we have some current approximation x_n. Then we can derive the formula for a better approximation, x_{n+1} by referring to the diagram on the right. The equation of the tangent line to the curve $y = f(x)$ at the point $x=x_n$ is

$$y = f'(x_n)\,(x - x_n) + f(x_n),$$

where, f' denotes the derivative of the function f.

The *x*-intercept of this line (the value of *x* such that *y*=0) is then used as the next approximation to the root, x_{n+1}. In other words, setting *y* to zero and *x* to x_{n+1} gives

$$0 = f'(x_n)\,(x_{n+1} - x_n) + f(x_n).$$

Solving for x_{n+1} gives

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

We start the process off with some arbitrary initial value x_0. (The closer to the zero, the better. But, in the absence of any intuition about where the zero might lie, a "guess and check" method might narrow the possibilities to a reasonably small interval by appealing to the intermediate value theorem.) The method will usually converge, provided this initial guess is close enough to the unknown zero, and that $f'(x_0) \neq 0$. Furthermore, for a zero of multiplicity 1, the convergence is at least quadratic (see rate of convergence) in a neighbourhood of the zero, which intuitively means that the number of correct digits roughly at least doubles in every step. More details can be found in the analysis section below.

The Householder's methods are similar but have higher order for even faster convergence. However, the extra computations required for each step can slow down the overall performance relative to Newton's method, particularly if *f* or its derivatives are computationally expensive to evaluate.

History [edit]

The name "Newton's method" is derived from Isaac Newton's description of a special case of the method in *De analysi per aequationes numero terminorum infinitas* (written in 1669, published in 1711 by William Jones) and in *De metodis fluxionum et serierum infinitarum* (written in 1671, translated and published as *Method of Fluxions* in 1736 by John Colson). However, his method differs substantially from the modern method given above: Newton applies the method only to polynomials. He does not compute the successive approximations x_n, but computes a sequence of polynomials, and only at the end arrives at an approximation for the root *x*. Finally, Newton views the method as purely algebraic and makes no mention of the connection with calculus. Newton may have derived his method from a similar but less precise method by Vieta. The essence of Vieta's method can be found in the work of the Persian mathematician Sharaf al-Din al-Tusi, while his successor Jamshīd al-Kāshī used a form of Newton's method to solve $x^P - N = 0$ to find roots of *N* (Ypma 1995). A special case of Newton's method for calculating square roots was known much earlier and is often called the Babylonian method.

Newton's method was used by 17th-century Japanese mathematician Seki Kōwa to solve single-variable equations, though the connection with calculus was missing.

Newton's method was first published in 1685 in *A Treatise of Algebra both Historical and Practical* by John Wallis. In 1690, Joseph Raphson published a simplified description in *Analysis aequationum universalis*. Raphson again viewed Newton's method purely as an algebraic method and restricted its use to polynomials, but he describes the method in terms of the successive approximations x_n instead of the more complicated sequence of polynomials used by Newton. Finally, in 1740, Thomas Simpson described Newton's method as an iterative method for solving general nonlinear equations using calculus, essentially giving the description above. In the same publication, Simpson also gives the generalization to systems of two equations and notes that Newton's method can be used for solving optimization problems by setting the gradient to zero.

Arthur Cayley in 1879 in *The Newton-Fourier imaginary problem* was the first to notice the difficulties in generalizing Newton's method to complex roots of polynomials with degree greater than 2 and complex initial values. This opened the way to the study of the theory of iterations of rational functions.

Practical considerations [edit]

Newton's method is an extremely powerful technique—in general the convergence is quadratic: as the method converges on the root, the difference between the root and the approximation is squared (the number of accurate digits roughly doubles) at each step. However, there are some difficulties with the method.

Difficulty in calculating derivative of a function [edit]

Newton's method requires that the derivative be calculated directly. An analytical expression for the derivative may not be easily obtainable and could be expensive to evaluate. In these situations, it may be appropriate to approximate the derivative by using the slope of a line through two nearby points on the function. Using this approximation would result in something like the secant method whose convergence is slower than that of Newton's method.

