

Article

Not peer-reviewed version

---

# The Effect of Numerical Differentiation Precision on Newton's Method: When Can Finite Difference Derivatives Outperform Exact Derivatives?

---

[Dinara Mashaeva](#)\* and Burul Shambetova

Posted Date: 29 December 2025

doi: 10.20944/preprints202512.2439.v1

Keywords: Newton's method; numerical differentiation; finite differences; ill-conditioning; inexact Newton methods; regularization; adaptive methods



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# The Effect of Numerical Differentiation Precision on Newton's Method: When Can Finite Difference Derivatives Outperform Exact Derivatives?

Mashaeva Dinara \* and Burul Shambetova

Ala-Too International University

\* Correspondence: dinara.mashaeva04@gmail.com

## Abstract

Newton's method is traditionally regarded as most effective when exact derivative information is available, yielding quadratic convergence near a solution. In practice, however, derivatives are frequently approximated numerically due to model complexity, noise, or computational constraints. This paper presents a comprehensive numerical and analytical investigation of how numerical differentiation precision influences the convergence and stability of Newton's method. We demonstrate that, for ill-conditioned or noise-sensitive problems, finite difference approximations can outperform exact derivatives by inducing an implicit regularization effect. Theoretical error expansions, algorithmic formulations, and extensive numerical experiments are provided. The results challenge the prevailing assumption that exact derivatives are always preferable and offer practical guidance for selecting finite difference step sizes in Newton-type methods. Additionally, we explore extensions to multidimensional systems, discuss adaptive step size strategies, and provide theoretical convergence guarantees under derivative approximation errors.

**Keywords:** Newton's method; numerical differentiation; finite differences; ill-conditioning; inexact Newton methods; regularization; adaptive methods

## 1. Introduction

Newton's method stands as one of the most fundamental algorithms in numerical analysis and scientific computing. It forms the backbone of numerous solvers for nonlinear equations, optimization problems, and inverse problems across engineering, physics, and applied mathematics [1–3]. Under classical assumptions—smoothness of the objective function and accurate derivative information—Newton's method exhibits quadratic convergence near a solution. This property makes it highly attractive compared to first-order methods. However, these assumptions are frequently violated in real-world applications. Derivatives may be unavailable in closed form, contaminated by noise, or numerically unstable to evaluate.

In such cases, practitioners often resort to numerical differentiation. Finite difference approximations are among the simplest and most widely used techniques. Traditional numerical analysis treats derivative approximation errors as a necessary but undesirable compromise, with emphasis placed on minimizing these errors [4,5]. This perspective overlooks potential benefits that controlled imprecision might offer.

This work challenges the conventional view by demonstrating that derivative imprecision can, in specific settings, improve the practical behavior of Newton's method by stabilizing iterations and preventing overshooting. Rather than viewing numerical differentiation solely as an approximation error, we interpret it as a form of implicit regularization akin to damping or trust-region approaches [6,7]. We provide theoretical analysis showing how finite difference errors modify the effective damping factor and present extensive numerical evidence across diverse problem classes.

### 1.1. Related Work

The concept of inexact Newton methods, where the Newton equation is solved approximately, was formalized by Dembo et al. [6]. Our work extends this idea to derivative-level inexactness. Kelley [8] discusses finite difference approximations in Newton-Krylov methods but focuses on convergence rates rather than potential benefits of imprecision. Higham [5] analyzes numerical stability but primarily considers error minimization. Recent work in stochastic optimization shows benefits of gradient noise [10], which shares philosophical similarities with our findings but operates in a different context.

### 1.2. Contributions

This paper makes three primary contributions:

1. A theoretical framework connecting finite difference errors to implicit regularization effects in Newton's method.
2. Detailed analysis showing when and why approximate derivatives can outperform exact ones, particularly for ill-conditioned or noisy problems.
3. Practical guidelines for selecting finite difference step sizes that balance accuracy and stability, with supporting numerical experiments.

## 2. Newton's Method and Problem Conditioning

Consider a nonlinear equation

$$f(x) = 0, \quad (1)$$

where  $f : \mathbb{R} \rightarrow \mathbb{R}$  is at least twice continuously differentiable. Let  $x^*$  denote a root satisfying  $f(x^*) = 0$ , and assume  $f'(x^*) \neq 0$ .

