FACULTY OF LAW, ECONOMICS AND FINANCE

Department of Economics and Management

Discussion Paper

2025-09

Economics

Department of Economics and Management
University of Luxembourg

Step size selection in numerical differences using a regression kink

https://www.uni.lu/fdef-en/research-departments/department-of-economics-and-management/publications/

Andreï Kostyrka, DEM, Université du Luxembourg, UL

May 2025

For editorial correspondence, please contact: dem@uni.lu
University of Luxembourg
Faculty of Law, Economics and Finance
4, rue Alphonse Weicker
L-2721 Luxembourg

Step size selection in numerical differences using a regression kink

Andreï Victorovitch KOSTYRKA
Department of Economics and Management, University of Luxembourg*

14th May 2025

Abstract

We propose a new step-size selection procedure for numerical differences based on fitting a piecewise linear shape to the observed estimate of truncation error and determining the position of its kink. The novelty of this method is in its use of the full information about the estimated total error behaviour at both sides around the optimum and in the incorporation of robust statistical tools for estimating the best V-shaped fit. The added safety checks ensure that the kink is detected if it exists, or a reasonable step size is returned in the case there is no kink. In numerical simulations, the proposed method algorithm outperforms two existing algorithms in terms of median error when tested on 5 well-behaved and 3 pathological functions.

Keywords: numerical differentiation, error analysis, optimal step size, floating-point arithmetic, finite differences

MSC: 65D25, 65G50, 41A55, 41A58

1 Introduction

Overview. Let $f : \mathbb{R} \mapsto \mathbb{R}$ be a sufficiently smooth function and let $x_0 \in \mathbb{R}$ be the point at which its derivative is required. Finite differences remain the most common approach: the central difference $\hat{f}_{\mathrm{CD},2}(x_0, h) := [-0.5f(x_0 - h) + 0.5f(x_0 + h)]/h$ approximates $f'(x_0)$ with a truncation error of $O(h^2)$. Lyness and Moler (1967) first made explicit, and Fornberg (1988) generalised the idea that derivatives of *any* order can be expressed, up to $O(h^a)$, as a weighted sum of function values on a large-enough and possibly non-uniform stencil $\vec{b} = \{b_i\}_{i=1}^n$, namely,

$$f^{(d)}(x) = h^{-d}\,\vec{w}^{\intercal}\vec{f}(x + \vec{b}h) + O(h^a) = h^{-d}\sum_{i=1}^{n} w_i f(x + b_i h) + O(h^a), \tag{1.1}$$

where d is the derivative order, a is the formal accuracy order, and $\vec{w} = \vec{w}(\vec{b}, d)$ the corresponding finite-difference coefficients obtained, in practice, by solving a Vandermonde system, preferably using a numerically stable algorithm (Björck & Pereyra, 1970). These weights ensure that in the Taylor expansion of $h^{-d}\sum_i w_i f(x + b_i h)$ around x, derivatives of orders 0 through $n-1$ cancel out except for $f^{(d)}(x)$, which has a coefficient of unity.

The only undefined variable in (1.1) is the step size h – the scaling factor that determines how finely or coarsely the function is sampled on a grid, and choosing h is a balancing act. Too large a step results in poor analytical approximation due to the presence of higher-order terms in the Taylor expansion. Steps too small will cause machine-rounding errors to swamp the signal: every

*We thank Gautam Tripathi for his helpful comments. We also thank seminar participants at the University of Luxembourg for their insightful suggestions.

evaluation, addition, and division in (1.1) incurs a loss of precision. Therefore, there should be a step size or a range of step sizes that yield the most accurate approximation of $f^{(d)}(x_0)$ in practice. Finding this step, however, is non-trivial because the truncation error is unobservable (but can be approximated), whereas the rounding error is essentially unpredictable: it is virtually impossible to correctly guess the nearest representable number that will be used to store each evaluation result. An algorithm is thus required to predict the behaviour of the truncation and rounding components of the total approximation error and to locate the step that minimises their sum.

History. The history of step-size selection methods is long yet relatively sparse. Early research on numerical approximation was focused on polynomial interpolation (Aitken, 1932; Jordan, 1928), notably Everett's even-difference scheme (Everett, 1901). The earliest explicit proposal to use finite differences to approximate derivatives appears in Aitken (1938). Subsequent studies addressed uniform stencils (Keller & Pereyra, 1978), arbitrary stencils (Oliver, 1980; Oliver & Ruffhead, 1975), and, ultimately, derivatives of any order on general stencils (Fornberg, 1988; Lakin, 1986). An algorithm that improves accuracy by re-combining central differences with different step sizes via Richardson extrapolation is due to Ridders (1982).

Despite huge advancements in the literature on interpolation and approximation, systematic step-size selection has attracted surprisingly little attention. Notable contributions include:

1. Optimal truncation–round-off ratio by Curtis and Reid (1974);
2. Theoretically optimal formula based on estimated third derivatives by Dumontet (1973) and Dumontet and Vignes (1977);
3. Monotonicity checks on the estimated truncation error by Stepleman and Winarsky (1979);
4. Acceptable range of relative condition errors by Gill et al. (1983) and its adaptive dynamic extension by Barton (1992);
5. Non-uniform 3-point spacing for $O(h^3)$ accuracy by Prentice (2011);
6. Safeguarded search of truncation-error slope deviation by Mathur (2012, 2013).
7. Parameter-shift estimator for quantum hardware by Mari et al. (2021).

