Algorithm for finding zeros of functionsThis article is about Newton's method for finding roots. For Newton's method for finding minima, see Newton's method in optimization.An illustration of Newton's method.In numerical analysis, the NewtonRaphson method, also known simply as Newton's 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 x0 for a root of f. If f satisfies certain assumptions and the initial guess is close, then x 1 = x 0 f (x 0) f (x 0) {\displaystyle x_{1}=x_{0}-{\frac {f(x_{0})}{f'(x_{0})}}} is a better approximation of the root than x0. Geometrically, (x1, 0) is the x-intercept of the tangent of the graph of f at (x0, f(x0)): that is, the improved guess, x1, is the unique root of the linear approximation of f at the initial guess, x0. The process is repeated as x n + 1 = x n f (x n) f (x n) {\displaystyle 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, and was succeeded by Halley's method. The method can also be extended to complex functions and to systems of equations.The purpose of Newton's method is to find a root of a function. The idea is to start with an initial guess at a root, approximate the function by its tangent line near the guess, and then take the root of the linear approximation as a next guess at the function's root. This will typically be closer to the function's root than the previous guess, and the method can be iterated.xn+1 is a better approximation than xn for the root x of the function f (blue curve)The best linear approximation to an arbitrary differentiable function f (x) {\displaystyle f(x)} near the point x = x n {\displaystyle x=x_{n}} is the tangent line to the curve, with equation f (x) f (x n) + f (x n) (x x n) . {\displaystyle f(x)\approx f(x_{n})+f'(x_{n})(x-x_{n}).} The root of this linear function, the place where it intercepts the x {\displaystyle x} -axis, can be taken as a closer approximate root x n + 1 {\displaystyle x_{n+1}} : x n + 1 = x n f (x n) f (x n) . {\displaystyle x_{n+1}=x_{n}-{\frac {f(x_{n})}{f'(x_{n})}}.} Iteration typically improves the approximationThe process can be started with any arbitrary initial guess x 0 {\displaystyle x_{0}} , though it will generally require fewer iterations to converge if the guess is close to one of the function's roots. The method will usually converge if f (x 0) 0 {\displaystyle f'(x_{0})eq 0} . Furthermore, for a root of multiplicity1, the convergence is at least quadratic (see Rate of convergence) in some sufficiently small neighbourhood of the root: the number of correct digits of the approximation roughly doubles with each additional step. More details can be found in Analysis below.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 {\displaystyle f} or its derivatives are computationally expensive to evaluate.In the Old Babylonian period (19th16th century BCE), the side of a square of known area could be effectively approximated, and this is conjectured to have been done using a special case of Newton's method, described algebraically below, by iteratively improving an initial estimate; an equivalent method can be found in Heron of Alexandria's Metrica (1st2nd century CE), so is often called Heron's method.[1] Jamshd al-Ksh used a method to solve xP N = 0 to find roots of N, a method that was algebraically equivalent to Newton's method, and in which a similar method was found in Trigonometria Britannica, published by Henry Briggs in 1633.[2]The method first appeared roughly in Isaac Newton's work 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).