WIKIPEDIA
The Free Encyclopedia

Newton's method

In numerical analysis, **Newton's method**, also known as the **Newton–Raphson method**, named after Isaac Newton and Joseph Raphson, is a root-finding algorithm which produces successively better approximations to the roots (or zeroes) of a real-valued function. The most basic version starts with a real-valued function f, its derivative f', and an initial guess x_0 for a root of f. If f satisfies certain assumptions and the initial guess is close, then

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

is a better approximation of the root than x_0. Geometrically, $(x_1, 0)$ is the x-intercept of the tangent of the graph of f at $(x_0, f(x_0))$: that is, the improved guess, x_1, is the unique root of the linear approximation of f at the initial guess, x_0. The process is repeated as

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

until a sufficiently precise value is reached. The number of correct digits roughly doubles with each step. 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.

Description

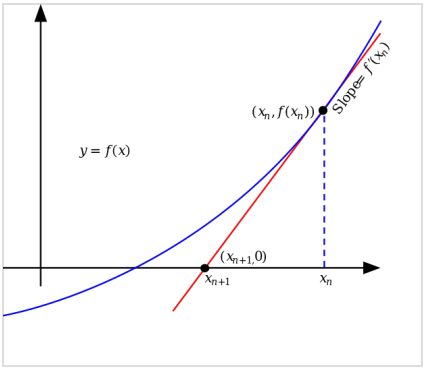
The idea is to start with an initial guess, then to approximate the function by its tangent line, and finally to compute the x-intercept of this tangent line. This x-intercept will typically be a better approximation to the original function's root than the first guess, and the method can be iterated.

If the tangent line to the curve $f(x)$ at $x = x_n$ intercepts the x-axis at x_{n+1} then the slope is

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

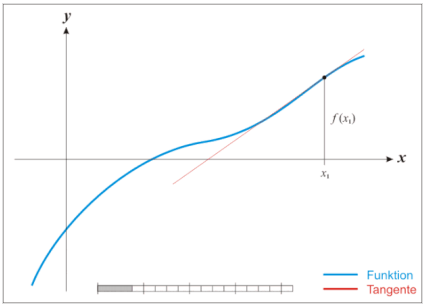
Solving for x_{n+1} gives

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



x_{n+1} is a better approximation than x_n for the root x of the function f (blue curve)

We start the process 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 doubles in every step. More details can be found in *§ Analysis* below.



Iteration typically improves the approximation

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

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 applied the method only to

polynomials, starting with an initial root estimate and extracting a sequence of error corrections. He used each correction to rewrite the polynomial in terms of the remaining error, and then solved for a new correction by neglecting higher-degree terms. He did not explicitly connect the method with derivatives or present a general formula. Newton applied this method to both numerical and algebraic problems, producing Taylor series in the latter case.

Newton may have derived his method from a similar, 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 since ancient times 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.[1]

Newton's method was first published in 1685 in *A Treatise of Algebra both Historical and Practical* by John Wallis.[2] In 1690, Joseph Raphson published a simplified description in *Analysis aequationum universalis*.[3] Raphson also applied the method only to polynomials, but he avoided Newton's tedious rewriting process by extracting each successive correction from the original polynomial. This allowed him to derive a reusable iterative expression for each problem. 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

Newton's method is a 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 the derivative of a function

Newton's method requires that the derivative can be calculated directly. An analytical expression for the derivative may not be easily obtainable or 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

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

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 $a = \frac{1}{2}$, the root will still be overshot, but the sequence will oscillate between two values. For $\frac{1}{2} < a < 1$, 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

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

A large error in the initial estimate can contribute to non-convergence of the algorithm. To overcome this problem one can often linearize the function that is being optimized using calculus, logs, differentials, or even using evolutionary algorithms, such as stochastic tunneling. Good initial estimates lie close to the final globally optimal parameter estimate. In nonlinear regression, the sum of squared errors (SSE) is only "close to" parabolic in the region of the final parameter estimates. Initial estimates found here will allow the Newton–Raphson method to quickly converge. It is only here that the Hessian matrix of the SSE is positive and the first derivative of the SSE is close to zero.

Mitigation of non-convergence

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 greater than 1

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, the following modified algorithm preserves the quadratic convergence rate:[4]

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

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.