Failure of the method to converge to the root [edit]

It is important to review the proof of quadratic convergence of Newton's Method before implementing it. Specifically, one should review the assumptions made in the proof. For situations where the method fails to converge, it is because the assumptions made in this proof are not met.

Overshoot [edit]

If the first derivative is not well behaved in the neighborhood of a particular root, the method may overshoot, and diverge from that root. An example of a function with one root, for which the derivative is not well behaved in the neighborhood of the root, is

$$f(x) = |x|^a, \quad 0 < a < \tfrac{1}{2}$$

for which the root will be overshot and the sequence of x will diverge. For $a = 1/2$, the root will still be overshot, but the sequence will oscillate between two values. For $1/2 < a < 1$, the root will still be overshot but the sequence will converge, and for $a \geq 1$ the root will not be overshot at all.

In some cases, Newton's method can be stabilized by using successive over-relaxation, or the speed of convergence can be increased by using the same method.

Stationary point [edit]

If a stationary point of the function is encountered, the derivative is zero and the method will terminate due to division by zero.

Poor initial estimate [edit]

A large error in the initial estimate can contribute to non-convergence of the algorithm.

Mitigation of non-convergence [edit]

In a robust implementation of Newton's method, it is common to place limits on the number of iterations, bound the solution to an interval known to contain the root, and combine the method with a more robust root finding method.

Slow convergence for roots of multiplicity > 1 [edit]

If the root being sought has multiplicity greater than one, the convergence rate is merely linear (errors reduced by a constant factor at each step) unless special steps are taken. When there are two or more roots that are close together then it may take many iterations before the iterates get close enough to one of them for the quadratic convergence to be apparent. However, if the multiplicity m of the root is known, one can use the following modified algorithm that preserves the quadratic convergence rate:

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

This is equivalent to using successive over-relaxation. On the other hand, if the multiplicity m of the root is not known, it is possible to estimate m after carrying out one or two iterations, and then use that value to increase the rate of convergence.

Analysis [edit]

This section **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. *(February 2014)*

Suppose that the function f has a zero at α, i.e., f(α) = 0, and f is differentiable in a neighborhood of α.

If f is continuously differentiable and its derivative is nonzero at α, then there exists a neighborhood of α such that for all starting values x_0 in that neighborhood, the sequence $\{x_n\}$ will converge to α.[2]

If the function is continuously differentiable and its derivative is not 0 at α and it has a second derivative at α then the convergence is quadratic or faster. If the second derivative is not 0 at α then the convergence is merely quadratic. If the third derivative exists and is bounded in a neighborhood of α, then:

$$\Delta x_{i+1} = \frac{f''(\alpha)}{2f'(\alpha)}(\Delta x_i)^2 + O[\Delta x_i]^3\,,$$

where $\Delta x_i \triangleq x_i - \alpha\,.$

If the derivative is 0 at α, then the convergence is usually only linear. Specifically, if f is twice continuously differentiable, $f\,'(\alpha) = 0$ and $f\,''(\alpha) \neq 0$, then there exists a neighborhood of α such that for all starting values x_0 in that neighborhood, the sequence of iterates converges linearly, with rate $\log_{10} 2$ (Süli & Mayers, Exercise 1.6). Alternatively if $f\,'(\alpha) = 0$ and $f\,'(x) \neq 0$ for $x \neq \alpha$, x in a neighborhood U of α, α being a zero of multiplicity r, and if $f \in C^r(U)$ then there exists a neighborhood of α such that for all starting values x_0 in that neighborhood, the sequence of iterates converges linearly.

However, even linear convergence is not guaranteed in pathological situations.

In practice these results are local, and the neighborhood of convergence is not known in advance. But there are also some results on global convergence: for instance, given a right neighborhood U_+ of α, if f is twice differentiable in U_+ and if $f' \neq 0$, $f \cdot f'' > 0$ in U_+, then, for each x_0 in U_+ the sequence x_k is monotonically decreasing to α.