[3][4] However, while Newton gave the basic ideas, his method differs from the modern method given above. He 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 mathematician Franois Vite, however, the two methods are not the same.[3] The essence of Vite's own method can be found in the work of the mathematician Sharaf al-Din al-Tusi.[5]The Japanese mathematician Seki Kwa used a form of Newton's method in the 1680s to solve single-variable equations, though the connection with calculus was missing.[6]Newton's method was first published in 1685 in A Treatise of Algebra both Historical and Practical by John Wallis.[7] In 1690, Joseph Raphson published a simplified description in Analysis aequationum universalis.[8] 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 NewtonFourier 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.Newton's method is a powerful techniqueif the derivative of the function at the root is nonzero, then the convergence is at least 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.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.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.For example, in some cases, if the first derivative is not well behaved in the neighborhood of a particular root, then it is possible that Newton's method will fail to converge no matter where the initialization is set. 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.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.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:[9] x n + 1 = x n m f (x n) f (x n) . {\displaystyle 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) = 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) = xm then g(x) = x/m and Newton's method finds the root in a single iteration with x n + 1 = x n g (x n) g (x n) = x n x n m 1 m = 0 . {\displaystyle x_{n+1}=x_{n}-{\frac {g(x_{n})}{g'(x_{n})}}=x_{n}-{\frac {\;{\frac {x_{n}}{m}}\;}{\frac {1}{m}}}=0\,.} The function f(x) = x2 has a root at 0.[10] Since f is continuously differentiable at its root, the theory guarantees that Newton's method as initialized sufficiently close to the root will converge. However, since the derivative f is zero at the root, quadratic convergence is not ensured by the theory. In this particular example, the Newton iteration is given by x n + 1 = x n f (x n) f (x n) = 1 2 x n . {\displaystyle x_{n+1}=x_{n}-{\frac {f(x_{n})}{f'(x_{n})}}={\frac {1}{2}}x_{n}.} It is visible from this that Newton's method could be initialized anywhere and converge to zero, but at only a linear rate. If initialized at 1, dozens of iterations would be required before ten digits of accuracy are achieved.The function f(x) = x + x4/3 also has a root at 0, where it is continuously differentiable. Although the first derivative f is nonzero at the root, the second derivative f is nonexistent there, so that quadratic convergence cannot be guaranteed. In fact the Newton iteration is given by x n + 1 = x n f (x n) f (x n) = x n 4 / 3 3 + 4 x n 1 / 3 x n x n 1 / 3 3 . {\displaystyle x_{n+1}=x_{n}-{\frac {f(x_{n})}{f'(x_{n})}}={\frac {x_{n}^{4/3}}{3+4x_{n}^{1/3}}}\approx x_{n}\cdot {\frac {x_{n}^{1/3}}{3}}.