If the multiplicity m of the root is finite then $g(x) = \frac{f(x)}{f'(x)}$ will have a root at the same location with multiplicity 1. Applying Newton's method to find the root of $g(x)$ recovers quadratic convergence in many cases although it generally involves the second derivative of $f(x)$. In a particularly simple case, if $f(x) = x^m$ then $g(x) = \frac{x}{m}$ and Newton's method finds the root in a single iteration with

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} = x_n - \frac{\frac{x_n}{m}}{\frac{1}{m}} = 0.$$

Analysis

Suppose that the function f has a zero at α, i.e., $f(\alpha) = 0$, 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.[5]

If f is continuously differentiable, its derivative is nonzero 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 $\frac{1}{2}$.[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

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 $f(\alpha)$ about x_n is:

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

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:

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

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

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

Remembering that x_{n+1} is defined by

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

one finds that

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

That is,

$$\varepsilon_{n+1} = \frac{-f''(\xi_n)}{2f'(x_n)} \cdot \varepsilon_n^2 . \quad (5)$$

Taking the absolute value of both sides gives

$$|\varepsilon_{n+1}| = \frac{|f''(\xi_n)|}{2|f'(x_n)|} \cdot \varepsilon_n^2 . \quad (6)$$

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

1. $f'(x) \neq 0$; for all $x \in I$, where I is the interval $[\alpha - |\varepsilon_0|, \alpha + |\varepsilon_0|]$;
2. $f''(x)$ is continuous, for all $x \in I$;
3. $M|\varepsilon_0| < 1$

where M is given by

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

If these conditions hold,

$$|\varepsilon_{n+1}| \leq M \cdot \varepsilon_n^2 .$$

Basins of attraction

The disjoint subsets of 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,[7] for the function $f(x) = x^3 - 2x^2 - 11x + 12 = (x - 4)(x - 1)(x + 3)$, the following initial conditions are in successive basins of attraction:

2.352 875 27 converges to 4;
2.352 841 72 converges to –3;
2.352 837 35 converges to 4;
2.352 836 327 converges to –3;
2.352 836 323 converges to 1.

Failure analysis

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

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

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:

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

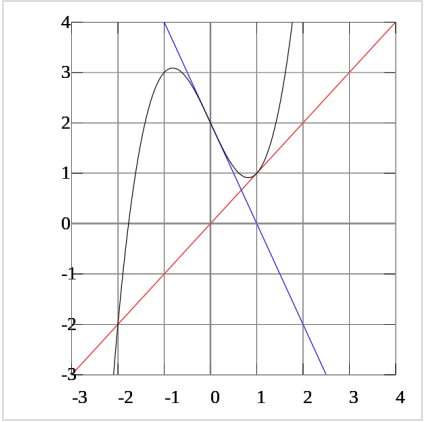
The same issue occurs if, instead of the starting point, any iteration point is stationary. Even if the derivative is small but not zero, the next iteration will be a far worse approximation.

Starting point enters a cycle

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 –1.769 292 35...



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

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

A simple example of a function where Newton's method diverges is trying to find the cube root of zero. The cube root is continuous and infinitely differentiable, except for $x = 0$, where its derivative is undefined:

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

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

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

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

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

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

Its derivative is:

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

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 $\frac{f(x)}{f'(x)}$ 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 $\frac{f(x)}{f'(x)}$).

Non-quadratic convergence

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

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

$$x - \frac{f(x)}{f'(x)} = \frac{x}{2}.$$

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 iterations starting at $x_0 = 1$ are

$x_0 = 1$
$x_1 = 0.500\ 250\ 376...$
$x_2 = 0.251\ 062\ 828...$
$x_3 = 0.127\ 507\ 934...$
$x_4 = 0.067\ 671\ 976...$
$x_5 = 0.041\ 224\ 176..$
$x_6 = 0.032\ 741\ 218...$
$x_7 = 0.031\ 642\ 362...$

it takes six iterations to reach a point where the convergence appears to be quadratic.

