

Newton's Method



Abdulaziz Almutairi and Vagelis Plevris
Department of Civil and Environmental
Engineering, College of Engineering, Qatar
University, Doha, Qatar

Synonyms

Newton optimization method; Newton's
iterative method; Newton–Raphson method;
Second-order optimization method

Definition

Newton's method in single-variable optimization is a second-order iterative procedure that locates stationary points of a differentiable function using both the first and second derivatives. Each iteration updates the estimate by minimizing a local quadratic model of the function, enabling fast convergence under appropriate smoothness conditions. The approach naturally generalizes to multivariable optimization, where the Hessian matrix replaces the scalar second derivative.

Historical Background and Importance

Newton's method is a classical iterative technique that originated in the seventeenth century, when Isaac Newton and Joseph Raphson introduced it as a procedure for solving nonlinear equations. For a detailed historical survey and discussion of Newton's method in modern optimization, see Polyak (2007). In its original form, known as the Newton–Raphson method, the algorithm generates successive approximations to a root of a function by using the function itself and its first derivative (Burden and Faires 2015). The basic iteration is given by

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad (1)$$

which provides an improved estimate of the root under suitable conditions.

This root-finding principle was later adapted to optimization. Instead of solving the equation $f(x) = 0$, the goal in optimization is to find stationary points where the first derivative vanishes, i.e., $f'(x) = 0$. Applying the Newton–Raphson idea to the derivative of the objective function naturally leads to Newton's method for optimization. As a result, the optimization variant requires both the first and second derivatives of the function: the numerator comes from the slope $f'(x)$, while the denominator involves the curvature $f''(x)$.

In the context of single-variable optimization, Newton's method constructs a local quadratic approximation of the objective function at each step and updates the iterate by minimizing this approximation. Compared with first-order methods such as gradient descent, which rely only on slope information, Newton's method also incorporates curvature, enabling quadratic convergence near a solution. Because of this efficiency and solid theoretical foundation, it is recognized as one of the fundamental algorithms in numerical optimization. It serves both as a practical tool for solving optimization problems and as a basis for more advanced techniques, such as quasi-Newton and trust-region methods, and it continues to provide a standard benchmark in the study and development of optimization algorithms (Nocedal and Wright 2006).

Mathematical Formulation

The basic principle of Newton's method in optimization is to approximate the objective function locally by a quadratic model and then to minimize this model iteratively. Consider a smooth scalar objective function $f : \mathbb{R} \rightarrow \mathbb{R}$ that is twice continuously differentiable. Expanding $f(x)$ in a second-order Taylor series around a current iterate x_i gives:

$$f(x) \approx f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(x_i)(x - x_i)^2 \quad (2)$$

This quadratic model approximates the local behavior of the objective function. To obtain the next iterate, the minimum of this quadratic approximation is sought. Differentiating the right-hand side with respect to x and setting it equal to zero gives:

$$f'(x_i) + f''(x_i)(x - x_i) = 0 \quad (3)$$

Solving for x leads to the update formula:

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)} \quad (4)$$

This is the iterative scheme of Newton's method for single-variable optimization. At each step, the current point is adjusted by a correction term given by the slope divided by the curvature. The process is repeated until convergence criteria are satisfied, typically when the first derivative becomes sufficiently small or successive iterates differ by less than a prescribed tolerance.

The interpretation of this update is straightforward:

- The **numerator** $f'(x_i)$ represents the slope of the function at the current point.
- The **denominator** $f''(x_i)$ represents the curvature of the function, which scales the correction.

In practice, the iteration stops when a convergence criterion is met or when a preset iteration limit is reached. Two common stopping criteria are when the step size becomes negligible or the derivative approaches zero. Either condition indicates that the method is close to a stationary point.

Convergence Properties

Newton's method in single-variable optimization is well known for its **quadratic convergence** when applied under suitable conditions. This means that once the iterates are sufficiently close to a stationary point, the error decreases proportionally to the square of the previous error, leading to very rapid progress toward the solution (Fletcher 2000). In practice, only a few iterations may be required to achieve high accuracy when the method is applied near a minimizer.