} From this, it can be seen that the rate of convergence is superlinear but subquadratic. This can be seen in the following tables, the left of which shows Newton's method applied to the above f(x) = x + x4/3 and the right of which shows Newton's method applied to f(x) = x + x2. The quadratic convergence in iteration shown on the right is illustrated by the orders of magnitude in the distance from the iterate to the true root (0,1,2,3,5,10,20,39,...) being approximately doubled from row to row. While the convergence on the left is superlinear, the order of magnitude is only multiplied by about 4/3 from row to row (0,1,2,4,5,7,10,13,...).xnx + x4/3nxnx + x2n12121.4286 1012.1754 1013.3333 1014.4444 1011.4669 1021.8260 1026.6666 1027.1111 1029.0241 1049.8961 1043.9216 1033.9369 1032.5750 1052.6511 1051.5259 1051.5259 1052.4386 1072.4539 1072.3283 10102.3283 10105.0366 10105.0406 10105.4210 10205.4210 10201.3344 10131.3344 10132.9387 10392.9387 1039The rate of convergence is distinguished from the number of iterations required to reach a given accuracy. For example, the function f(x) = x20 1 has a root at 1. Since f (1) 0 and f is smooth, it is known that any Newton iteration convergent to 1 will converge quadratically. However, if initialized at 0.5, the first few iterates of Newton's method are approximately 26214, 24904, 23658, 22476, decreasing slowly, with only the 200th iterate being 1.0371. The following iterates are 1.0103, 1.00093, 1.0000082, and 1.00000000065, illustrating quadratic convergence. This highlights that quadratic convergence of a Newton iteration does not mean that only few iterates are required; this only applies once the sequence of iterates is sufficiently close to the root.[11]The function f(x) = x(1 + x2)1/2 has a root at 0. The Newton iteration is given by x n + 1 = x n f (x n) f (x n) = x n x n (1 + x n 2) 1 / 2 (1 + x n 2) 3 / 2 = x n 3 . {\displaystyle x_{n+1}=x_{n}-{\frac {f(x_{n})}{f'(x_{n})}}=x_{n}-{\frac {x_{n}(1+x_{n}^{2})^{-1/2}}{(1+x_{n}^{2})^{-3/2}}}=-x_{n}^{3}.} From this, it can be seen that there are three possible phenomena for a Newton iteration. If initialized strictly between 1, the Newton iteration will converge (super-)quadratically to 0; if initialized exactly at 1 or 1, the Newton iteration will oscillate endlessly between 1; if initialized anywhere else, the Newton iteration will diverge.[12] This same trichotomy occurs for f(x) = arctan x.[10]In cases where the function in question has multiple roots, it can be difficult to control, via choice of initialization, which root (if any) is identified by Newton's method. For example, the function f(x) = x(x2 1)(x 3)e(x 1)2/2 has roots at 1, 0, 1, and 3.[13] If initialized at 1.488, the Newton iteration converges to 0; if initialized at 1.487, it diverges to ; if initialized at 1.486, it converges to 1; if initialized at 1.485, it diverges to ; if initialized at 1.4843, it converges to 3; if initialized at 1.484, it converges to 1. This kind of subtle dependence on initialization is not uncommon; it is frequently studied in the complex plane in the form of the Newton fractal.Consider the problem of finding a root of f(x) = x1/3. The Newton iteration is x n + 1 = x n f (x n) f (x n) = x n x n 1 / 3 1 3 x n 2 / 3 = 2 x n . {\displaystyle x_{n+1}=x_{n}-{\frac {f(x_{n})}{f'(x_{n})}}=x_{n}-{\frac {x_{n}^{1/3}}{{\frac {1}{3}}x_{n}^{-2/3}}}=-2x_{n}.