Proof of quadratic convergence for Newton's iterative method [edit]

According to Taylor's theorem, any function $f(x)$ which has a continuous second derivative can be represented by an expansion about a point that is close to a root of f(x). Suppose this root is α . Then the expansion of f(α) about x_n is:

$$f(\alpha) = f(x_n) + f'(x_n)(\alpha - x_n) + R_1 \qquad (1)$$

where the Lagrange form of the Taylor series expansion remainder is

$$R_1 = \frac{1}{2!} f''(\xi_n)(\alpha - x_n)^2,$$

where ξ_n is in between x_n and α .

Since α is the root, (1) becomes:

$$0 = f(\alpha) = f(x_n) + f'(x_n)(\alpha - x_n) + \frac{1}{2} f''(\xi_n)(\alpha - x_n)^2 \qquad (2)$$

Dividing equation (2) by $f'(x_n)$ and rearranging gives

$$\frac{f(x_n)}{f'(x_n)} + (\alpha - x_n) = \frac{-f''(\xi_n)}{2f'(x_n)} (\alpha - x_n)^2 \qquad (3)$$

Remembering that x_{n+1} is defined by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\,, \qquad (4)$$

one finds that

$$\underbrace{\alpha - x_{n+1}}_{\epsilon_{n+1}} = \frac{-f''(\xi_n)}{2f'(x_n)} (\underbrace{\alpha - x_n}_{\epsilon_n})^2\,.$$

That is,

$$\epsilon_{n+1} = \frac{-f''(\xi_n)}{2f'(x_n)} \,{\epsilon_n}^2\,. \qquad (5)$$

Taking absolute value of both sides gives

$$\left|\epsilon_{n+1}\right| = \frac{\left|f''(\xi_n)\right|}{2\left|f'(x_n)\right|} \,{\epsilon_n}^2 \qquad (6)$$

Equation (6) shows that the rate of convergence is quadratic if the following conditions are satisfied:

1. $f'(x) \neq 0; \forall x \in I,$ where I is the interval $[\alpha - r, \alpha + r]$ for some $r \geq \left|(\alpha - x_0)\right|;$
2. $f''(x)$ is continuous, $\forall x \in I;$
3. x_0 *sufficiently* close to the root α

The term *sufficiently* close in this context means the following:

(a) Taylor approximation is accurate enough such that we can ignore higher order terms,

(b) $\frac{1}{2}\left|\frac{f''(x_n)}{f'(x_n)}\right| < C\left|\frac{f''(\alpha)}{f'(\alpha)}\right|,$ for some $C < \infty,$

(c) $C\left|\frac{f''(\alpha)}{f'(\alpha)}\right|\epsilon_n < 1,$ for $n \in \mathbb{Z}^+ \cup \{0\}$ and C satisfying condition (b) .

Finally, (6) can be expressed in the following way:

$$\left|\epsilon_{n+1}\right| \leq M {\epsilon_n}^2$$

where M is the supremum of the variable coefficient of ${\epsilon_n}^2$ on the interval I defined in the condition 1, that is:

$$M = \sup_{x \in I} \frac{1}{2}\left|\frac{f''(x)}{f'(x)}\right|.$$

The initial point x_0 has to be chosen such that conditions 1 through 3 are satisfied, where the third condition requires that $M\left|\epsilon_0\right| < 1.$

Basins of attraction [edit]

The basins of attraction—the regions of the real number line such that within each region iteration from any point leads to one particular root—can be infinite in number and arbitrarily small. For example,[3] for the function $f(x) = x^3 - 2x^2 - 11x + 12$, the following initial conditions are in successive basins of attraction:

2.35287527 converges to 4;

2.35284172 converges to −3;

2.35283735 converges to 4;

2.352836327 converges to −3;

2.352836323 converges to 1.

Failure analysis [edit]

Newton's method is only guaranteed to converge if certain conditions are satisfied. If the assumptions made in the proof of quadratic convergence are met, the method will converge. For the following subsections, failure of the method to converge indicates that the assumptions made in the proof were not met.

Bad starting points [edit]

In some cases the conditions on the function that are necessary for convergence are satisfied, but the point chosen as the initial point is not in the interval where the method converges. This can happen, for example, if the function whose root is sought approaches zero asymptotically as *x* goes to ∞ or $-\infty$. In such cases a different method, such as bisection, should be used to obtain a better estimate for the zero to use as an initial point.