Remarkably, there is even less research on the optimal step size for Jacobians and mixed partials. Curtis and Reid (1974) mention that the step-size search for vector functions may be coordinate-wise, albeit 'wasteful'. Mathur (2012) proposes a solution to cut down the number of evaluations to choose a step size for a multi-variate problem. General cross-derivatives, typically used as Hessian entries, are rarely scrutinised because Hessians are frequently approximated via computationally lightweight techniques (e. g. outer-product rules or quasi-Newton updates). In cases where Hessians themselves are of interest, like inferential methods in statistics, they can be computed as the Jacobian of a gradient via repeated differencing, which is why the default behaviour in software is often repeated differencing with a rule-of-thumb step size. The consequences of such an approach can be dire, such as negative estimated variances.

Illustration. The motivating example comes from the field of quantitative finance. Consider fitting an AR(1)-GARCH(1,1) model (Bollerslev, 1986) to the daily logarithmic returns $r_t \coloneqq \log(P_t/P_{t-1})$ specified as $r_t \coloneqq \mu + \rho r_{t-1} + U_t$, where μ is the mean return, ρ is the auto-regressive coefficient and U_t is the zero-mean shock. This shock is assumed to be conditionally heteroskedastic with variance equation $\sigma_t^2 \coloneqq \omega + \alpha U_{t-1}^2 + \beta\sigma_{t-1}^2$. Estimating this model via maximum likelihood with Gaussian errors on a sample of 1164 observations (1990-03-27 – 1994-09-09) using the R package `rugarch` yields convergence declared by the solver, yet the inverse of the negative of the numerically computed Hessian matrix contains a negative diagonal entry – an impossible negative variance. Other packages produce standard errors for the same model and the same data that differ by factors of three to four, drastically altering significance test results and confidence regions. The root cause of these disparate results is the differences in the step sizes that each package uses when finite-differencing the log-likelihood to approximate its Hessian.

We focus on the problem of step-size selection for a scalar-valued function. The phenomenon that sparked this investigation was the poor empirical behaviour of step-size selection algorithms

where the function of interest was either truncation-error-free or near-symmetric around the point of interest (local minima of M-estimators). In these cases, the behaviour of the approximation error might be erratic or even multi-modal, causing failures of many step-size algorithms.

Contribution. We propose a novel step-size selection algorithm that is based on an optimisation procedure with well-defined termination criteria and several adequacy checks. So far, there has been very little application of statistical methods to step-size selection for numerical derivatives, if any. The reason is, treating each function evaluation as a statistical observation – described by a conditional distribution and contaminated by numerical noise – is possible only if the number of function evaluations is large enough for statistical approximations to be applicable. Previous step-size selection methods have relied on checking certain local properties of approximated errors, such as signs of differences or ratios within tolerances, as opposed to analysing the global shape of the error function using dozens of evaluations. Yet as technology marches on and accuracy is preferred to speed, in settings with a moderate-to-large number of parameters, obtaining multiple evaluations of a reasonably fast function is a necessity for a researcher to have data-backed evidence that their numerical derivatives attain acceptable accuracy. Therefore, the novelty of this proposed method is in the use of the global information on function behaviour across a wide range of step sizes rather than local slope conditions at a single point.

Notation. We use $\tilde{x} = \log_2 x$ to denote the binary logarithm of x and $\mathtt{t}, \mathtt{r}, \mathtt{c}$ for truncation, rounding, and combined error moduli, respectively.

Structure. The rest of this paper is organised as follows. Section 2 develops a practical error model for finite-difference derivative approximations based on truncation and rounding components, and shows how the combined error can be estimated from data. Section 3 introduces the novel kink-based step-size selection algorithm, formalises its optimisation framework, and discusses diagnostic safeguards for edge cases. Section 4 benchmarks the new algorithm against established ones on a suite of test functions, quantifying accuracy, robustness, and computational cost. Section 5 summarises the main findings, outlines practical recommendations, and highlights avenues for future research.

2 Error analysis

2.1 Truncation error

Consider an a^{th}-order-accurate central difference which approximates the d^{th} derivative of f at x_0 on a symmetric stencil $\vec{b} = \{b_i\}_{i=1}^n$ of length $n = a + d - 1$. Using the previously defined notation $\vec{w}$ for the corresponding coefficients, we have $f^{(d)} = h^{-d}\,\vec{w}^\intercal \vec{f}(x + \vec{b}h) + O(h^a) = f^{(d)}_{\text{CD},a} + O(h^a)$, where the constant hidden in the $O(\cdot)$ term is governed by the higher-order derivatives:[1] $O(h^a) = \sum_i c_i f^{(d+a)}(x + \eta_i \alpha_i h) h^a$, where $\alpha_i \in [0,1]$, η_i depend on the chosen stencil, and c_i depend on $\vec{w}$. Yet it is more convenient to approximate the upper bound of the error modulus for small h using the triangle inequality:

$$|O(h^a)| \le \Big(\sum_i |c_i|\Big) \max_{\alpha\in[0,1],\eta\in\vec{b}} |f^{(d+a)}(x + \eta\alpha h)| \approx \Big(\sum_i |c_i|\Big) |f^{(d+a)}(x)|$$

because h is small and $f^{(d+a)}$ is assumed to be continuous.