} Unless Newton's method is initialized at the exact root 0, it is seen that the sequence of iterates will fail to converge. For example, even if initialized at the reasonably accurate guess of 0.001, the first several iterates are 0.002, 0.004, 0.008, 0.016, reaching 1048.58, 2097.15, ... by the 20th iterate. This failure of convergence is not contradicted by the analytic theory, since in this case f is not differentiable at its root.In the above example, failure of convergence is reflected by the failure of f(xn) to get closer to zero as n increases, as well as by the fact that successive iterates are growing further and further apart. However, the function f(x) = x1/3ex2 also has a root at 0. The Newton iteration is given by x n + 1 = x n f (x n) f (x n) = x n (1 3 1 6 x n 2) . {\displaystyle x_{n+1}=x_{n}-{\frac {f(x_{n})}{f'(x_{n})}}=x_{n}\left(1-{\frac {3}{1-6x_{n}^{2}}}\right).} In this example, where again f is not differentiable at the root, any Newton iteration not starting exactly at the root will diverge, but with both xn + 1 xn and f(xn) converging to zero.[14] This is seen in the following table showing the iterates with initialization 1:nxf(xn)10.367881.69.0416 1021.93422.9556 1022.20481.0076 1022.43963.5015 1032.65051.2307 1032.84374.3578 1043.02321.5513 104Although the convergence of xn + 1 xn in this case is not very rapid, it can be proved from the iteration formula. This example highlights the possibility that a stopping criterion for Newton's method based only on the smallness of xn + 1 xn and f(xn) might falsely identify a root.The tangent lines of x3 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.It is easy to find situations for which Newton's method oscillates endlessly between two distinct values. For example, for Newton's method as applied to a function f to oscillate between 0 and 1, it is only necessary that the tangent line to f at 0 intersects the x-axis at 1 and that the tangent line to f at 1 intersects the x-axis at 0.[14] This is the case, for example, if f(x) = x3 2x + 2. For this function, it is even the case that Newton's iteration as initialized sufficiently close to 0 or 1 will asymptotically oscillate between these values. For example, Newton's method as initialized at 0.99 yields iterates 0.99, 0.06317, 1.00628, 0.03651, 1.00196, 0.01162, 1.00020, 0.00120, 1.000002, and so on. This behavior is present despite the presence of a root of f approximately equal to 1.76929.In some cases, it is not even possible to perform the Newton iteration. For example, if f(x) = x2 1, then the Newton iteration is defined by x n + 1 = x n f (x n) f (x n) = x n x n 2 1 2 x n = x n 2 + 1 2 x n . {\displaystyle x_{n+1}=x_{n}-{\frac {f(x_{n})}{f'(x_{n})}}=x_{n}-{\frac {x_{n}^{2}-1}{2x_{n}}}={\frac {x_{n}^{2}+1}{2x_{n}}}.} So Newton's method cannot be initialized at 0, since this would make x1 undefined. Geometrically, this is because the tangent line to f at 0 is horizontal (i.e. f (0) = 0), never intersecting the x-axis.Even if the initialization is selected so that the Newton iteration can begin, the same phenomenon can block the iteration from being indefinitely continued.If f has an incomplete domain, it is possible for Newton's method to send the iterates outside of the domain, so that it is impossible to continue the iteration.[14] For example, the natural logarithm function f(x) = ln x has a root at 1, and is defined only for positive x. Newton's iteration in this case is given by x n + 1 = x n f (x n) f (x n) = x n (1 ln x n) . {\displaystyle x_{n+1}=x_{n}-{\frac {f(x_{n})}{f'(x_{n})}}=x_{n}(1-\ln x_{n}).} So if the iteration is initialized at e, the next iterate is 0; if the iteration is initialized at a value larger than e, then the next iterate is negative. In either case, the method cannot be continued.Suppose that the function f has a zero at , i.e., f() = 0, and f is differentiable in a neighborhood of .If f is continuously differentiable and its derivative is nonzero at, then there exists a neighborhood of such that for all starting values x0 in that neighborhood, the sequence (xn) will converge to .