The key requirements for quadratic convergence are:

- The function $f(x)$ must be twice continuously differentiable in a neighborhood of the solution.
- The stationary point x^* must satisfy $f'(x^*) = 0$ and $f''(x^*) \neq 0$, ensuring a nondegenerate stationary point.

- The initial guess x_0 must be sufficiently close to the solution so that the Taylor series approximation used in the derivation is valid.

When these conditions hold, the method is remarkably efficient. However, if the starting point is too far from the solution, the method may fail to converge or may converge to an undesired stationary point. In particular:

- If $f''(x_i)$ is very small or zero at some iterate, the correction step becomes unstable.
- If $f''(x^*) < 0$, the method may converge to a local maximum rather than a minimum.
- If $f''(x^*) = 0$, the stationary point is an inflection point, and Newton's method is not applicable.

Compared with first-order methods such as gradient descent, Newton's method has a smaller region of attraction but achieves much faster convergence when started close to a suitable solution. This trade-off highlights both its power and its limitations: while highly effective near well-behaved stationary points, it requires careful initialization and cannot guarantee global convergence.

In practical implementations, convergence is usually checked by prescribing tolerances on both the change in iterates and the derivative. The condition $|x_i - x_{i-1}| < \varepsilon_x$ ensures that the sequence of iterates has stabilized, while $|f'(x_i)| < \varepsilon_g$ verifies that the stationarity condition is satisfied. Together, these criteria provide a reliable stopping rule. At the same time, the method is highly sensitive to the initial guess: because it is only locally convergent, starting points within the basin of attraction of a minimum typically lead to rapid convergence, whereas starting points near maxima or inflection points may result in divergence or convergence to undesired stationary points. This dual nature underscores both the effectiveness and the limitations of Newton's method.

Applications

Newton's method for single-variable optimization is widely used whenever a smooth scalar objective with available first and second derivatives must be minimized efficiently. Typical applications include benchmarking and teaching in numerical analysis, optimizing one-parameter energy functions or likelihood-based objectives, and handling scalar design or calibration variables across engineering and the sciences. Because of its quadratic local convergence, it is particularly effective when a reasonable initial guess is available.

Beyond purely single-variable settings, the underlying Newton–Raphson principle extends naturally to multivariate optimization and nonlinear equation solving, and many practical applications rely on this generalization. In engineering, Newton iterations form the basis of nonlinear finite element equilibrium analysis (Belytschko et al. 2014) and are used in design optimization where design parameters are adjusted to improve performance (Rao 2009). In electrical power engineering, the Newton–Raphson procedure is the standard solver for power-flow equations because of its robustness and convergence behavior (Grainger and Stevenson 2021). Similar Newton-based techniques are employed in control engineering and robotics for solving nonlinear implicit relationships such as inverse kinematics.

In the physical sciences, Newton and quasi-Newton methods are widely used for energy minimization and equilibrium calculations, including determining molecular geometries on potential energy surfaces (Schlegel 2011). These problems are often multivariate, yet they rely on the same second-order principles as the single-variable method discussed here. Single-parameter Newton steps also appear in chemical equilibrium calculations when isolating one scalar variable within a coupled system.

Economics and finance frequently use Newton's method in nonlinear estimation and optimization tasks. Scalar or low-dimensional maximum likelihood problems and nonlinear regressions commonly employ Newton–

Raphson updates, enabling rapid refinement of parameter estimates (Greene 2018). In portfolio optimization and related problems, multivariate Newton-based interior-point algorithms are used to solve convex formulations efficiently.

In computer science and machine learning, Newton's method inspires a range of second-order optimization approaches. Although full Hessian computations are typically multivariate, the core Newton idea, using curvature information, remains central. Examples include logistic regression via iteratively reweighted least squares and advanced large-scale optimizers that approximate Newton steps to accelerate convergence (Bottou et al. 2018).

Taken together, these applications highlight both the practical importance of the single-variable Newton method and its influence on many multivariate extensions. Its reliability, fast local convergence, and foundational role in numerous algorithms explain why Newton's method remains one of the most widely used techniques in modern optimization and scientific computing.