To illustrate the point, we derive the modulus of approximate truncation error $\mathtt{t}(x,h)$ of $f'_{\text{CD},2}(x,h)$, $f''_{\text{CD},2}(x,h)$, and $f'_{\text{CD},4}(x,h)$. Since $f'_{\text{CD},2}(x,h) = f'(x) + \frac{f'''(x+\alpha h)}{6} h^2$ for some $\alpha \in [-1,1]$, the modulus of the truncation error is $\frac{|f'''(x+\alpha h)|}{6} h^2 \approx \frac{|f'''(x)|}{6} h^2$. Similar calculations show that

[1] A Taylor expansion around x_0 gives the largest neglected term explicitly. For a two-point grid $\{x_0 \pm h\}$, $O(h^a) = f^{(d+a)}(x_0 + \alpha h) h^a / (d+a)! \cdot \vec{w}^\intercal \vec{b}^n$ with $\alpha \in [-1,1]$. Coefficients for long symmetric stencils usually alternate in sign and the generalised mean-value theorem does not yield a single term for the Lagrange remainder.

$\mathtt{t}(x,h)$ for central-difference-based second derivatives, computed as $\frac{f(x+h)-2f(x)+f(x-h)}{h^2}$, is approximately $\frac{|f^{(\mathrm{IV})}(x)|}{12}h^2$. For $f'_{\mathrm{CD},4}$ computed using the stencil $\vec{b}=(-2,-1,1,2)$ and coefficients $\vec{w}=(1/12,-2/3,2/3,-1/12)$, $\mathtt{t}(h)=\frac{|f^{(\mathrm{V})}(x)|}{30}h^4$, and using the uniformly spaced stencil $(-3,-1,1,3)$ with weights $(1/48,-9/16,9/16,-1/48)$, $\mathtt{t}(h)=\frac{3|f^{(\mathrm{V})}(x)|}{40}h^4$.

These higher-order derivatives in the truncation error can be estimated via finite differences, too, by augmenting the existing stencil with a additional points, evaluating f on them, and applying (1.1).[2] For simple central differences computed with $f(x\pm h)$, adding $f(x\pm 2h)$ and using the stencil $(-2,-1,1,2)$ with weights $(-1/2,1,-1,1/2)$ yields an estimate of $f'''_{\mathrm{CD},2}(x)$, a second-order-accurate approximation of the third derivative.

An alternative estimator of $\mathtt{t}(x,h)$ – going back to Curtis and Reid (1974) – is to subtract a higher-order-accurate derivative from a low-order-accurate one, so that the main derivative cancels out, and the difference is dominated by the leading truncation term. Since $f_{\mathrm{CD},4}(x,h)=f'(x)+O(h^4)$, the truncation error can be estimated as $\mathtt{t}(x,h)=|f_{\mathrm{CD},2}(x,h)-f_{\mathrm{CD},4}(x,h)|=\frac{|f'''(x)|}{6}h^2+O(h^4)$

Yet another equivalent estimation method is Richardson extrapolation. A linear combination of $f'_{\mathrm{CD},2}(x,h)$ and $f'_{\mathrm{CD},2}(x,2h)$ can be used to calculate $f'_{\mathrm{CD},4}(x,h)$ and infer the truncation error therefrom. If the same stencil is used to compute these three estimators of the truncation error – direct weighted summation, difference of two derivatives, and Richardson extrapolation – then, all three are algebraically identical.

Key fact 1. The coefficients $\vec{w}$ annihilate only the Taylor terms except the d^{th} derivative up until $f^{(d+a-1)}$. The coefficient on h^a in the Taylor expansion is therefore generally non-zero and equal in absolute value to at most $\mathtt{t}(x,h):=|\vec{w}^{\mathsf{T}}\vec{b}^{\,n}f^{(d+a)}(x)|/(d+a)!$. A more conservative bound for it would be $\|\vec{w}\odot\vec{b}\|_1|f^{(d+a)}(x)|/(d+a)!\cdot h^a$.[3]

Key fact 2. Since $\mathtt{t}(x,h)$ can be written as $c_{\mathtt{t}}(x)h^a$, halving h multiplies $\mathtt{t}$ by 2^{-a}. Hence, taking the binary logarithm of both sides yields $\tilde{\mathtt{t}}(x,\tilde{h})=\tilde{c}_{\mathtt{t}}(x)+a\tilde{h}$; halving the step size reduces the truncation error by a factor of 2^a. In log–log coordinates, the truncation error therefore traces an almost-straight line with slope a. We label the interval of h values over which this behaviour is observed as the *'valid truncation range'*, and the interval in which $f^{(d+a)}(x+\alpha h)\not\approx f^{(d+a)}(x)$ for $\alpha\in[-1,1]$ because h is large, making the slope too different from a, the *'invalid truncation range'*. If the function at x_0 is pathological in the sense that $f^{(d+a)}(x_0)=0$, then, the slope of the line will be equal to the power in the next non-zero Taylor term.

2.2 Rounding error

On any computer with finite memory, most operations, including function evaluations, are lossy. On modern 64-bit machines, the loss is usually small enough to be neglected in most empirical applications, but in numerical derivation, the error due to machine rounding may become arbitrarily large.

A central difference $(-0.5f(x-h)+0.5f(x+h))/h$ is evaluated on a computer as follows:

1. If x and h differ by more than one order of magnitude, their difference may be inexact (by Sterbenz lemma): in general, $\widehat{x-h}\neq x-h$.[4]
2. The function evaluation $\hat{f}(x-h)$ is noisy and not equal to $f(x-h)$ because rounding towards the nearest representable number takes place. Some functions, usually outputs of numerical routines, are not accurate to the last bit. Denote $\mathrm{p}:=|(\hat{f}(x)-f(x))/f(x)|$ the maximum relative

[2] Reusing points for computing different derivatives is theoretically valid, yet in practice, higher derivation or accuracy orders require a larger step size for adequate accuracy, i. e. a sparser grid.