[15]If f is continuously differentiable, its derivative is nonzero at, and it has a second derivative at, then the convergence is quadratic or faster. If the second derivative is not 0 at then the convergence is merely quadratic. If the third derivative exists and is bounded in a neighborhood of , then: x i + 1 = f ( ) 2 f ( ) (x i) 2 + O (x i) 3 , {\displaystyle \Delta x_{i+1}={\frac {f''(\alpha)}{2f'(\alpha)}}\left(\Delta x_{i}\right)^{2}+O\left(\Delta x_{i}\right)^{3}\,,} where x i x i . {\displaystyle \Delta x_{i}\triangleq x_{i}-\alpha \,.} If the derivative is 0 at , then the convergence is usually only linear. Specifically, if f is twice continuously differentiable, f() = 0 and f() 0, then there exists a neighborhood of such that, for all starting values x0 in that neighborhood, the sequence of iterates converges linearly, with rate 1/2.[16] Alternatively, if f() = 0 and f(x) 0 for x , xin a neighborhood U of , being a zero of multiplicity r, and if f Cr(U), then there exists a neighborhood of such that, for all starting values x0 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 , if f is twice differentiable in U+ and if f 0, f f > 0 in U+, then, for each x0 in U+ the sequence xk is monotonically decreasing to .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 . Then the expansion of f() about xn is: f ( ) = f (x n) + f (x n) (x n) + R 1 {\displaystyle f(\alpha)=f(x_{n})+f'(x_{n})(\alpha -x_{n})+R_{1}\,} 1where the Lagrange form of the Taylor series expansion remainder is R 1 = 1 2 ! f (n) (x n) 2 , {\displaystyle R_{1}={\frac {1}{2!}}f''(\xi _{n})\left(\alpha -x_{n}\right)^{2}\,,} where n is in between xn and .Since is the root, (1) becomes: 0 = f ( ) = f (x n) + f (x n) (x n) + 1 2 f (n) (x n) 2 {\displaystyle 0=f(\alpha)=f(x_{n})+f'(x_{n})(\alpha -x_{n})+{\tfrac {1}{2}}f''(\xi _{n})\left(\alpha -x_{n}\right)^{2}\,} 2Dividing equation (2) by f(xn) and rearranging gives f (x n) f (x n) + (x n) = f (n) 2 f (x n) (x n) 2 {\displaystyle {\frac {f(x_{n})}{f'(x_{n})}}+\left(\alpha -x_{n}\right)={\frac {-f''(\xi _{n})}{2f'(x_{n})}}\left(\alpha -x_{n}\right)^{2}} 3Remembering that xn + 1 is defined by x n + 1 = x n f (x n) f (x n) , {\displaystyle x_{n+1}=x_{n}-{\frac {f(x_{n})}{f'(x_{n})}}\,,} 4one finds that x n + 1 n + 1 = f (n) 2 f (x n) (x n n) 2 . {\displaystyle \underbrace {\alpha -x_{n+1}} _{\varepsilon _{n+1}}={\frac {-f''(\xi _{n})}{2f'(x_{n})}}{(\,\underbrace {\alpha -x_{n}} _{\varepsilon _{n}}\,)}^{2}\,.} That is, n + 1 = f (n) 2 f (x n) n 2 . {\displaystyle \varepsilon _{n+1}={\frac {-f''(\xi _{n})}{2f'(x_{n})}}\cdot \varepsilon _{n}^{2}\,.} 5Taking the absolute value of both sides gives | n + 1 | = | f (n) | 2 | f (x n) | n 2 . {\displaystyle \left|{\varepsilon _{n+1}}\right|={\frac {\left|f''(\xi _{n})\right|}{2\left|f'(x_{n})\right|}}\cdot \varepsilon _{n}^{2}\,.} 6Equation (6) shows that the order of convergence is at least quadratic if the following conditions are satisfied:f(x) 0; for all x I, where I is the interval [|0|, + |0|];f(x) is continuous, for all x I;M |0| < 1where M is given by M = 1 2 (sup x I | f (x) |) (sup x I 1 | f (x) |) . {\displaystyle M={\frac {1}{2}}\left(\sup _{x\in I}\vert f''(x)\vert \right)\left(\sup _{x\in I}{\frac {1}{\vert f'(x)\vert }}\right).