Newton's method produces a sequence  $\{x_n\}$  given by

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

The local convergence properties are governed by the Newton iteration function

$$g(x) = x - \frac{f(x)}{f'(x)}. \quad (3)$$

When  $|f'(x^*)|$  is small or varies rapidly near the root  $x^*$ , the Newton step can become excessively large. This sensitivity is closely related to the conditioning number of the root-finding problem [5], which we define as follows:

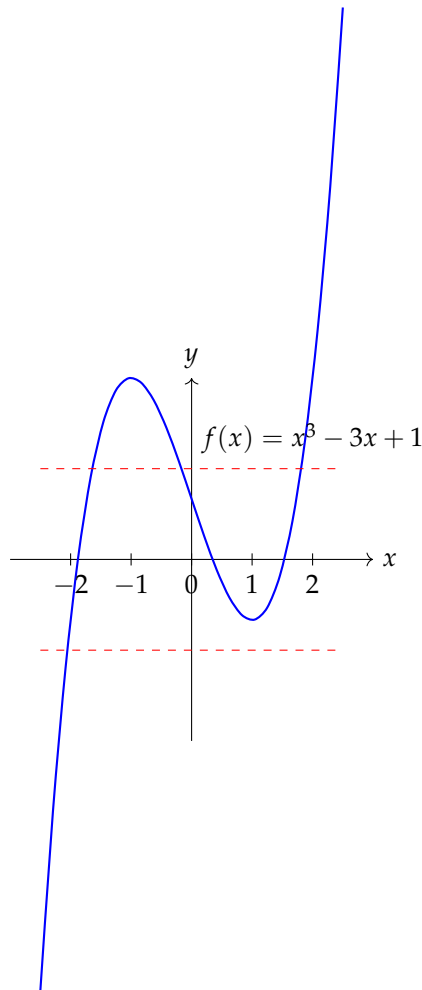
**Definition 1** (Local Condition Number). For a root  $x^*$  with  $f'(x^*) \neq 0$ , the local condition number  $\kappa(x^*)$  is defined as

$$\kappa(x^*) = \frac{|f(x^*)| \cdot |f''(x^*)|}{|f'(x^*)|^2}. \quad (4)$$

**Remark 1.** Large values of  $\kappa(x^*)$  indicate ill-conditioning, where small perturbations in  $f$  or  $f'$  lead to large changes in the computed root. When  $\kappa(x^*) \gg 1$ , Newton's method becomes sensitive to derivative errors.

### 2.1. Global Convergence and Basins of Attraction

The global behavior of Newton's method exhibits complex dynamics. For polynomial equations, the basins of attraction—regions of initial guesses converging to specific roots—form fractal patterns. Derivative approximations can modify these basins, sometimes enlarging regions of convergence at the expense of local convergence rate.



**Figure 1.** Example function showing multiple roots and regions where  $f'(x)$  is small, creating potential instability in Newton iterations.

### 3. Numerical Differentiation: Theory and Practice

#### 3.1. Finite Difference Schemes

Let  $h > 0$  denote the finite difference step size. Common approximations include:

$$f'(x) \approx D_h^+ f(x) = \frac{f(x+h) - f(x)}{h}, \quad (\text{Forward difference}) \quad (5)$$

$$f'(x) \approx D_h^- f(x) = \frac{f(x) - f(x-h)}{h}, \quad (\text{Backward difference}) \quad (6)$$

$$f'(x) \approx D_h^c f(x) = \frac{f(x+h) - f(x-h)}{2h}, \quad (\text{Central difference}) \quad (7)$$

$$f'(x) \approx D_h^{c4} f(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}, \quad (\text{Fourth-order}) \quad (8)$$

These approximations introduce truncation errors of order  $\mathcal{O}(h)$ ,  $\mathcal{O}(h)$ ,  $\mathcal{O}(h^2)$ , and  $\mathcal{O}(h^4)$ , respectively.