[3] Here, $\odot$ denotes the Hadamard (element-wise product), and $\|\cdot\|_1$ denotes the L_1 norm – the sum of absolute values.

[4] In practice, to avoid this representation error in the argument, it is useful to compute $\widehat{x-h}$, $\hat{h}=x-\widehat{x-h}$, and use $x\pm\hat{h}$ because this pair of points will be symmetric in machine memory around x by construction.

error. If f is fully accurate, then $\mathtt{p} = \epsilon_\mathrm{m}/2$, where ϵ_m is the interval machine epsilon (2^{-52} in IEEE-754 double precision).
3. The subtraction $\hat{f}(x+h) - \hat{f}(x-h)$ combines the two function-evaluation errors.
4. The result is divided by a small number, h^d, which amplifies all previous errors.

To put this amplification into perspective: if an intuitive value $h = 10^{-8}$ is used to compute $f''_{\mathrm{CD},2}(x,h)$, even if f has full precision, the function-evaluation error – at most $|f(x)|\epsilon_\mathrm{m}/2$ – is multiplied by 10^{16}, resulting in a maximum absolute error of the same magnitude as $|f(x)|$!

Following the majority of papers in the numerical-analysis literature, we ignore the operations (1) and (3), which usually contribute far less, and focus on (2) and (4) to bound the rounding error. Denoting $\overline{f} \coloneqq \max_i |\hat{f}(x+b_i h)|$,[5] bounding the relative evaluation error at $\mathtt{p} \geq \epsilon_\mathrm{m}/2$, and ignoring the errors stemming from arithmetic operations, we obtain the conservative bound

$$h^{-d} \sum_{i=1}^{n} w_i \hat{f}(x+b_i h) \leq \overline{f}(x) \cdot \mathtt{p} h^{-d} \|\vec{w}\|_1 \coloneqq \mathtt{r}(x,h).$$

For the common cases from the previous section, this gives the relative error of $\hat{f}'_{\mathrm{CD},2}(x,h)$ at most $\mathtt{p}/h$, the relative error of $\hat{f}''_{\mathrm{CD},2}(x,h)$ at most $4\mathtt{p}/h^2$, and the relative error of $\hat{f}'_{\mathrm{CD},4}(x,h)$ at most $1.5\mathtt{p}/h$.

Although it may seem that the rounding-error bound is a decreasing function of h, if f grows faster than x^d, then, the rounding error is bound to eventually grow with respect to h. This fact does not complicate the analysis, nor does it have to be explicitly taken into account because the combined error described in the next section is agnostic about this property of $\mathtt{r}(x,h)$. It simply explains the discrepancy in behaviour of $\mathtt{r}(x,h)$ between functions with different growth rates.

Key fact 3. With the rounding error $\mathtt{r}(x,h) \approx c_\mathtt{r}(x)h^{-d}$, halving the step multiplies the round-off error by 2^d. Taking the binary logarithm yields $\tilde{\mathtt{r}}(x,\tilde{h}) = \tilde{c}_\mathtt{r}(x) + d\tilde{h}$. Therefore, in log-log axes, the rounding error for a range of small step sizes – which we label *'rounding range'* – appears as a dispersed cloud with slope $-d$ (since the actual relative rounding error of one function evaluation is distributed almost uniformly on $[-\mathtt{p},\mathtt{p}]$).

2.3 Combined error and its minimisation

We established that the total numerical error of finite-difference approximations consists of two primary contributions: the truncation error, arising from the omitted higher-order terms in Taylor expansions, and the rounding error, resulting from finite numerical precision. Under the approximations presented earlier, the combined error is defined as

$$\mathtt{c}(x,h) \coloneqq \mathtt{t}(h) + \mathtt{r}(h) = |\vec{w}^\mathsf{T}\vec{b}^n| \frac{|f^{(d+a)}(x)|}{(d+a)!} h^a + \frac{\mathtt{p}\|\vec{w}\|}{h^d} |\overline{f}(x)| = c_\mathtt{t}(x)h^a + c_\mathtt{r}(x)h^{-d}. \tag{2.1}$$

Minimising the combined error analytically involves solving the first-order optimality condition, $ac_t(x)h^{a-1} = dc_r(x)h^{-d-1}$, which occurs at $h^*(x) = \sqrt[a+d]{(dc_r(x))/(ac_t(x))}$. This expression is widely adopted as an initial step-size estimate in practice. For the aforementioned common cases, once $f^{(d+a)}$ has been approximated, the plug-in optimal step sizes are calculated as $h^*_{\mathrm{CD},2}(x) = \sqrt[3]{3\mathtt{p}f'(x)/f'''(x)}$, $h^{**}_{\mathrm{CD},2}(x) = \sqrt[4]{48\cdot\mathtt{p}f'(x)/f^{(\mathrm{IV})}(x)}$, and $h^*_{\mathrm{CD},4}(x) = \sqrt[5]{11.25\mathtt{p}f'(x)/f^{(\mathrm{V})}(x)}$. At these optimal steps, the truncation-to-rounding error ratios, $\mathtt{t}(x,h^*)/\mathtt{r}(x,h^*)$, are equal to $1/2$, 1, and $1/4$ (more generally, d/a) – this fact will be used in obtaining a correction for the step size.