Iteration point is stationary [edit]

Consider the function:

$$f(x) = 1 - x^2.$$

It has a maximum at $x = 0$ and solutions of $f(x) = 0$ at $x = \pm 1$. If we start iterating from the stationary point $x_0 = 0$ (where the derivative is zero), x_1 will be undefined, since the tangent at (0,1) is parallel to the *x*-axis:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{1}{0}.$$

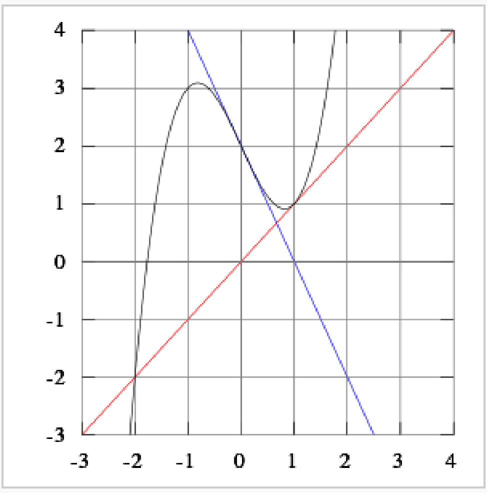
The same issue occurs if, instead of the starting point, any iteration point is stationary. Even if the derivative is small but not zero, the next iteration will be a far worse approximation.

Starting point enters a cycle [edit]

For some functions, some starting points may enter an infinite cycle, preventing convergence. Let

$$f(x) = x^3 - 2x + 2$$

and take 0 as the starting point. The first iteration produces 1 and the second iteration returns to 0 so the sequence will alternate between the two without converging to a root. In fact, this 2-cycle is stable: there are neighborhoods around 0 and around 1 from which all points iterate asymptotically to the 2-cycle (and hence not to the root of the function). In general, the behavior of the sequence can be very complex (see Newton fractal).The real solution of this equation is -1,76929235...



The tangent lines of $x^3 - 2x + 2$ at 0 and 1 intersect the *x*-axis at 1 and 0 respectively, illustrating why Newton's method oscillates between these values for some starting points.

Derivative issues [edit]

If the function is not continuously differentiable in a neighborhood of the root then it is possible that Newton's method will always diverge and fail, unless the solution is guessed on the first try.

Derivative does not exist at root [edit]

A simple example of a function where Newton's method diverges is the cube root, which is continuous and infinitely differentiable, except for $x = 0$, where its derivative is undefined (this, however, does not affect the algorithm, since it will never require the derivative if the solution is already found):

$$f(x) = \sqrt[3]{x}.$$

For any iteration point x_n, the next iteration point will be:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{{x_n}^{\frac{1}{3}}}{\frac{1}{3}\,{x_n}^{\frac{1}{3}-1}} = x_n - 3\,x_n = -2\,x_n.$$

The algorithm overshoots the solution and lands on the other side of the *y*-axis, farther away than it initially was; applying Newton's method actually doubles the distances from the solution at each iteration.

In fact, the iterations diverge to infinity for every $f(x) = |x|^\alpha$, where $0 < \alpha < \frac{1}{2}$. In the limiting case of $\alpha = \frac{1}{2}$ (square root), the iterations will alternate indefinitely between points x_0 and $-x_0$, so they do not converge in this case either.

Discontinuous derivative [edit]

If the derivative is not continuous at the root, then convergence may fail to occur in any neighborhood of the root. Consider the function

$$f(x) = \begin{cases} 0 & \text{if } x = 0,\\ x + x^2\sin\left(\frac{2}{x}\right) & \text{if } x \neq 0.\end{cases}$$

Its derivative is:

$$f'(x) = \begin{cases} 1 & \text{if } x = 0,\\ 1 + 2\,x\,\sin\left(\frac{2}{x}\right) - 2\,\cos\left(\frac{2}{x}\right) & \text{if } x \neq 0.\end{cases}$$

Within any neighborhood of the root, this derivative keeps changing sign as *x* approaches 0 from the right (or from the left) while $f(x) \geq x - x^2 > 0$ for $0 < x < 1$.