Advantages and Limitations

Newton's method offers quadratic local convergence near a suitable stationary point and can often reach high accuracy within only a few iterations. By exploiting both slope and curvature, it builds a locally accurate quadratic model and typically requires fewer iterations than first-order methods once close to the solution. It also serves as a cornerstone for more advanced second-order techniques.

Despite these strengths, the method is highly sensitive to the initial guess: starting far from the solution may cause slow progress, divergence, or convergence to an unintended stationary point. It also relies on second-derivative information, which can be costly to compute and may produce unstable or excessively large steps when the local curvature is close to zero. With negative curvature, the method may move toward a local maximum rather than a minimum. In practice, safeguards such as line searches or trust-region

strategies are commonly employed to ensure stability.

Compared with first-order methods such as gradient descent, Newton's method usually requires far fewer iterations to converge because it incorporates curvature information through the second derivative. This advantage, however, comes at the cost of higher per-iteration complexity. Other approaches, such as the Secant method, reduce computational effort by approximating second derivatives from successive first derivatives, though at the expense of a slower superlinear convergence rate. These trade-offs explain why Newton's method remains a standard benchmark against which other optimization algorithms are evaluated.

Algorithm and Implementation

Goal. Minimize a twice continuously differentiable function $f : \mathbb{R} \rightarrow \mathbb{R}$.

Step-by-step guide:

Step 1. Initialization

Choose an initial guess x_0 for the minimizer; any real starting value is acceptable, though values closer to the true solution generally improve performance. Set the iteration counter $i = 0$ and specify a maximum number of iterations. Optionally define simple stopping checks with small thresholds, such as when the change in x becomes negligible or when the derivative is close to zero.

Step 2. Compute derivatives

At the current point x_i , evaluate the first derivative $f'(x_i)$ and the second derivative $f''(x_i)$.

Step 3. Update

Form the Newton iterate using Eq. (4) to find the next candidate solution, provided the second derivative at x_i is nonzero (to avoid division by zero).

Step 4. Convergence check

After computing the new iterate, test for convergence. Stop if the step size is sufficiently small, if the magnitude of the derivative is below the prescribed tolerance, or if the maximum number of iterations has been reached.

Step 5. Result

If a stopping condition is satisfied, accept the current point x_i as an approximation of the solution. To classify the point, inspect the second derivative: positive indicates a local minimum, negative a local maximum, and values near zero are inconclusive. Otherwise, set $i \leftarrow i + 1$ and return to Step 2.

Numerical Example

Consider the function

$$f(x) = x^4 - 3x^3 + 2 \quad (5)$$

The value of $x \in [-1, 3]$ that minimizes f is sought. The initial point is set to $x_0 = 3$.

Solution

The graph of the function over the interval $x \in [-1, 3]$ is shown in Fig. 1. From the plot, it is clear that the minimum value of the function occurs between $x = 2$ and $x = 2.5$.

To apply the method, the first and second derivatives of f are first computed:

$$f'(x) = 4x^3 - 9x^2 \quad (6)$$

$$f''(x) = 12x^2 - 18x \quad (7)$$

Using these expressions for the first and second derivatives, Newton's method is applied starting from the initial guess $x_0 = 3$. The iteration results are summarized in Table 1.

The procedure is terminated at iteration 5, since at this point the derivative magnitude is below a chosen tolerance (i.e., sufficiently close to zero for practical purposes). The computed minimizer is