Although $\mathtt{c}(h)$ itself does not simplify linearly under a logarithmic transformation, its asymptotic branches – truncation, when $h \ll h*$, and rounding, when $h \gg h^*$ – do exhibit linear behaviour in

[5] In the literature, it is sometimes assumed for simplicity that $\forall i,\ f(x+b_i h) \approx f(x)$ because h is small and f is assumed to be sufficiently smooth.

double-logarithmic axes:

$$\tilde{\mathtt{c}}(\tilde{h}) \approx \begin{cases} \tilde{c}_{\mathtt{r}}(x) - d\tilde{h}, & h < h^* \text{ (rounding branch)}, \\ \tilde{c}_{\mathtt{t}}(x) + a\tilde{h}, & h \geq h^* \text{ (truncation branch)}. \end{cases} \tag{2.2}$$

Thus, the empirical log-log plot of the estimated combined error typically resembles a 'V' shape, facilitating estimation of h^* without explicitly observing each error component separately.

There are several special cases in which this approximation is inaccurate.

1. *Local symmetry.* If $|f(x)|$ is symmetric around x_0, all higher-order derivatives of odd order vanish, eliminating the truncation branch. *Example:* $|x|^p$ for $p \geq 0$.
2. *Low-degree polynomials.* If $f(x)$ is a low-degree (less than $d+a$) polynomial – the truncation error vanished entirely, leaving only numeric noise. *Example:* x^2 at $x_0 = 1$.
3. *Vanishing leading-order derivatives.* If $f^{(d+a)}(x_0) = 0$, the first non-zero truncation term appears at a higher power of h, altering the truncation slope. *Example:* $x^3 + 1/x$ at $x_0 = 1$ has a truncation-branch slope of 4 instead of 2 (in this case, $a+2$ instead of a).

Additionally, if the combined error is always increasing for a wide range of h, it does not necessarily indicate that the rounding component is zero – it may result from a faster growth of $|f(x+h)|$ compared to the decay of h^{-d}.

Empirical estimation of $\mathtt{c}(x,h)$ proceeds as follows. For the chosen x_0, use a geometric sequence $h \in \{h_0, r\,h_0, r^2 h_0, \ldots\}$ for $r > 1$, and a sufficiently large ($n = d + a + 1$) symmetric stencil $\vec{b}$ to estimate $\mathtt{t}(x, h_i)$ at $x = x_0$ via finite differences by computing $|f'''_{\mathrm{CD},2}(x_0)|$ and multiplying it by $|\vec{w}^\top \vec{b}^n|/(d+a)! \cdot h^a$.[6] This will produce a near-linear sequence of slopes in the valid truncation range and erratic behaviour in the invalid truncation range. As $\mathtt{t}(h) \xrightarrow{h\to 0} 0$, realisations of the practically unpredictable $\mathtt{r}(h)$ will appear in the left branch, bounded from above by $\mathtt{p}|f(x)|h^{-d}$.[7] The lowest value that $\mathtt{r}(h)$ can take is zero, which is uninformative for error analysis because the combined error should be analysed in double logarithmic axes; nevertheless, with a high probability, at least several step sizes will produce a non-zero rounding error.

In the most common case – obtaining the truncation error of $f'_{\mathrm{CD},2}(x_0,h)$ – the estimated $f'''_{\mathrm{CD},2}(x_0,h)$, when divided by 6, yields $\mathtt{t}(x,h)$, although it is recommended to use a more accurate approximation $f'''_{\mathrm{CD},4}(x_0,h)$.[8]

Example 1. If $f(x) = \sin(x)$ and $x_0 = 1$, a sequence of doubling step sizes $\{h_i\}$ between 2^{-46} and 2^{14} yields the values of $f'''_{\mathrm{CD},4}(x_0,h)$ such that $\frac{|f'''_{\mathrm{CD},2}(x,h)|}{6}h^2$ begins convincingly increasing at the expected rate, a, around the value of $h_1 = 2^{-17}$. The invalid truncation range starts at $h = 2$. (Figure 1, left panel.)

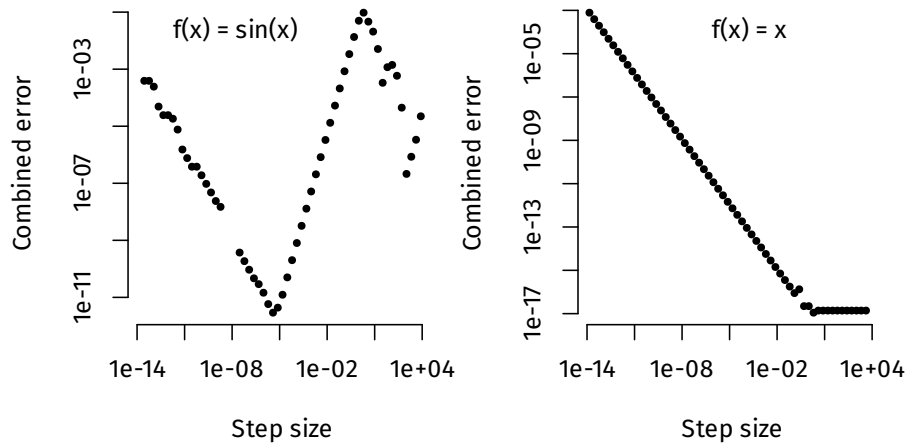
Example 2. If $f(x) = x^2$ and $x_0 = 0$, then, regardless of the step size, $f'''_{\mathrm{CD},2}(x,h) \equiv 0$, and due to symmetry, the estimated truncation error is always 0. Therefore, no global minimum of $\mathtt{c}(h)$ exists, and the step size should be chosen in such a manner that $\mathtt{r}(x_0,h)$ be reasonably small but not too small (to prevent impractically large steps).