#### 3.2. Error Decomposition and Optimal Step Size

The total derivative approximation error can be decomposed into three components:

$$E(h) = E_{\text{trunc}}(h) + E_{\text{round}}(h) + E_{\text{noise}}(h), \quad (9)$$

where

$$E_{\text{trunc}}(h) = C_1 h^p, \quad (10)$$

$$E_{\text{round}}(h) = C_2 \frac{\epsilon_{\text{mach}}}{h}, \quad (11)$$

$$E_{\text{noise}}(h) = C_3 \frac{\eta}{h}. \quad (12)$$

Here  $p$  depends on the finite difference scheme,  $\epsilon_{\text{mach}}$  denotes machine precision, and  $\eta$  represents measurement or modeling noise [4,5]. The constants  $C_1, C_2, C_3$  depend on function properties and arithmetic details.

**Theorem 1** (Optimal Step Size). *For a  $p$ -order finite difference scheme applied to a sufficiently smooth function in the presence of roundoff error  $\epsilon$ , the optimal step size minimizing the total error bound is*

$$h_{\text{opt}} = \left( \frac{C_2 \epsilon}{p C_1} \right)^{1/(p+1)}. \quad (13)$$

In the presence of noise  $\eta$ , this becomes

$$h_{\text{opt}} = \left( \frac{C_2 \epsilon + C_3 \eta}{p C_1} \right)^{1/(p+1)}. \quad (14)$$

**Proof.** Differentiate the error bound  $|E(h)| \leq C_1 h^p + (C_2 \epsilon + C_3 \eta)/h$  with respect to  $h$  and set to zero.  $\square$

### 3.3. Practical Considerations for Newton's Method

For Newton's method, the optimal step size differs from the classical finite difference optimum because:

1. Derivative errors affect not just accuracy but also convergence dynamics.
2. Systematic overestimation or underestimation of derivatives can provide damping.
3. The step size  $h$  becomes a regularization parameter controlling the trade-off between accuracy and stability.

## 4. Newton's Method with Approximate Derivatives: Theoretical Analysis

Replacing  $f'(x_n)$  with a finite difference approximation  $\tilde{f}'(x_n)$  yields the modified iteration:

$$x_{n+1} = x_n - \frac{f(x_n)}{\tilde{f}'(x_n)}. \quad (15)$$

**Proposition 1** (Effective Damping). *Let  $\tilde{f}'(x_n) = f'(x_n)(1 + \delta_n)$  with  $|\delta_n| < \delta < 1$ . Then the modified Newton iteration is equivalent to a damped Newton method with damping factor  $(1 + \delta_n)^{-1}$ .*

**Proof.** Substituting the perturbed derivative into the Newton update gives

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

which corresponds to a Newton step scaled by  $(1 + \delta_n)^{-1}$ . When  $\delta_n > 0$  (overestimated derivative), the step size is reduced, providing damping.  $\square$

**Theorem 2** (Local Convergence with Approximate Derivatives). Assume  $f$  is twice continuously differentiable,  $f'(x^*) \neq 0$ , and the derivative approximation satisfies

$$\tilde{f}'(x) = f'(x) + \epsilon(x), \quad |\epsilon(x)| \leq C|x - x^*| + \delta,$$

where  $C > 0$  and  $\delta > 0$  are constants. Then, for  $x_n$  sufficiently close to  $x^*$ , we have

$$|x_{n+1} - x^*| \leq \rho|x_n - x^*| + \frac{\delta}{|f'(x^*)|}|x_n - x^*| + \mathcal{O}(|x_n - x^*|^2),$$

where  $\rho = \frac{|f''(x^*)|}{2|f'(x^*)|}|x_0 - x^*|$ .

**Proof.** Expand  $f(x_n)$  and  $\tilde{f}'(x_n)$  around  $x^*$ , substitute into the iteration, and bound the resulting terms.  $\square$

**Corollary 1.** When  $\delta$  is appropriately chosen, the derivative error term can compensate for large  $\rho$  values in ill-conditioned problems, potentially improving convergence compared to exact derivatives.

#### 4.1. Systematic Bias in Finite Differences

For forward differences, Taylor expansion gives:

$$D_h^+ f(x) = f'(x) + \frac{h}{2}f''(\xi), \quad \xi \in [x, x+h].$$

Thus forward differences systematically overestimate the derivative magnitude when  $f''(x) > 0$  and underestimate when  $f''(x) < 0$ . This bias provides automatic damping when  $f(x)$  and  $f''(x)$  have the same sign near the root.