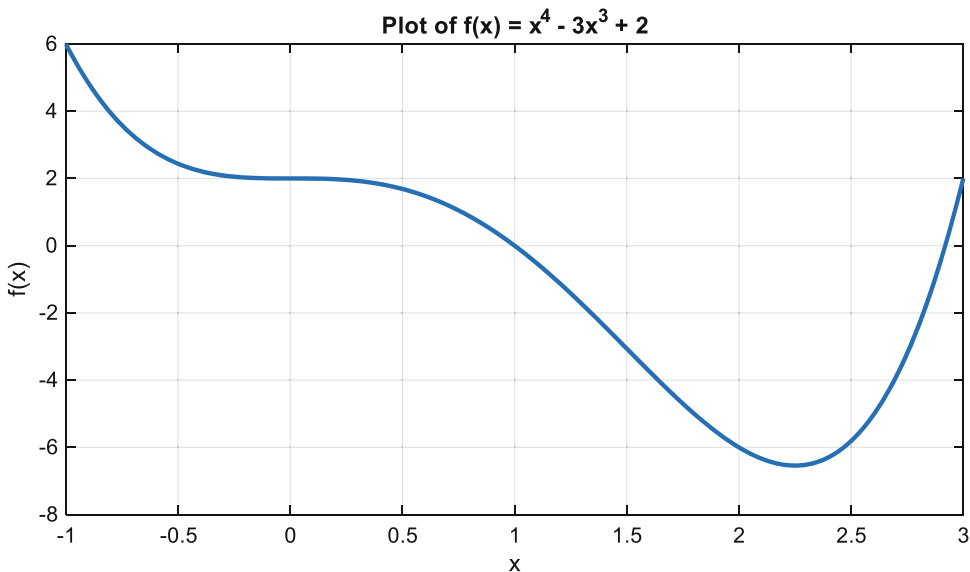
$$x^* = 2.25, f(x^*) \approx -6.54297 \quad (8)$$

To confirm that the obtained solution corresponds to a minimum, the second derivative is evaluated at $x^* = 2.25$. Since

$$f''(2.25) = 20.25 > 0, \quad (9)$$

the function is locally convex at this point. This establishes that $x^* = 2.25$ is indeed a local minimum of $f(x)$.

Table 2 presents the application of Newton's method to the same function, but with the initial guess $x_0 = 1$. In this case, the iteration sequence



Newton's Method, Fig. 1 Graph of the function $f(x)$ on the interval $x \in [-1, 3]$

Newton's Method, Table 1 Iteration results for Newton's method applied to $f(x)$ with $x_0 = 3$

Iteration i	x_i	$f(x_i)$	$f'(x_i)$	$f''(x_i)$	Step size $ x_i - x_{i-1} $
0	3	2	27	54	
1	2.5	-5.8125	6.25	30	0.5
2	2.291667	-6.52495	0.875289	21.77083	0.208333
3	2.251462	-6.54295	0.029644	20.30266	0.040205
4	2.250002	-6.54297	3.84×10^{-5}	20.25007	0.00146
5	2.25	-6.54297	6.47×10^{-11}	20.25	1.9×10^{-6}

Newton's Method, Table 2 Iteration results for Newton's method applied to $f(x)$ with $x_0 = 1$

Iteration i	x_i	$f(x_i)$	$f'(x_i)$	$f''(x_i)$	Step size $ x_i - x_{i-1} $
0	1	0	-5	-6	
1	0.166667	1.986883	-0.23148	-2.66667	0.833333
2	0.079861	1.998513	-0.05536	-1.36097	0.086806
3	0.039182	1.999822	-0.01358	-0.68685	0.040679
4	0.019416	1.999978	-0.00336	-0.34496	0.019766
5	0.009666	1.999997	-0.00084	-0.17286	0.00975
6	0.004822	2	-0.00021	-0.08652	0.004843
7	0.002409	2	-5.2×10^{-5}	-0.04328	0.002414
8	0.001204	2	-1.3×10^{-5}	-0.02165	0.001205

converges after about eight steps to the point $x \approx 0$, where the function value is $f(x) = 2$.

However, inspection of the function graph in Fig. 1 shows that $x = 0$ does not correspond to either a local minimum or a local maximum. Instead, it is an inflection point, where the curvature changes sign. This example demonstrates that Newton's method may converge to stationary points that are not minimizers, depending on the initial guess. Figure 2 illustrates this behavior by showing Newton's method paths from the two different starting points.