So $f(x)/f'(x)$ is unbounded near the root, and Newton's method will diverge almost everywhere in any neighborhood of it, even though:

- the function is differentiable (and thus continuous) everywhere;
- the derivative at the root is nonzero;
- *f* is infinitely differentiable except at the root; and
- the derivative is bounded in a neighborhood of the root (unlike $f(x)/f'(x)$).

Non-quadratic convergence [edit]

In some cases the iterates converge but do not converge as quickly as promised. In these cases simpler methods converge just as quickly as Newton's method.

Zero derivative [edit]

If the first derivative is zero at the root, then convergence will not be quadratic. Let

$$f(x) = x^2$$

then $f'(x) = 2x$ and consequently $x - f(x)/f'(x) = x/2$. So convergence is not quadratic, even though the function is infinitely differentiable everywhere.

Similar problems occur even when the root is only "nearly" double. For example, let

$$f(x) = x^2(x-1000)+1.$$

Then the first few iterates starting at $x_0 = 1$ are 1, 0.500250376, 0.251062828, 0.127507934, 0.067671976, 0.041224176, 0.032741218, 0.031642362; it takes six iterations to reach a point where the convergence appears to be quadratic.

No second derivative [edit]

If there is no second derivative at the root, then convergence may fail to be quadratic. Let

$$f(x) = x + x^{\frac{4}{3}}.$$

Then

$$f'(x) = 1 + \frac{4}{3}x^{\frac{1}{3}}.$$

And

$$f''(x) = \frac{4}{9}x^{-\frac{2}{3}}$$

except when $x = 0$ where it is undefined. Given x_n,

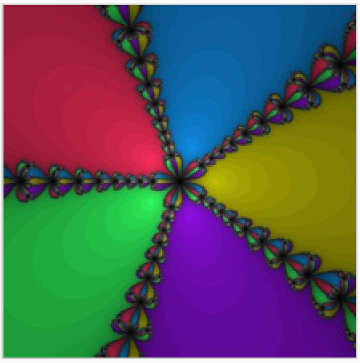
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = \frac{\frac{1}{3}{x_n}^{\frac{4}{3}}}{(1+\frac{4}{3}{x_n}^{\frac{1}{3}})}$$

which has approximately 4/3 times as many bits of precision as x_n has. This is less than the 2 times as many which would be required for quadratic convergence. So the convergence of Newton's method (in this case) is not quadratic, even though: the function is continuously differentiable everywhere; the derivative is not zero at the root; and f is infinitely differentiable except at the desired root.

Generalizations [edit]

Complex functions [edit]

Main article: Newton fractal


Basins of attraction for $x^5 - 1 = 0$; darker means more iterations to converge.

When dealing with complex functions, Newton's method can be directly applied to find their zeroes. Each zero has a basin of attraction in the complex plane, the set of all starting values that cause the method to converge to that particular zero. These sets can be mapped as in the image shown. For many complex functions, the boundaries of the basins of attraction are fractals.

In some cases there are regions in the complex plane which are not in any of these basins of attraction, meaning the iterates do not converge. For example,[4] if one uses a real initial condition to seek a root of x^2+1, all subsequent iterates will be real numbers and so the iterations cannot converge to either root, since both roots are non-real. In this case almost all real initial conditions lead to chaotic behavior, while some initial conditions iterate either to infinity or to repeating cycles of any finite length.

Curt McMullen has shown that for any possible purely iterative algorithm similar to Newton's Method, the algorithm will diverge on some open regions of the complex plane when applied to some polynomial of degree d ≥ 4. However, McMullen gave a generally convergent algorithm for polynomials of degree d = 3.[5]

Nonlinear systems of equations [edit]

k variables, k functions [edit]

One may also use Newton's method to solve systems of *k* (non-linear) equations, which amounts to finding the zeroes of continuously differentiable functions $F : \mathbf{R}^k \to \mathbf{R}^k$. In the formulation given above, one then has to left multiply with the inverse of the *k*-by-*k* Jacobian matrix $J_F(x_n)$ instead of dividing by $f'(x_n)$.