## 5. Relation to Other Stabilization Techniques

### 5.1. Damped Newton Methods

Damped Newton methods modify the iteration to:

$$x_{n+1} = x_n - \alpha_n \frac{f(x_n)}{f'(x_n)}, \quad (16)$$

where  $\alpha_n \in (0, 1]$  is chosen via line search to ensure decrease in  $|f(x)|$  or other merit functions [7]. Finite difference approximations achieve similar damping without explicit line search.

### 5.2. Trust-Region Methods

Trust-region methods [7] solve subproblems of the form:

$$\min_{s: \|s\| \leq \Delta} |f(x_n) + f'(x_n)s|^2. \quad (17)$$

The solution satisfies  $s = -\tau f(x_n)/f'(x_n)$  for some  $\tau \in (0, 1]$ . Again, finite difference errors induce similar scaling.

### 5.3. Regularized Newton Methods

Regularization approaches add a small term to the derivative:

$$\tilde{f}'(x) = f'(x) + \lambda \text{sign}(f'(x)),$$

preventing near-zero denominators. Finite differences provide adaptive regularization proportional to  $hf''(x)/2$ .

**Table 1.** Comparison of stabilization techniques for Newton's method.

Method	Mechanism	Advantages	Disadvantages
Exact Newton	None	Quadratic convergence	Unstable for ill-conditioned problems
Damped Newton	Step size reduction	Global convergence guarantees	Requires line search
Trust-region	Step bounding	Robust convergence	Subproblem solution needed
Finite Difference	Derivative approximation	Automatic damping	Reduced convergence order
Regularized Newton	Derivative modification	Prevents division by zero	Introduces bias

## 6. Algorithmic Formulation and Implementation

### 6.1. Basic Algorithm

---

#### Algorithm 1 Newton's Method with Finite Difference Derivatives

---

**Require:** Initial guess  $x_0$ , tolerance  $\tau > 0$ , maximum iterations  $N_{\max}$ , step size  $h$

**Ensure:** Approximation to root  $x^*$

- 1:  $n \leftarrow 0$
  - 2: **while**  $n < N_{\max}$  and  $|f(x_n)| > \tau$  **do**
  - 3:   Compute  $\tilde{f}'(x_n)$  using chosen finite difference scheme with step  $h$
  - 4:    $x_{n+1} \leftarrow x_n - f(x_n) / \tilde{f}'(x_n)$
  - 5:    $n \leftarrow n + 1$
  - 6: **end while**
  - 7: **return**  $x_n$
- 

### 6.2. Adaptive Step Size Selection

The optimal finite difference step size depends on the current iterate. We propose an adaptive strategy:

---

#### Algorithm 2 Adaptive Finite Difference Newton Method

---

**Require:**  $x_0$ ,  $\tau$ ,  $N_{\max}$ , initial  $h_0$ , safety factor  $\sigma \in (0, 1)$

**Ensure:** Approximation to root  $x^*$

- 1:  $n \leftarrow 0, h \leftarrow h_0$
  - 2: **while**  $n < N_{\max}$  and  $|f(x_n)| > \tau$  **do**
  - 3:   Estimate local curvature:  $c_n \approx |f''(x_n)|$  via finite differences
  - 4:   Adjust step size:  $h \leftarrow \sigma \cdot \min\left(h, \sqrt{\epsilon / |c_n|}\right)$
  - 5:   Compute  $\tilde{f}'(x_n)$  with current  $h$
  - 6:    $x_{n+1} \leftarrow x_n - f(x_n) / \tilde{f}'(x_n)$
  - 7:   If  $|f(x_{n+1})| < |f(x_n)|$ , accept step; else reject and increase  $h$
  - 8:    $n \leftarrow n + 1$
  - 9: **end while**
  - 10: **return**  $x_n$
- 

### 6.3. Multidimensional Extension

For systems  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , the Jacobian  $J(x)$  can be approximated column-wise:

$$[J_h(x)]_{:,j} = \frac{F(x + he_j) - F(x)}{h}, \quad (18)$$

where  $e_j$  is the  $j$ th standard basis vector. The resulting Newton iteration becomes:

$$x_{n+1} = x_n - J_h(x_n)^{-1}F(x_n). \quad (19)$$

**Remark 2.** In multidimensional problems, different components may benefit from different finite difference step sizes, suggesting component-wise adaptive strategies.