Example 3. If $f(x) = x$ and $x_0 = 1$, then, the rounding branch has a slope of $-d$, but the truncation branch is flat because $c_{\mathtt{r}}(x)/h \cdot h$ is constant – hence the need to resort to a fail-safe method because the minimiser of the observed combined error is not necessarily a reasonable step-size value. (Figure 1, right panel.)

[6] Asymmetric stencils generally require one more call than symmetric ones to achieve the same order of accuracy, but we do not consider them here due to their generally inferior properties.

[7] Since $\mathtt{p}$ is the *maximum* absolute relative error, the *average* absolute relative error is marginally smaller, but this discrepancy can be safely ignored in practice.

[8] In practice, the larger stencil $(-4,-2,-1,1,2,4)$ with weights $(1/48, -17/24, 4/3, -4/3, 17/24, -1/48)$ is highly recommended for $f'''_{\mathrm{CD},4}(x,h)$ because a sum with more terms achieves two goals: produce a more reliable approximation of the rounding branch and reduce the probability that $\mathtt{r}(x,h)$ is exactly zero. The author has encountered cases where the rounding branch had no non-zero points with a 4-term $f'''_{\mathrm{CD},2}$, but contained more than 10 non-zero points with a 6-term $f'''_{\mathrm{CD},4}$.



Both panels show the estimated combined error at $x_0 = 1$.

Figure 1: Estimated combined error

A local minimum of the combined error should exist to the left of the valid truncation range if both $\mathtt{t}(x,h)$ and $\mathtt{r}(x,h)$ are well-behaved. Define the V-shape function – a piecewise linear function with slopes $-d$ and a – as

$$V(\tilde{h};\beta,\gamma) := \begin{cases} \beta - d(\tilde{h}-\gamma), & \tilde{h} < \gamma, \\ \beta + a(\tilde{h}-\gamma), & \tilde{h} \geq \gamma \end{cases} \tag{2.3}$$

Thus, minimising $\mathtt{c}(x,h)$ w. r. t. h in practice means minimising the discrepancy between this V-shape and the observed combined error over (β,γ) and taking the optimal uncorrected step size $h^\dagger := 2^{\hat{\gamma}}$. For most common functions, excluding the cases discussed in Section 3.2, this approach recovers the near-optimal value of the step size. One advantage of this method over the methods of Stepleman and Winarsky (1979) and Mathur (2012) is its robustness to locally non-monotone slopes of the right branch of noisy functions. A piecewise linear fit captures the overall shape, ignoring the possible sporadic fluctuations, and even allows statistical hypotheses to be tested about the equality of the rounding-branch slope to $-m$ and the truncation-branch slope to a.

Because the fitted kink location $h^\dagger$ equalises estimated truncation and rounding errors, it is necessary to multiplicatively correct $h^\dagger$ to obtain the true optimal step size h^*. At the minimum of $\mathtt{c}(x,h)$, the ratio $\mathtt{t}(x,h^*)/\mathtt{r}(x,h^*)$ is equal to d/a. Hence, $\mathtt{t}(h^*) = \frac{d}{a}\mathtt{r}(h^*) \iff h^* = h^\dagger \cdot \sqrt[d+a]{d/a}$. For $f'_{\mathrm{CD},2}$, $f''_{\mathrm{CD},2}$, $f'_{\mathrm{CD},4}$, these ratios are $1/2$, 1, and $1/4$; therefore, $h^\dagger$ should be multiplied by $\sqrt[3]{1/2}$, 1 (no adjustment for $f''_{\mathrm{CD},2}$), and $\sqrt[5]{1/4}$, respectively.

3 Kink-based step-size selection

3.1 Algorithm steps

Given the function f, derivative order d, desired accuracy order for $f^{(d)}$ denoted as a, chosen accuracy order for $f^{(d+a)}$ denoted as a', and the evaluation point x_0,

1. Select an appropriate initial step size h_0 and define an ample search range $[h_{\min}, h_{\max}]$ around it. Create a grid of candidate step sizes forming a geometric progression $\{h_{\min}, r\,h_{\min}, \ldots, h_{\max}/r, h_{\max}\}$ with common ratio $r > 1$.
   Suggested values: $h_0 = \sqrt[m+a]{\epsilon_{\mathrm{m}}}$, $h_{\min} = h_0 \cdot 2^{-24}$, $h_{\max} = h_0 \cdot 2^{24}$, $r = 2$ for functions with high precision and $r = \sqrt{2}$ for noisy functions.
2. For each candidate step h_i, generate a symmetric evaluation grid $\{x_0 + b_j h_i\}$ using a sufficiently long ($n \geq d + a' + 1$) symmetric stencil of the form $\{b_j\}_{j=1}^n = \{0, \pm r, \pm r^2, \ldots\}$. Exclude

the zero is $d+a'$ is odd. Compute the values $\{f(x_0+b_j h_i)\}$, preferably in parallel, and obtain $f^{(d+a)}_{\mathrm{CD},a'}(x_0,h_i)$ by summing them with the corresponding finite-difference coefficients $\vec{w}$.