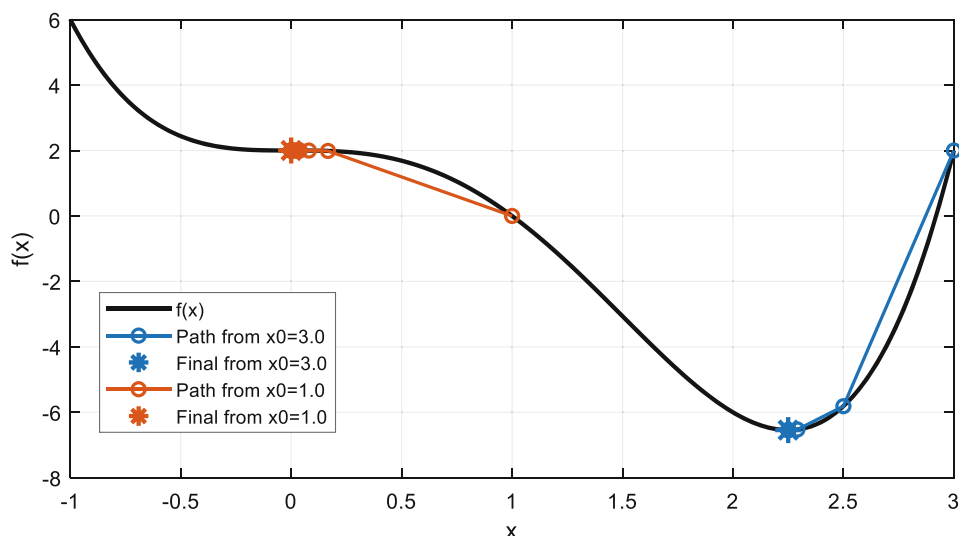
The black curve represents $f(x)$. The blue path corresponds to the results in Table 1 ($x_0 = 3$) and converges to the true minimizer near $x \approx 2.25$, with the final iterate marked by an asterisk. The orange path corresponds to Table 2 ($x_0 = 1$) and converges to $x = 0$, the inflection point. The figure highlights the basins of attraction: initial guesses to the right of about 1.5 converge to the minimizer at 2.25, while those to the left are drawn toward the stationary point at 0 or stall near

1.5 where the local curvature vanishes. Starting exactly at 1.5 fails because the Newton update is undefined there.

Taken together, the two examples highlight both the strengths and limitations of Newton's method. With a suitable starting point, it converges rapidly and demonstrates its characteristic quadratic rate, but with an unfavorable initialization, it may instead converge to a stationary point that is not a minimizer or fail to converge altogether. This balance of efficiency and sensitivity captures the essence of Newton's method and explains why it has inspired numerous variants designed to combine speed with greater robustness.

Cross-References

- [Gradient Descent](#)
- [Secant Method](#)



Newton's Method, Fig. 2 Newton's method iteration paths from different starting points

Competing Interest Declaration The author(s) has no competing interests to declare that are relevant to the content of this manuscript.

References

- Belytschko, T., W.K. Liu, B. Moran, and K. Elkhodary, Nonlinear finite elements for continua and structures. 2nd ed. 2014: Wiley. ISBN: 978-1-118-63270-3
- Bottou L, Curtis FE, Nocedal J (2018) Optimization methods for large-scale machine learning. SIAM Rev 60(2):223–311. <https://doi.org/10.1137/16m1080173>
- Burden RL, Faires JD (2015) Numerical analysis, 10th edn. Cengage Learning. ISBN: 978-1305253667
- Fletcher, R., Practical methods of optimization. 2nd ed. 2000: Wiley. ISBN: 978-0471494638
- Grainger J, Stevenson W (2021) Power system analysis (SI). McGraw Hill. ISBN: 978-1259008351
- Greene, W.H., Econometric analysis. 8th ed. 2018: Pearson. ISBN: 978-0134461366
- Nocedal J, Wright S (2006) Numerical optimization, Springer series in operations research and financial engineering, 2nd edn. Springer. ISBN: 978-0387303031.
- Polyak BT (2007) Newton's method and its use in optimization. Eur J Oper Res 181(3):1086–1096. <https://doi.org/10.1016/j.ejor.2005.06.076>
- Rao, S.S., Engineering optimization: theory and practice. 4th ed. 2009: Wiley. ISBN: 978-0470183526
- Schlegel HB (2011) Geometry optimization. WIREs Comput Mol Sci 1(5):790–809. <https://doi.org/10.1002/wcms.34>