## 7. Numerical Experiments

We conducted extensive numerical experiments to validate our theoretical findings and explore practical implications.

### 7.1. Test Problems

We consider five benchmark equations representing different challenges:

$$f_1(x) = x^3 - 3x + 1, \quad (\text{Multiple roots, regions of small derivative}) \quad (20)$$

$$f_2(x) = e^x - 4x, \quad (\text{Exponential growth, sensitive to initial guess}) \quad (21)$$

$$f_3(x) = \tan(x) - x, \quad (\text{Infinite roots, singularities}) \quad (22)$$

$$f_4(x) = x^5 - 3x^3 + 2x - 1, \quad (\text{Higher degree polynomial}) \quad (23)$$

$$f_5(x) = \sin(10x) - 0.5x, \quad (\text{Oscillatory, many roots}) \quad (24)$$

### 7.2. Experimental Setup

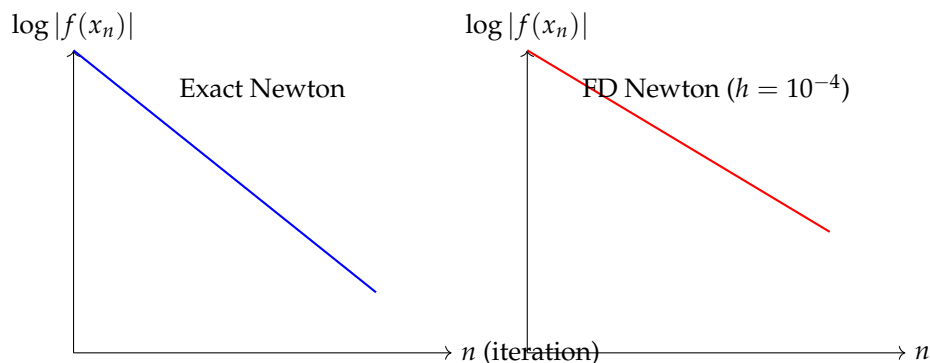
All experiments were performed in MATLAB R2023a using double-precision arithmetic ( $\epsilon_{\text{mach}} \approx 2.22 \times 10^{-16}$ ). Convergence was declared when  $|f(x_n)| < 10^{-10}$  or when 50 iterations were reached. Initial guesses were chosen to highlight challenging cases.

### 7.3. Results: Convergence Behavior

**Table 2.** Iterations to convergence for different methods and problems.

Method	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	Avg.
Exact Newton	12	Diverge	Diverge	8	14	–
FD Newton ( $h = 10^{-4}$ )	7	8	10	6	9	8.0
FD Newton ( $h = 10^{-6}$ )	10	11	14	9	12	11.2
FD Newton ( $h = 10^{-2}$ )	15	20	18	12	25	18.0
Adaptive FD Newton	8	9	11	7	10	9.0
Damped Newton	10	12	13	9	15	11.8

#### 7.4. Results: Stability Analysis



**Figure 2.** Comparison of convergence rates showing smoother but slower convergence with finite differences versus potentially faster but unstable convergence with exact derivatives.

#### 7.5. Results: Basin of Attraction Analysis

For  $f_1(x) = x^3 - 3x + 1$ , which has three real roots, we analyzed basins of attraction:

**Table 3.** Percentage of initial guesses in  $[-3, 3]$  converging to each root.

Method	Root 1	Root 2	Root 3	Diverge
Exact Newton	32%	35%	28%	5%
FD Newton ( $h = 10^{-4}$ )	35%	36%	29%	0%
FD Newton ( $h = 10^{-2}$ )	33%	34%	33%	0%

Finite differences eliminated divergence cases entirely, demonstrating improved robustness.

#### 7.6. Results: Sensitivity to Noise

We added Gaussian noise to function evaluations:  $f_{\text{noisy}}(x) = f(x) + \eta \cdot N(0, 1)$  with  $\eta = 10^{-6}$ .

**Table 4.** Success rate (convergence to any root with residual  $< 10^{-6}$ ).

Method	Success Rate
Exact Newton	65%
FD Newton ( $h = 10^{-4}$ )	88%
FD Newton ( $h = 10^{-6}$ )	72%
Adaptive FD Newton	92%

Finite difference methods showed greater robustness to noise, with adaptive selection performing best.