\,} If these conditions hold, | n + 1 | M n 2 . {\displaystyle \vert \varepsilon _{n+1}\vert \leq M\cdot \varepsilon _{n}^{2}\,.} Suppose that f(x) is a concave function on an interval, which is strictly increasing. If it is negative at the left endpoint and positive at the right endpoint, the intermediate value theorem guarantees that there is a zero of f somewhere in the interval. From geometrical principles, it can be seen that the Newton iteration xi starting at the left endpoint is monotonically increasing and convergent, necessarily to .[17]Joseph Fourier introduced a modification of Newton's method starting at the right endpoint: y i + 1 = y i f (y i) f (x i) . {\displaystyle y_{i+1}=y_{i}-{\frac {f(y_{i})}{f'(x_{i})}}.} This sequence is monotonically decreasing and convergent. By passing to the limit in this definition, it can be seen that the limit of yi must also be the zero .[17] So, in the case of a concave increasing function with a zero, initialization is largely irrelevant. Newton iteration starting anywhere left of the zero will converge, as will Fourier's modified Newton iteration starting anywhere right of the zero. The accuracy at any step of the iteration can be determined directly from the difference between the location of the iteration from the left and the location of the iteration from the right. If f is twice continuously differentiable, it can be proved using Taylor's theorem that lim i y i + 1 x i + 1 (y i x i) 2 = 1 2 f ( ) f ( ) , {\displaystyle \lim _{i\to \infty }{\frac {y_{i+1}-x_{i+1}}{(y_{i}-x_{i})^{2}}}=-{\frac {1}{2}}{\frac {f''(\zeta)}{f'(\zeta)}},} showing that this difference in locations converges quadratically to zero.[17]All of the above can be extended to systems of equations in multiple variables, although in that context the relevant concepts of monotonicity and concavity are more subtle to formulate.[18] In the case of single equations in a single variable, the above monotonic convergence of Newton's method can also be generalized to replace concavity by positivity or negativity conditions on an arbitrary higher-order derivative of f. However, in this generalization, Newton's iteration is modified so as to be based on Taylor polynomials rather than the tangent line. In the case of concavity, this modification coincides with the standard Newton method.[19]If we seek the root of a single function f : R n R {\displaystyle f:\mathbf {R} ^{n}\to \mathbf {R} } then the error n = x n {\displaystyle \epsilon _{n}=x_{n}-\alpha } is a vector such that its components obey k (n + 1) = 1 2 ((n)) T Q k (n) + O ((n) 3) {\displaystyle \epsilon _{k}^{(n+1)}={\frac {1}{2}}(\epsilon ^{(n)})^{T}Q_{k}\epsilon ^{(n)}+O(\|\epsilon ^{(n)}\|^{3})} where Q k {\displaystyle Q_{k}} is a quadratic form: (Q k) i , j = ((D 2 f) 1) i , 3 f x j x k x {\displaystyle (Q_{k})_{i,j}=\sum _{\ell }((D^{2}f)^{-1})_{i,\ell }{\frac {\partial ^{3}f}{\partial x_{j}\partial x_{k}\partial x_{\ell }}}} evaluated at the root {\displaystyle \alpha } (where D 2 f {\displaystyle D^{2}f} is the 2nd derivative Hessian matrix).Newton's method is one of many known methods of computing square roots. Given a positive number a, the problem of finding a number x such that x2 = a is equivalent to finding a root of the function f(x) = x2 a. The Newton iteration defined by this function is given by x n + 1 = x n f (x n) f (x n) = x n x n 2 a 2 x n = 1 2 (x n + a x n) . {\displaystyle 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(x_{n}+{\frac {a}{x_{n}}}\right).