Rather than actually computing the inverse of this matrix, one can save time by solving the system of linear equations

$$J_F(x_n)(x_{n+1} - x_n) = -F(x_n)$$

for the unknown $x_{n+1} - x_n$.

k variables, m equations, with m > k [edit]

The k-dimensional Newton's method can be used to solve systems of >*k* (non-linear) equations as well if the algorithm uses the generalized inverse of the non-square Jacobian matrix $J^+ = ((J^TJ)^{-1})J^T$ instead of the inverse of J. If the nonlinear system has no solution, the method attempts to find a solution in the non-linear least squares sense. See Gauss–Newton algorithm for more information.

Nonlinear equations in a Banach space [edit]

Another generalization is Newton's method to find a root of a functional *F* defined in a Banach space. In this case the formulation is

$$X_{n+1} = X_n - [F'(X_n)]^{-1}F(X_n),$$

where $F'(X_n)$ is the Fréchet derivative computed at X_n. One needs the Fréchet derivative to be boundedly invertible at each X_n in order for the method to be applicable. A condition for existence of and convergence to a root is given by the Newton–Kantorovich theorem.

Nonlinear equations over *p*-adic numbers [edit]

In *p*-adic analysis, the standard method to show a polynomial equation in one variable has a *p*-adic root is Hensel's lemma, which uses the recursion from Newton's method on the *p*-adic numbers. Because of the more stable behavior of addition and multiplication in the *p*-adic numbers compared to the real numbers (specifically, the unit ball in the *p*-adics is a ring), convergence in Hensel's lemma can be guaranteed under much simpler hypotheses than in the classical Newton's method on the real line.

Newton-Fourier method [edit]

The Newton-Fourier method is Joseph Fourier's extension of Newton's method to provide bounds on the absolute error of the root approximation, while still providing quadratic convergence.

Assume that $f(x)$ is twice continuously differentiable on $[a, b]$ and that f contains a root in this interval. Assume that $f'(x)f''(x) \neq 0$ on this interval (this is the case for instance if $f(a) < 0$, $f(b) > 0$, and $f'(x) > 0$, and $f''(x) > 0$ on this interval). This guarantees that there is a unique root on this interval, call it α. If it is concave down instead of concave up then replace $f(x)$ by $-f(x)$ since they have the same roots.

Let $x_0 = b$ be the right endpoint of the interval and let $z_0 = a$ be the left endpoint of the interval. Given x_n, define $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$, which is just Newton's method as before. Then define $z_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)}$ and note that the denominator has $f'(x_n)$ and not $f'(z_n)$. The iterates x_n will be strictly decreasing to the root while the iterates z_n will be strictly increasing to the root. Also, $\lim_{n\to\infty} \frac{x_{n+1} - z_{n+1}}{(x_n - z_n)^2} = \frac{f''(\alpha)}{2f'(\alpha)}$ so that distance between x_n and z_n decreases quadratically.

Quasi-Newton methods [edit]

When the Jacobian is unavailable or too expensive to compute at every iteration, a Quasi-Newton method can be used.

Applications [edit]

Minimization and maximization problems [edit]

Main article: Newton's method in optimization

Newton's method can be used to find a minimum or maximum of a function. The derivative is zero at a minimum or maximum, so minima and maxima can be found by applying Newton's method to the derivative. The iteration becomes:

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

Multiplicative inverses of numbers and power series [edit]

An important application is Newton–Raphson division, which can be used to quickly find the reciprocal of a number, using only multiplication and subtraction.

Finding the reciprocal of *a* amounts to finding the root of the function

$$f(x) = a - \frac{1}{x}$$

Newton's iteration is

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \\ &= x_n - \frac{a - \frac{1}{x_n}}{\frac{1}{x_n^2}} \\ &= x_n(2 - a x_n) \end{aligned}$$

Therefore, Newton's iteration needs only two multiplications and one subtraction.

This method is also very efficient to compute the multiplicative inverse of a power series.

Solving transcendental equations [edit]

Many transcendental equations can be solved using Newton's method. Given the equation

$$g(x) = h(x),$$

with *g(x)* and/or *h(x)* a transcendental function, one writes

$$f(x) = g(x) - h(x).$$

The values of *x* that solves the original equation are then the roots of *f(x)*, which may be found via Newton's method.