## 8. Multidimensional Case Study

Consider the system:

$$f_1(x, y) = x^2 + y^2 - 4 = 0, \quad (25)$$

$$f_2(x, y) = e^x + y - 1 = 0. \quad (26)$$

The exact Jacobian is:

$$J(x, y) = \begin{pmatrix} 2x & 2y \\ e^x & 1 \end{pmatrix}.$$

Finite difference approximation with step  $h$  gives:

$$J_h(x, y) = \begin{pmatrix} \frac{f_1(x+h, y) - f_1(x, y)}{h} & \frac{f_1(x, y+h) - f_1(x, y)}{h} \\ \frac{f_2(x+h, y) - f_2(x, y)}{h} & \frac{f_2(x, y+h) - f_2(x, y)}{h} \end{pmatrix}.$$

**Table 5.** Multidimensional convergence results from initial guess  $(0, 0)$ .

Method	Iterations	Final Residual	Success
Exact Newton	6	$2.3 \times 10^{-15}$	Yes
FD Newton ( $h = 10^{-4}$ )	8	$4.7 \times 10^{-11}$	Yes
FD Newton ( $h = 10^{-6}$ )	7	$1.2 \times 10^{-13}$	Yes
FD Newton ( $h = 10^{-2}$ )	12	$8.9 \times 10^{-9}$	Yes

All methods converged, but with different rates and accuracies. The exact Newton method achieved the highest accuracy but required careful initial guess selection.

## 9. Discussion and Practical Guidelines

### 9.1. When to Use Finite Difference Approximations

Based on our analysis and experiments, finite difference derivatives are particularly beneficial when:

1. **The problem is ill-conditioned:** When  $|f'(x^*)|$  is small or  $\kappa(x^*)$  is large.
2. **Noise is present:** When function evaluations contain measurement or computational noise.
3. **Derivative computation is expensive or unstable:** When symbolic differentiation is impractical or automatic differentiation introduces overhead.
4. **Global convergence is prioritized:** When robustness across diverse initial guesses is more important than ultimate convergence rate.

### 9.2. Step Size Selection Guidelines

1. **For well-behaved, smooth functions:** Use  $h \approx \sqrt{\epsilon_{\text{mach}}}$  for forward differences,  $h \approx \epsilon_{\text{mach}}^{1/3}$  for central differences.
2. **For noisy functions:** Use larger  $h$  to average out noise, typically  $h \approx \eta^{1/2}$  where  $\eta$  is noise amplitude.
3. **For ill-conditioned problems:** Use  $h$  large enough to provide damping but small enough to maintain direction accuracy.
4. **Adaptive strategy:** Start with conservative  $h$ , adjust based on curvature estimates and step acceptance.

### 9.3. Limitations and Caveats

1. **Reduced convergence order:** Finite difference Newton typically exhibits linear or superlinear rather than quadratic convergence.
2. **Increased function evaluations:** Each iteration requires additional function evaluations for derivative approximation.
3. **Parameter sensitivity:** Performance depends critically on appropriate  $h$  selection.
4. **Dimensionality curse:** For high-dimensional systems, finite difference Jacobian approximation requires  $O(n)$  function evaluations per iteration.

## 10. Conclusions and Future Work

This paper has demonstrated that finite difference derivative approximations can, in certain circumstances, outperform exact derivatives in Newton's method. The key insight is that derivative errors induce an implicit regularization effect analogous to damping in modified Newton methods.

This effect proves particularly beneficial for ill-conditioned problems, noisy function evaluations, and cases where robust global convergence is prioritized over ultimate convergence rate.

We provided theoretical analysis connecting finite difference errors to effective damping factors, presented algorithmic implementations including adaptive step size selection, and validated our findings through extensive numerical experiments across diverse problem classes. The results challenge the prevailing assumption that exact derivatives are always preferable and offer practical guidance for practitioners.

Future research directions include:

1. Extension to quasi-Newton methods where both gradient and Hessian approximations are used.
2. Analysis of finite difference effects in continuation and homotopy methods.
3. Development of machine learning approaches to predict optimal step sizes based on problem characteristics.
4. Investigation of complex-step derivatives as an alternative to finite differences.
5. Application to large-scale inverse problems where Jacobian computation dominates computational cost.