No second derivative

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

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

And

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

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 $\frac{4}{3}$ 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

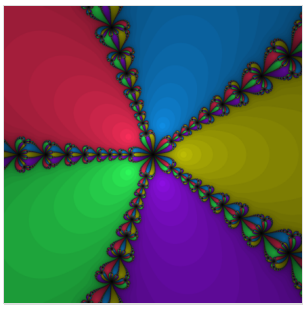
Complex functions

When dealing with complex functions, Newton's method can be directly applied to find their zeroes.[8] 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,[9] 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 4 or higher. However, McMullen gave a generally convergent algorithm for polynomials of degree 3.[10] Also, for any polynomial, Hubbard, Schleicher, and Sutherland gave a method for selecting a set of initial points such that Newton's method will certainly

converge at one of them at least.[11]



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

Chebyshev's third-order method

Nash–Moser iteration

Systems of equations

k variables, k functions

One may also use Newton's method to solve systems of k equations, which amounts to finding the (simultaneous) zeroes of k continuously differentiable functions $\boldsymbol{f} : \mathbb{R}^k \to \mathbb{R}$. This is equivalent to finding the zeroes of a single vector-valued function $\boldsymbol{F} : \mathbb{R}^k \to \mathbb{R}^k$. In the formulation given above, the scalars x_n are replaced by vectors $\mathbf{x}_n$ and instead of dividing the function $f(x_n)$ by its derivative $f'(x_n)$ one instead has to left multiply the function $F(\mathbf{x}_n)$ by the inverse of its $k \times k$ Jacobian matrix $J_F(\mathbf{x}_n)$. This results in the expression

$$\mathbf{x}_{n+1} = \mathbf{x}_n - J_F(\mathbf{x}_n)^{-1} F(\mathbf{x}_n).$$

Rather than actually computing the inverse of the Jacobian matrix, one may save time and increase numerical stability by solving the system of linear equations

$$J_F(\mathbf{x}_n)(\mathbf{x}_{n+1} - \mathbf{x}_n) = -F(\mathbf{x}_n)$$

for the unknown $\mathbf{x}_{n+1} - \mathbf{x}_n$.[12]

k variables, m equations, with $m > k$

The k-dimensional variant of Newton's method can be used to solve systems of greater than k (nonlinear) equations as well if the algorithm uses the generalized inverse of the non-square Jacobian matrix $J^+ = (J^{\mathrm{T}}J)^{-1}J^{\mathrm{T}}$ 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.

In a Banach space

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

$$X_{n+1} = X_n - \left(F'(X_n)\right)^{-1} F(X_n),$$

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.[13]

Over p-adic numbers

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

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

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)},$$

where the denominator is $f'(x_n)$ and not $f'(z_n)$. The iterations x_n will be strictly decreasing to the root while the iterations 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

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

q-analog

Newton's method can be generalized with the q-analog of the usual derivative.[14]

Modified Newton methods

Maehly's procedure

A nonlinear equation has multiple solutions in general. But if the initial value is not appropriate, Newton's method may not converge to the desired solution or may converge to the same solution found earlier. When we have already found N solutions of $f(x) = 0$, then the next root can be found by applying Newton's method to the next equation:[15][16]

$$F(x) = \frac{f(x)}{\prod_{i=1}^{N}(x - x_i)} = 0.$$

This method is applied to obtain zeros of the Bessel function of the second kind.[17]

Hirano's modified Newton method

Hirano's modified Newton method is a modification conserving the convergence of Newton method and avoiding unstableness.[18] It is developed to solve complex polynomials.

Interval Newton's method

Combining Newton's method with interval arithmetic is very useful in some contexts. This provides a stopping criterion that is more reliable than the usual ones (which are a small value of the function or a small variation of the variable between consecutive iterations). Also, this may detect cases where Newton's method converges theoretically but diverges numerically because of an insufficient floating-point precision (this is typically the case for polynomials of large degree, where a very small change of the variable may change dramatically the value of the function; see Wilkinson's polynomial).[19][20]

Consider $f \to C^1(X)$, where X is a real interval, and suppose that we have an interval extension F' of f', meaning that F' takes as input an interval $Y \subseteq X$ and outputs an interval $F'(Y)$ such that:

$$F'([y,y]) = \{f'(y)\}$$
$$F'(Y) \supseteq \{f'(y) \mid y \in Y\}.$$

We also assume that $0 \notin F'(X)$, so in particular f has at most one root in X. We then define the interval Newton operator by:

$$N(Y) = m - \frac{f(m)}{F'(Y)} = \left\{ m - \frac{f(m)}{z} \;\middle|\; z \in F'(Y) \right\}$$

where $m \in Y$. Note that the hypothesis on F' implies that $N(Y)$ is well defined and is an interval (see interval arithmetic for further details on interval operations). This naturally leads to the following sequence:

$$\begin{aligned} X_0 &= X \\ X_{k+1} &= N(X_k) \cap X_k. \end{aligned}$$

The mean value theorem ensures that if there is a root of f in X_k, then it is also in X_{k+1}. Moreover, the hypothesis on F' ensures that X_{k+1} is at most half the size of X_k when m is the midpoint of Y, so this sequence converges towards $[x^*, x^*]$, where x^* is the root of f in X.