Examples [edit]

Square root of a number [edit]

Consider the problem of finding the square root of a number. Newton's method is one of many methods of computing square roots.

For example, if one wishes to find the square root of 612, this is equivalent to finding the solution to

$$x^2 = 612$$

The function to use in Newton's method is then,

$$f(x) = x^2 - 612$$

with derivative,

$$f'(x) = 2x.$$

With an initial guess of 10, the sequence given by Newton's method is

$$\begin{matrix} x_1 & = & x_0 - \dfrac{f(x_0)}{f'(x_0)} & = & 10 - \dfrac{10^2 - 612}{2 \cdot 10} & = & 35.6 \\ x_2 & = & x_1 - \dfrac{f(x_1)}{f'(x_1)} & = & 35.6 - \dfrac{35.6^2 - 612}{2 \cdot 35.6} & = & \underline{2}6.395505617978\dots \\ x_3 & = & \vdots & = & \vdots & = & \underline{24.7}90635492455\dots \\ x_4 & = & \vdots & = & \vdots & = & \underline{24.7386}88294075\dots \\ x_5 & = & \vdots & = & \vdots & = & \underline{24.73863375376}7\dots \end{matrix}$$

where the correct digits are underlined. With only a few iterations one can obtain a solution accurate to many decimal places.

Solution of cos(*x*) = x^3 [edit]

Consider the problem of finding the positive number *x* with cos(*x*) = x^3. We can rephrase that as finding the zero of *f*(*x*) = cos(*x*) − x^3. We have *f'*(*x*) = −sin(*x*) − 3x^2. Since cos(*x*) ≤ 1 for all *x* and x^3 > 1 for *x* > 1, we know that our solution lies between 0 and 1. We try a starting value of x_0 = 0.5. (Note that a starting value of 0 will lead to an undefined result, showing the importance of using a starting point that is close to the solution.)

$$\begin{matrix} x_1 & = & x_0 - \dfrac{f(x_0)}{f'(x_0)} & = & 0.5 - \dfrac{\cos(0.5) - (0.5)^3}{-\sin(0.5) - 3(0.5)^2} & = & 1.112141637097 \\ x_2 & = & x_1 - \dfrac{f(x_1)}{f'(x_1)} & = & \vdots & = & \underline{0.}909672693736 \\ x_3 & = & \vdots & = & \vdots & = & \underline{0.86}7263818209 \\ x_4 & = & \vdots & = & \vdots & = & \underline{0.86547}7135298 \\ x_5 & = & \vdots & = & \vdots & = & \underline{0.8654740331}11 \\ x_6 & = & \vdots & = & \vdots & = & \underline{0.865474033102} \end{matrix}$$

The correct digits are underlined in the above example. In particular, x_6 is correct to the number of decimal places given. We see that the number of correct digits after the decimal point increases from 2 (for x_3) to 5 and 10, illustrating the quadratic convergence.

Pseudocode [edit]

The following is an example of using the Newton's Method to help find a root of a function `f` which has derivative `fprime`.

The initial guess will be $x_0 = 1$ and the function will be $f(x) = x^2 - 2$ so that $f'(x) = 2x$.

Each new iterative of Newton's method will be denoted by `x1`. We will check during the computation whether the denominator (`yprime`) becomes too small (smaller than `epsilon`), which would be the case if $f'(x_n) \approx 0$, since otherwise a large amount of error could be introduced.

```
%These choices depend on the problem being solved
x0 = 1                        %The initial value
f = @(x) x^2 - 2              %The function whose root we are trying to find
fprime = @(x) 2*x             %The derivative of f(x)
tolerance = 10^(-7)           %7 digit accuracy is desired
epsilon = 10^(-14)            %Don't want to divide by a number smaller than this

maxIterations = 20            %Don't allow the iterations to continue indefinitely
haveWeFoundSolution = false %Have not converged to a solution yet

for i = 1 : maxIterations

    y = f(x0)
    yprime = fprime(x0)

    if(abs(yprime) < epsilon)                          %Don't want to divide by too
small of a number
        % denominator is too small
        break;                                         %Leave the loop
    end

    x1 = x0 - y/yprime                                 %Do Newton's computation

    if(abs(x1 - x0)/abs(x1) < tolerance)               %If the result is within the
desired tolerance
        haveWeFoundSolution = true
        break;                                         %Done, so leave the loop
    end

    x0 = x1                                            %Update x0 to start the
process again

end

if (haveWeFoundSolution)
   ... % x1 is a solution within tolerance and maximum number of iterations
else
   ... % did not converge
end
```