} This happens to coincide with the "Babylonian" method of finding square roots, which consists of replacing an approximate root xn by the arithmetic mean of xn and axn. By performing this iteration, it is possible to evaluate a square root to any desired accuracy by only using the basic arithmetic operations.The following three tables show examples of the result of this computation for finding the square root of 612, with the iteration initialized at the values of 1, 10, and 20. Each row in a "xn" column is obtained by applying the preceding formula to the entry above it, for instance 306.5 = 1 2 (1 + 612 1) . {\displaystyle 306.5={\frac {1}{2}}\left(1+{\frac {612}{1}}\right).} xnf(xn)xnf(xn)xnf(xn)16111051220212306.59.3330 10435.6655.3625.328.09154.24836867862.3180 10426.395505618084.72224.74486166010.3081879.10799786445.6461 10324.79063549252.575624.73863453743.8777 10543.42212868221.2735 10324.73868829412.6985 10324.73863375376.1424 101328.7581624288215.0324.73863375382.9746 10925.019538536913.97724.74021067127.8024 10224.73863380402.4865 10624.73863375372.5256 1015The correct digits are underlined. It is seen that with only a few iterations one can obtain a solution accurate to many decimal places. The first table shows that this is true even if the Newton iteration were initialized by the very inaccurate guess of 1.When computing any nonzero square root, the first derivative of f must be nonzero at the root, and that f is a smooth function. So, even before any computation, it is known that any convergent Newton iteration has a quadratic rate of convergence. This is reflected in the above tables by the fact that once a Newton iterate gets close to the root, the number of correct digits approximately doubles with each iteration.Consider the problem of finding the positive number x with cos x = x3. We can rephrase that as finding the zero of f(x) = cos(x) x3. We have f(x) = sin(x) 3x2. Since cos(x) 1 for all x and x3 > 1 for x > 1, we know that our solution lies between 0 and 1. A starting value of 0 will lead to an undefined result which illustrates the importance of using a starting point close to the solution. For example, with an initial guess x0 = 0.5, the sequence given by Newton's method is: x 1 = x 0 f (x 0) f (x 0) = 0.5 cos 0.5 0.5 3 sin 0.5 3 0.5 2 = 1.112 141 637 097 x 2 = x 1 f (x 1) f (x 1) = = 0. _ 909 672 693 736 x 3 = = = 0.86 _ 7 263 818 209 x 4 = = = 0.865 47 _ 7 135 298 x 5 = = = 0.865 474 033 1 _ 11 x 6 = = = 0.865 474 033 102 _ {\displaystyle {\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\times 0.5^{2}}}&=&1.112\,141\,637\,097\dots \\x_{2}&=&x_{1}-{\dfrac {f(x_{1})}{f'(x_{1})}}&=&\vdots &=&{\underline {0.}}909\,672\,693\,736\dots \\x_{3}&=&\vdots &=&\vdots &=&{\underline {0.86}}7\,263\,818\,209\dots \\x_{4}&=&\vdots &=&\vdots &=&{\underline {0.865\,47}}7\,135\,298\dots \\x_{5}&=&\vdots &=&\vdots &=&{\underline {0.865\,474\,033\,1}}11\dots \\x_{6}&=&\vdots &=&\vdots &=&{\underline {0.865\,474\,033\,102}}\dots \end{matrix}}} The correct digits are underlined in the above example. In particular, x6 is correct to 12 decimal places. We see that the number of correct digits after the decimal point increases from 2 (for x3) to 5 and 10, illustrating the quadratic convergence.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 f : R k R . {\displaystyle f:\mathbb {R} ^{k}\to \mathbb {R} .} This is equivalent to finding the zeroes of a single vector-valued function F : R k R k . {\displaystyle F:\mathbb {R} ^{k}\to \mathbb {R} ^{k}.