If $F'(X)$ strictly contains 0, the use of extended interval division produces a union of two intervals for $N(X)$; multiple roots are therefore automatically separated and bounded.

Applications

Minimization and maximization problems

Newton's method can be used to find a minimum or maximum of a function $f(x)$. The derivative is zero at a minimum or maximum, so local 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

An important application is Newton–Raphson division, which can be used to quickly find the reciprocal of a number a, using only multiplication and subtraction, that is to say the number x such that $\frac{1}{x} = a$. We can rephrase that as finding the zero of $f(x) = \frac{1}{x} - a$. We have $f'(x) = -\frac{1}{x^2}$.

Newton's iteration is

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

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

Many transcendental equations can be solved up to an arbitrary precision by using Newton's method.

When Newton's method can be applied to a transcendental equation, and converges to a solution of the equation, this implies that the solution is a computable number that is exactly represented by the pair formed by an initial approximation and an algorithm for increasing the accuracy of any approximation.

Obtaining zeros of special functions

Newton's method is applied to the ratio of Bessel functions in order to obtain its root.[21]

Numerical verification for solutions of nonlinear equations

A numerical verification for solutions of nonlinear equations has been established by using Newton's method multiple times and forming a set of solution candidates.[22][23]

Cumulative probability density functions

Finding the cumulative probability density function, such as a Normal distribution to fit a known probability generally involves integral functions with no known means to solve in closed form. However, computing the derivatives needed to solve them numerically with Newton's method is generally known, making numerical solutions possible. For an example, see the numerical solution to the inverse Normal cumulative distribution.

Examples

Square root

Consider the problem of finding the square root of a number a, that is to say the positive number x such that $x^2 = a$. Newton's method is one of many methods of computing square roots. We can rephrase that as finding the zero of $f(x) = x^2 - a$. We have $f'(x) = 2x$.

For example, for finding the square root of 612 with an initial guess $x_0 = 10$, the sequence given by Newton's method is:

$$\begin{aligned}
x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} = 10 - \frac{10^2 - 612}{2 \times 10} = 35.6 \\
x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} = 35.6 - \frac{35.6^2 - 612}{2 \times 35.6} = 26.395\,505\,617\,978\ldots \\
x_3 &= \vdots = \vdots = 24.790\,635\,492\,455\ldots \\
x_4 &= \vdots = \vdots = 24.738\,688\,294\,075\ldots \\
x_5 &= \vdots = \vdots = 24.738\,633\,753\,767\ldots
\end{aligned}$$

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

Rearranging the formula as follows yields the Babylonian method of finding square roots:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - a}{2x_n} = \frac{1}{2}\left(2x_n - \left(x_n - \frac{a}{x_n}\right)\right) = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right)$$

i.e. the arithmetic mean of the guess, x_n and $\frac{a}{x_n}$.

Solution of $\cos(x) = x^3$

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 $\cos(x) \leq 1$ for all x and $x^3 > 1$ for $x > 1$, we know that our solution lies between 0 and 1.

For example, with an initial guess $x_0 = 0.5$, the sequence given by Newton's method is (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):

$$\begin{aligned}
x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} = 0.5 - \frac{\cos 0.5 - 0.5^3}{-\sin 0.5 - 3 \times 0.5^2} = 1.112\,141\,637\,097\ldots \\
x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} = \vdots = 0.909\,672\,693\,736\ldots \\
x_3 &= \vdots = \vdots = 0.867\,263\,818\,209\ldots \\
x_4 &= \vdots = \vdots = 0.865\,477\,135\,298\ldots \\
x_5 &= \vdots = \vdots = 0.865\,474\,033\,111\ldots \\
x_6 &= \vdots = \vdots = 0.865\,474\,033\,102\ldots
\end{aligned}$$

The correct digits are underlined in the above example. In particular, x_6 is correct to 12 decimal places. 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.

Code

The following is an implementation example of the Newton's method in the Python (version 3.x) programming language for finding a root of a function `f` which has derivative `f_prime`.