3. Compute the combined error estimates $\mathtt{c}(x_0,h_i)$ as $|f^{(d+a)}_{\mathrm{CD},a'}(x_0,h_i)|\cdot|\vec{w}^\top\vec{b}^n|/(d+a)!\cdot h^a$. If an estimate is exactly zero, recompute with a different accuracy order $a'\pm 2$. If multiple accuracy orders yield zero, mark these points as `NA`/`NaN`.
4. Compute binary logarithms, $\tilde{\mathtt{c}}(x_0,\tilde{h}_i)$. Determine the existence of a truncation branch by estimating slopes via computing the between-point slopes of the combined error: $s_i \coloneqq [\tilde{\mathtt{c}}(x_0,\tilde{h}_{i+1})-\tilde{\mathtt{c}}(x_0,\tilde{h}_{i-1})]/(\tilde{h}_{i+1}-\tilde{h}_{i-1})$, for $i=2,\dots,n-1$. Define the *valid-truncation-error* range as the largest consecutive set of indices at which the relative slope deviation from a is less than 10%: $\mathscr{A}\coloneqq\{i\colon |s_i-a|/a<0.1\}$. If 3 or more points satisfy this criterion, the search for the V-shape may continue on the range of indices $i=1,\dots,\max\mathscr{A}$; otherwise, invoke the *'Zero truncation error'* fall-back described in the next sub-section.
5. Optionally, visually inspect the plot of $\tilde{c}(\tilde{h})$ versus $\tilde{h}$. Absence of a clear V-shape suggests invoking the *'Zero truncation error'* fall-back or selecting the step visually.
6. Fit the piecewise-linear V-shaped model defined in (2.3). Define the robust pseudo-Huber loss function as $\ell_\rho(x)\coloneqq\rho^2(\sqrt{1+(x/\rho)^2}-1)$. Let the knee size ρ be the median absolute deviation of the initial residuals $\tilde{\mathtt{c}}(x_0,\tilde{h}_i)-V(\tilde{h}_i;\check{\beta},\check{\gamma})$.[9] Define the objective function as the aggregated loss applied to the discrepancies between $\tilde{c}(\tilde{h}_i)$ and the fitted check exclusively in the rounding and valid truncation ranges, and minimise it numerically:

$$(\hat{\beta},\hat{\gamma})\coloneqq\operatorname*{arg\,min}_{\beta,\gamma\in\mathbb{R}}\sum_{i=1}^{\max\mathscr{A}}\ell_\rho(\tilde{c}(\tilde{h}_i)-V(\tilde{h}_i;\beta,\gamma))$$

   Optimisation can be box-constrained: $\gamma\in[\tilde{h}_{\min},\tilde{h}_{\max\mathscr{A}}]$ and $\beta\in[\min_i\tilde{c}(\tilde{h}_i)-a,(\min_i\tilde{c}(\tilde{h}_i)+\max_i\tilde{c}(\tilde{h}_i))/2]$.
   Suggested values: $\check{\gamma}\coloneqq\arg\min_{\tilde{h}\in\{\tilde{h}_i\}_{i=1}^n}\tilde{c}(\tilde{h})$ and $\check{\beta}\coloneqq\tilde{c}(\check{\gamma})$.
7. Finally, correct the fitted step size $h^\dagger=2^{\hat{\gamma}}$ by adjusting to match the theoretical truncation-to-rounding error ratio of d/a: $h^*\coloneqq h^\dagger\cdot\sqrt[d+a]{d/a}$.

For gradients, this procedure needs to be repeated for each coordinate of x, keeping other coordinated fixed, yielding a vector of step sizes.

3.2 Edge cases and solutions for them

Zero truncation error. Polynomials of degree up to $(d+a-1)$ yield zero or numerically negligible truncation errors, making a logarithmic transformation impossible. This can be detected with two checks. Firstly, if the median of the combined error falls below the numerical precision bound $\mathtt{p}$, the truncation error is too small to measure. Secondly, if fewer than three non-zero points $\mathtt{c}(h_i)$ exist, the truncation error may be deemed zero. In such scenarios, employ a rounding-error–only approximation: compute $\mathtt{r}(h)$ for the entire grid of step sizes and choose such h° that $\mathtt{r}(h^\circ)$ be close to the rounding error minimum for well-defined functions: $\mathtt{r}(h^*)=c_r((dc_r)/(ac_t))^{-d/(a+d)}$. For $f'_{\mathrm{CD},2}$, $f''_{\mathrm{CD},2}$ and $f'_{\mathrm{CD},4}$, these target values $\mathtt{r}(h^*)$ are approximately equal to $0.693\sqrt[3]{\mathtt{p}^2 f(x)^2 f'''(x)}$, $0.577\sqrt{\mathtt{p}f(x)f^{(\mathrm{IV})}(x)}$, and $0.924\sqrt[5]{\mathtt{p}^4 f(x)^4 f^{(\mathrm{V})}(x)}$. Compute the higher-order derivatives from the evaluation grid and replace any exact zeros with $\mathtt{p}$. This fail-safe method ensures that the step size is not too large because it requires that the rounding error should not be smaller than that of the theoretically optimal estimator.

[9] The use of robust loss functions is essential to prevent unstable fits, like the least-squares ones, if one or more higher-order derivatives are null and the truncation branch has multiple slopes, the steepest erratic ones appearing to the right of the optimal step size. The proposed algorithm recovers near-optimal step sizes for functions like x^3+1/x at $x=1$.

Slow constrained optimisation. Although constrained optimisation in step 6 prevents unrealistic parameter values, it may be computationally expensive. The use of the adaptive barrier algorithm results in run times of 15 ms for $f(x) = \sin x$, compared to 2 ms for a simple unconstrained method such as BFGS. The objective function is smooth in (β, γ), which is why quasi-Newton methods exhibit reliable performance at a reduced computational cost.