} In the formulation given above, the scalars xn are replaced by vectors xn and instead of dividing the function f(xn) by its derivative f(xn) one instead has to left multiply the function F(xn) by the inverse of its k k Jacobian matrix JF(xn).[20][21][22] This results in the expression x n + 1 = x n J F (x n) 1 F (x n) . {\displaystyle \mathbf {x} _{n+1}=\mathbf {x} _{n}-J_{F}(\mathbf {x} _{n})^{-1}F(\mathbf {x} _{n}).} or, by solving the system of linear equations J F (x n) (x n + 1 x n) = F (x n) {\displaystyle J_{F}(\mathbf {x} _{n})(\mathbf {x} _{n+1}-\mathbf {x} _{n})=-F(\mathbf {x} _{n})} for the unknown xn + 1 xn.[23]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+ = (JTJ)1JT 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 GaussNewton algorithm for more information.For example, the following set of equations needs to be solved for vector of points [x 1 , x 2] , {\displaystyle \ [\ x_{1},x_{2}\ ]\ ,} given the vector of known values [2 , 3] . {\displaystyle \ [\ 2,3\ ]~.} [24] 5 x 1 2 + x 1 x 2 2 + sin 2 (2 x 2) = 2 e 2 x 1 x 2 + 4 x 2 = 3 {\displaystyle {\begin{array}{lcr}5\ x_{1}^{2}+x_{1}\ x_{2}^{2}+\sin ^{2}(2\ x_{2})&=\quad 2\\e^{2\ x_{1}-x_{2}}+4\ x_{2}&=\quad 3\end{array}}} the function vector, F (X k) , {\displaystyle \ F(X_{k})\ ,} and Jacobian Matrix, J (X k) {\displaystyle \ J(X_{k})\ } for iteration k, and the vector of known values, Y , {\displaystyle \ Y\ ,} are defined below. F (X k) = [f 1 (X k) f 2 (X k)] = [5 x 1 2 + x 1 x 2 2 + sin 2 (2 x 2) e 2 x 1 x 2 + 4 x 2] k J (X k) = [f 1 (X) x 1 , f 1 (X) x 2 f 2 (X) x 1 , f 2 (X) x 2] k = [10 x 1 + x 2 2 , 2 x 1 x 2 + 4 sin (2 x 2) cos (2 x 2) 2 e 2 x 1 x 2 , e 2 x 1 x 2 + 4] k Y = [2 3] {\displaystyle {\begin{aligned}~&F(X_{k})~=~{\begin{bmatrix}{\begin{aligned}~&f_{1}(X_{k})\\~&f_{2}(X_{k})\end{aligned}}\end{bmatrix}}~=~{\begin{bmatrix}{\begin{aligned}~&5\ x_{1}^{2}+x_{1}\ x_{2}^{2}+\sin ^{2}(2\ x_{2})\\~&e^{2\ x_{1}-x_{2}}+4\ x_{2}\end{aligned}}\end{bmatrix}}_{k}\\~&J(X_{k})={\begin{bmatrix}~{\frac {\ \partial {f_{1}(X)}\ }{\partial {x_{1}}}}\ ,&~{\frac {\ \partial {f_{1}(X)}\ }{\partial {x_{2}}}}~\\~{\frac {\ \partial {f_{2}(X)}\ }{\partial {x_{1}}}}\ ,&~{\frac {\ \partial {f_{2}(X)}\ }{\partial {x_{2}}}}~\end{bmatrix}}_{k}~=~{\begin{bmatrix}{\begin{aligned}~&10\ x_{1}+x_{2}^{2}\ ,&&2\ x_{1}\ x_{2}+4\ \sin(2\ x_{2})\ \cos(2\ x_{2})\\~&2\ e^{2\ x_{1}-x_{2}}\ ,&&-e^{2\ x_{1}-x_{2}}+4\end{aligned}}\end{bmatrix}}_{k}\\~&Y={\begin{bmatrix}~2~\\~3~\end{bmatrix}}\end{aligned}}} Note that F (X k) {\displaystyle \ F(X_{k})\ } could have been rewritten to absorb Y , {\displaystyle \ Y\ ,} and thus eliminate Y {\displaystyle Y} from the equations. The equation to solve for each iteration are [10 x 1 + x 2 2 , 2 x 1 x 2 + 4 sin (2 x 2) cos (2 x 2) 2 e 2 x 1 x 2 , e 2 x 1 x 2 + 4] k [c 1 c 2] k + 1 = [5 x 1 2 + x 1 x 2 2 + sin 2 (2 x 2) 2 e 2 x 1 x 2 + 4 x 2 3] k {\displaystyle {\begin{aligned}{\begin{bmatrix}{\begin{aligned}~&~10\ x_{1}+x_{2}^{2}\ ,&&2x_{1}x_{2}+4\ \sin(2\ x_{2})\ \cos(2\ x_{2})~\\~&~2\ e^{2\ x_{1}-x_{2}}\ ,&&-e^{2\ x_{1}-x_{2}}+4~\end{aligned}}\end{bmatrix}}_{k}{\begin{bmatrix}~c_{1}~\\~c_{2}~\end{bmatrix}}_{k+1}={\begin{bmatrix}~5\ x_{1}^{2}+x_{1}\ x_{2}^{2}+\sin ^{2}(2\ x_{2})-2~\\~e^{2\ x_{1}-x_{2}}+4\ x_{2}-3~\end{bmatrix}}_{k}\end{aligned}}} and X k + 1 = X k C k + 1 {\displaystyle X_{k+1}~=~X_{k}-C_{k+1}} The iterations should be repeated until [i = 1 i = 2 | f (x i) k (y i) k |] < E , {\displaystyle \ {\Bigg [}\sum _{i=1}^{i=2}{\Bigl |}f(x_{i})_{k}-(y_{i})_{k}{\Bigr |}{\Bigg]}}