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 iteration 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.

```
def f(x):
    return x**2 - 2   # f(x) = x^2 - 2

def f_prime(x):
    return 2*x        # f'(x) = 2x

def newtons_method(x0, f, f_prime, tolerance, epsilon, max_iterations):
    """Newton's method

    Args:
      x0:              The initial guess
      f:               The function whose root we are trying to find
      f_prime:         The derivative of the function
      tolerance:       Stop when iterations change by less than this
      epsilon:         Do not divide by a number smaller than this
      max_iterations:  The maximum number of iterations to compute
    """
    for _ in range(max_iterations):
        y = f(x0)
        yprime = f_prime(x0)

        if abs(yprime) < epsilon:       # Give up if the denominator is too small
            break

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

        if abs(x1 - x0) <= tolerance:   # Stop when the result is within the desired tolerance
            return x1                   # x1 is a solution within tolerance and maximum number of iterations

        x0 = x1                         # Update x0 to start the process again

    return None                         # Newton's method did not converge
```

See also

- Aitken's delta-squared process
- Bisection method
- Euler method
- Fast inverse square root
- Fisher scoring
- Gradient descent
- Integer square root
- Kantorovich theorem
- Laguerre's method
- Methods of computing square roots
- Newton's method in optimization
- Richardson extrapolation
- Root-finding algorithm
- Secant method
- Steffensen's method
- Subgradient method

Notes

1. "Chapter 2. Seki Takakazu" (http://www.ndl.go.jp/math/e/s1/2.html). *Japanese Mathematics in the Edo Period*. National Diet Library. Retrieved 24 February 2019.
2. Wallis, John (1685). *A Treatise of Algebra, both Historical and Practical* (http://www.e-rara.ch/zut/content/titleinfo/2507537). Oxford: Richard Davis. doi:10.3931/e-rara-8842 (https://doi.org/10.3931%2Fe-rara-8842).
3. Raphson, Joseph (1697). *Analysis Æequationum Universalis* (https://archive.org/details/bub_gb_4nlbAAAAQAAJ) (in Latin) (2nd ed.). London: Thomas Bradyll. doi:10.3931/e-rara-13516 (https://doi.org/10.3931%2Fe-rara-13516).
4. "Accelerated and Modified Newton Methods" (https://web.archive.org/web/20190524083302/http://mathfaculty.fullerton.edu/mathews/n2003/NewtonAccelerateMod.html). Archived from the original (http://mathfaculty.fullerton.edu/mathews/n2003/newtonacceleratemod.html) on 24 May 2019. Retrieved 4 March 2016.
5. Ryaben'kii, Victor S.; Tsynkov, Semyon V. (2006), *A Theoretical Introduction to Numerical Analysis* (https://books.google.com/books?id=V8gWP031-hEC&pg=PA243), CRC Press, p. 243, ISBN 9781584886075.
6. Süli & Mayers 2003, Exercise 1.6