See also [edit]

- Aitken's delta-squared process
- Bisection method
- Euler method
- Fast inverse square root
- Fisher scoring
- Gradient descent
- Integer square root
- Laguerre's method
- Leonid Kantorovich, who initiated the convergence analysis of Newton's method in Banach spaces.
- Methods of computing square roots
- Newton's method in optimization
- Richardson extrapolation
- Root-finding algorithm
- Secant method
- Steffensen's method
- Subgradient method

References [edit]

1. ^ "Accelerated and Modified Newton Methods".
2. ^ Ryaben'kii, Victor S.; Tsynkov, Semyon V. (2006), *A Theoretical Introduction to Numerical Analysis*, CRC Press, p. 243, ISBN 9781584886075.
3. ^ Dence, Thomas, "Cubics, chaos and Newton's method", *Mathematical Gazette* 81, November 1997, 403-408.
4. ^ Strang, Gilbert, "A chaotic search for *i*", "*The College Mathematics Journal* 22, January 1991, pp. 3-12 (esp. p. 6).
5. ^ McMullen, Curt, "Families of rational maps and iterative root-finding algorithms", *Ann. of Math. (2)* 125 (1987), no. 3, 467–493.

External links [edit]

Wikimedia Commons has media related to ***Newton Method***.

For a list of words relating to Newton's method, see the ***Newton's method*** category of article in Wikibooks.

- Hazewinkel, Michiel, ed. (2001), "Newton method", *Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4
- Weisstein, Eric W., "Newton's Method", *MathWorld*.
- Kendall E. Atkinson, *An Introduction to Numerical Analysis*, (1989) John Wiley & Sons, Inc, ISBN 0-471-62489-6
- Tjalling J. Ypma, Historical development of the Newton-Raphson method, *SIAM Review* **37** (4), 531–551, 1995. doi:10.1137/1037125.
- Bonnans, J. Frédéric; Gilbert, J. Charles; Lemaréchal, Claude; Sagastizábal, Claudia A. (2006). *Numerical optimization: Theoretical and practical aspects*. Universitext (Second revised ed. of translation of 1997 French ed.). Berlin: Springer-Verlag. pp. xiv+490. doi:10.1007/978-3-540-35447-5. ISBN 3-540-35445-X. MR 2265882.
- P. Deuflhard, *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms.* Springer Series in Computational Mathematics, Vol. 35. Springer, Berlin, 2004. ISBN 3-540-21099-7.
- C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, no 1 in Fundamentals of Algorithms, SIAM, 2003. ISBN 0-89871-546-6.
- J. M. Ortega, W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables.* Classics in Applied Mathematics, SIAM, 2000. ISBN 0-89871-461-3.
- Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Chapter 9. Root Finding and Nonlinear Sets of Equations Importance Sampling". *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York: Cambridge University Press. ISBN 978-0-521-88068-8.. See especially Sections 9.4, 9.6, and 9.7.
- Endre Süli and David Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, 2003. ISBN 0-521-00794-1.
- Kaw, Autar; Kalu, Egwu (2008). "Numerical Methods with Applications" (1st ed.).

v · t · e **Optimization: Algorithms, methods, and heuristics** [hide]

Unconstrained nonlinear: Methods calling ... [show]

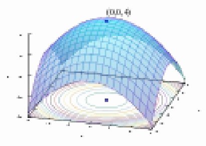
Constrained nonlinear [show]

Convex optimization [show]

Combinatorial [show]

Metaheuristics [show]

Categories (Algorithms and methods · Heuristics) · **Software**



Categories: Optimization algorithms and methods | Root-finding algorithms