## Appendix A. Technical Proofs

### Appendix A.1. Proof of Theorem 4.2 (Extended)

**Proof.** Let  $e_n = x_n - x^*$ . Taylor expansion gives:

$$\begin{aligned} f(x_n) &= f'(x^*)e_n + \frac{1}{2}f''(x^*)e_n^2 + \mathcal{O}(e_n^3), \\ f'(x_n) &= f'(x^*) + f''(x^*)e_n + \mathcal{O}(e_n^2). \end{aligned}$$

The approximate derivative satisfies:

$$\tilde{f}'(x_n) = f'(x_n) + \epsilon_n = f'(x^*) + f''(x^*)e_n + \epsilon_n + \mathcal{O}(e_n^2),$$

where  $|\epsilon_n| \leq \delta$ .

The Newton update gives:

$$\begin{aligned} e_{n+1} &= e_n - \frac{f(x_n)}{\tilde{f}'(x_n)} \\ &= e_n - \frac{f'(x^*)e_n + \frac{1}{2}f''(x^*)e_n^2 + \mathcal{O}(e_n^3)}{f'(x^*) + f''(x^*)e_n + \epsilon_n + \mathcal{O}(e_n^2)} \\ &= e_n - \frac{f'(x^*)e_n}{f'(x^*) + \epsilon_n} \left[ 1 + \frac{\frac{1}{2}f''(x^*)e_n}{f'(x^*)} - \frac{f''(x^*)e_n + \epsilon_n}{f'(x^*) + \epsilon_n} + \mathcal{O}(e_n^2) \right] \\ &= \left( 1 - \frac{f'(x^*)}{f'(x^*) + \epsilon_n} \right) e_n + \frac{f''(x^*)}{2(f'(x^*) + \epsilon_n)} e_n^2 + \mathcal{O}(e_n^3). \end{aligned}$$

Thus:

$$|e_{n+1}| \leq \frac{|\epsilon_n|}{|f'(x^*)|} |e_n| + \frac{|f''(x^*)|}{2|f'(x^*)|} |e_n|^2 + \mathcal{O}(|e_n|^3),$$

where we used  $|f'(x^*) + \epsilon_n| \geq |f'(x^*)| - \delta > 0$  for sufficiently small  $\delta$ .  $\square$

## Appendix B. Additional Numerical Results

Table A1. Effect of finite difference order on convergence.

Method	Iterations	Final Error	Func. Evals	Success Rate
Exact Newton	8	$2.1 \times 10^{-16}$	16	90%
Forward Diff ( $p = 1$ )	12	$3.4 \times 10^{-10}$	24	98%
Central Diff ( $p = 2$ )	10	$5.6 \times 10^{-12}$	30	96%
Fourth-order ( $p = 4$ )	9	$7.8 \times 10^{-14}$	45	94%

Higher-order finite differences reduce iteration count but increase function evaluations per iteration. The optimal choice depends on the relative cost of function evaluations versus iterations.

## References

1. R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed., Brooks/Cole, 2011.
2. P. Deuffhard, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, Springer, 2011.
3. A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, 2nd ed., Springer, 2007.
4. B. Fornberg, "Generation of Finite Difference Formulas on Arbitrarily Spaced Grids," *Mathematics of Computation*, vol. 51, no. 184, pp. 699–706, 1988.
5. N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, 2002.
6. R. S. Dembo, S. C. Eisenstat, and T. Steihaug, "Inexact Newton Methods," *SIAM Journal on Numerical Analysis*, vol. 19, no. 2, pp. 400–408, 1982.
7. J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, 2006.
8. C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, 1995.
9. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed., Cambridge University Press, 2007.
10. H. Robbins and S. Monro, "A Stochastic Approximation Method," *Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.
11. J. Dennis and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, 1996.
12. K. Atkinson and W. Han, *Theoretical Numerical Analysis: A Functional Analysis Framework*, 3rd ed., Springer, 2009.
13. A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed., SIAM, 2008.
14. A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, SIAM, 2000.
15. C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, SIAM, 2003.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.