7. Dence, Thomas (November 1997). "Cubics, chaos and Newton's method". *Mathematical Gazette*. **81** (492): 403–408. doi:10.2307/3619617 (https://doi.org/10.2307%2F3619617). JSTOR 3619617 (https://www.jstor.org/stable/3619617). S2CID 125196796 (https://api.semanticscholar.org/CorpusID:125196796).
8. Henrici, Peter (1974). "Applied and Computational Complex Analysis". **1**. {{cite journal}}: Cite journal requires |journal= (help)
9. Strang, Gilbert (January 1991). "A chaotic search for *i*". *The College Mathematics Journal*. **22** (1): 3–12. doi:10.2307/2686733 (https://doi.org/10.2307%2F2686733). JSTOR 2686733 (https://www.jstor.org/stable/2686733).
10. McMullen, Curt (1987). "Families of rational maps and iterative root-finding algorithms" (https://dash.harvard.edu/bitstream/handle/1/9876064/McMullen_FamiliesRationalMap.pdf?sequence=1) (PDF). *Annals of Mathematics*. Second Series. **125** (3): 467–493. doi:10.2307/1971408 (https://doi.org/10.2307%2F1971408). JSTOR 1971408 (https://www.jstor.org/stable/1971408).
11. Hubbard, John; Schleicher, Dierk; Sutherland, Scott (October 2001). "How to find all roots of complex polynomials by Newton's method" (https://dx.doi.org/10.1007/s002220100149). *Inventiones Mathematicae*. **146** (1): 1–33. Bibcode:2001InMat.146....1H (https://ui.adsabs.harvard.edu/abs/2001InMat.146....1H). doi:10.1007/s002220100149 (https://doi.org/10.1007%2Fs002220100149). ISSN 0020-9910 (https://www.worldcat.org/issn/0020-9910). S2CID 12603806 (https://api.semanticscholar.org/CorpusID:12603806).
12. Kiusalaas, Jaan (March 2013). *Numerical Methods in Engineering with Python 3* (https://www.cambridge.org/9781107033856) (3rd ed.). New York: Cambridge University Press. pp. 175–176. ISBN 978-1-107-03385-6.
13. Yamamoto, Tetsuro (2001). "Historical Developments in Convergence Analysis for Newton's and Newton-like Methods". In Brezinski, C.; Wuytack, L. (eds.). *Numerical Analysis: Historical Developments in the 20th Century*. North-Holland. pp. 241–263. ISBN 0-444-50617-9.
14. Rajkovic, Stankovic & Marinkovic 2002
15. Press et al. 1992
16. Stoer & Bulirsch 1980
17. Zhang & Jin 1996
18. Murota, Kazuo (1982). "Global Convergence of a Modified Newton Iteration for Algebraic Equations". *SIAM Journal on Numerical Analysis*. **19** (4): 793–799. Bibcode:1982SJNA...19..793M (https://ui.adsabs.harvard.edu/abs/1982SJNA...19..793M). doi:10.1137/0719055 (https://doi.org/10.1137%2F0719055).
19. Moore, R. E. (1979). *Methods and applications of interval analysis* (Vol. 2). Siam.
20. Hansen, E. (1978). Interval forms of Newtons method. *Computing*, 20(2), 153–163.
21. Gil, Segura & Temme (2007)
22. Kahan (1968)
23. Krawczyk (1969)

References

- Gil, A.; Segura, J.; Temme, N. M. (2007). *Numerical methods for special functions* (https://www.researchgate.net/publication/220693008). Society for Industrial and Applied Mathematics. ISBN 978-0-89871-634-4.
- Süli, Endre; Mayers, David (2003). *An Introduction to Numerical Analysis*. Cambridge University Press. ISBN 0-521-00794-1.

Further reading

- 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 (https://doi.org/10.1137%2F1037125).
- Bonnans, J. Frédéric; Gilbert, J. Charles; Lemaréchal, Claude; Sagastizábal, Claudia A. (2006). *Numerical optimization: Theoretical and practical aspects* (https://www.springer.com/mathematics/applications/book/978-3-540-35445-1). Universitext (Second revised ed. of translation of 1997 French ed.). Berlin: Springer-Verlag. pp. xiv+490. doi:10.1007/978-3-540-35447-5 (https://doi.org/10.1007%2F978-3-540-35447-5). ISBN 3-540-35445-X. MR 2265882 (https://mathscinet.ams.org/mathscinet-getitem?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, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. (2007). "Chapter 9. Root Finding and Nonlinear Sets of Equations Importance Sampling" (http://apps.nrbook.com/empanel/index.html#pg=442). *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 (http://apps.nrbook.com/empanel/index.html#pg=456), 9.6 (http://apps.nrbook.com/empanel/index.html#pg=473), and 9.7 (http://apps.nrbook.com/empanel/index.html#pg=477).
- Avriel, Mordecai (1976). *Nonlinear Programming: Analysis and Methods*. Prentice Hall. pp. 216–221. ISBN 0-13-623603-0.

External links

- "Newton method" (https://www.encyclopediaofmath.org/index.php?title=Newton_method), *Encyclopedia of Mathematics*, EMS Press, 2001 [1994]
- Weisstein, Eric W. "Newton's Method" (https://mathworld.wolfram.com/NewtonsMethod.html). *MathWorld*.
- Newton's method, Citizendium. (http://en.citizendium.org/wiki/Newton%27s_method)
- Mathews, J., The Accelerated and Modified Newton Methods, Course notes. (https://web.archive.org/web/20190524083302/http://mathfaculty.fullerton.edu/mathews/n2003/NewtonAccelerateMod.html)
- Wu, X., Roots of Equations, Course notes. (http://www.ece.mcmaster.ca/~xwu/part2.pdf)