Visual diagnostics. The importance of visual diagnostics for adequacy checks cannot be overstated because it may reveal pathological cases such as vanishing leading-order derivatives or multiple slope segments. If the derivative or gradient is the object of interest, then, a visualisation should be carried out to determine if the function is accurate, noisy, or ill-behaved. Overlaying the fitted V-shape on the empirical error helps identify edge cases where leading-order derivatives may be zero. One such example is $f(x) = 6x^5 - 56x^3 + x^{-2}$ at $x_0 = 1$. Since $f'''(x) = 360x^2 - 24/x^5 - 330$ and $f^{(\mathrm{V})}(x) = 720(1 - 1/x^7)$ are equal to zero at $x = 1$, the first non-zero truncated term is $|f^{(\mathrm{VII})}(x)|h^6/7!$, hence the slope 6 of the truncation branch, not 2. Such examples are highly unlikely in real applications because they are usually restricted to a single point x_0, whereas in numerical optimisation, the value of x is updated multiple times. Thus, we mention only in passing that the correct slope of the right branch should be determined by finding the first higher-order derivative that is not exactly zero.

4 Benchmark

We evaluate the accuracy of the proposed kink-based step-size selection method using a well-established benchmark consisting of five functions originally introduced by Dumontet and Vignes (1977) and Stepleman and Winarsky (1979) and three mildly pathological ones: the linear x, the quadratic x^2, and the numerically ill-conditioned $\sin(x^2 + 10^6 x)$. For x, we draw a grid of 10 000 random uniform points on the interval $[0.1, 12.5]$. At each point, we compute $\hat{f}'_{\mathrm{CD},2}(x, h)$ using optimal step sizes determined by three algorithms: Dumontet–Vignes (D–V), Stepleman–Winarsky (S–W), and the proposed kink-based one (K). The implementations used in this benchmark are available in the R package `pnd` version 0.1.0 (Kostyrka, 2025), specifically via the functions `step.DV`, `step.SW`, and `step.K`.

Table 1 reports the median relative error across the test grid for each function and for each step-size-selection method. The final column indicates the percentage of grid points where the kink-based method yielded the lowest error among all three techniques.

Function	D–V	S–W	Kink-based	Best
$\sin x$	$6.14 \cdot 10^{-12}$	$2.26 \cdot 10^{-11}$	$4.59 \cdot 10^{-12}$	52%
$\exp x$	$1.67 \cdot 10^{-11}$	$2.09 \cdot 10^{-11}$	$5.67 \cdot 10^{-12}$	67%
$\log x$	$1.31 \cdot 10^{-11}$	$9.32 \cdot 10^{-12}$	$8.78 \cdot 10^{-12}$	46%
$\sqrt{x}$	$1.27 \cdot 10^{-11}$	$1.15 \cdot 10^{-11}$	$7.89 \cdot 10^{-12}$	46%
$\tan^{-1}(x)$	$5.24 \cdot 10^{-11}$	$3.77 \cdot 10^{-11}$	$3.35 \cdot 10^{-11}$	42%
x	$8.24 \cdot 10^{-12}$	$8.97 \cdot 10^{-12}$	$1.63 \cdot 10^{-12}$	70%
x^2	$1.67 \cdot 10^{-12}$	$8.14 \cdot 10^{-12}$	$1.24 \cdot 10^{-12}$	56%
$\sin(x^2 + 10^6 x)$	$3.10 \cdot 10^{-6}$	1.00	$3.57 \cdot 10^{-7}$	87%

Table 1: Median relative errors of step-size-selection methods

The kink-based method consistently outperforms the Dumontet–Vignes and Stepleman–Winarsky methods across all eight test cases. While the kink-based approach does not necessarily yield the minimal error at every individual grid point, it consistently produces the lowest median relative error overall. Remarkably, it demonstrates advantages for numerically challenging scenarios

such as the highly oscillatory function $\sin(x^2 + 10^6 x)$ and the linear function x, suggesting robust performance across both linear and highly nonlinear cases.

It is worth noting that the benchmark implementations of the Dumontet–Vignes and Stepleman–Winarsky methods already include numerical safeguards not mentioned in the original articles, such as safe handling of near-zero third derivatives or provision of fail-safe defaults that prevent impractically large step sizes when the truncation error is near-zero. Even with these enhancements, the kink-based method remains dominant in accuracy.

The improved accuracy does come at a higher computational cost: the kink-based method used a fixed number of 61 function evaluations. In comparison, the other two methods typically require fewer evaluations, though their exact computational cost depends on the local behaviour of the combined error. Nevertheless, the new algorithm is infinitely parallelisable, capable of fully exploiting modern multi-core architectures and high-performance computing clusters. In contrast, the maximum parallel threads counts in two other algorithms are 4 and 2, respectively, and their iteration counts are function-dependent and unpredictable. Hence, despite the higher absolute evaluation count, the kink-based method provides a predictable computational budget and is guaranteed to terminate in predictable time.

5 Conclusion

We have introduced a statistically motivated, kink-based procedure for selecting the step size in finite-difference approximations of numerical derivatives. By interpreting the combined truncation–rounding error curve in double-logarithmic coordinates as a piecewise linear V-shape, the method allows identification of the regression-kink location – the point at which the dominant source of error switches from rounding to truncation. Explicit diagnostic checks ensure that the kink is detected whenever it exists and that sensible fall-backs are invoked in pathological cases in which the truncation branch is absent or flat, such as low-degree polynomials or functions locally symmetric about the evaluation point.

Accuracy. Across the 8-function test suite, the proposed algorithm exhibits the lowest median absolute error compared with the popular rules of Dumontet and Vignes (1977) and Stepleman and Winarsky (1979) and a percentage of best-out-of-three step sizes in 42–87% experiments.

Robustness. The regression formulation makes the method insensitive to local oscillations or slope reversals that may cause premature termination in monotonicity-based schemes. The piecewise-linear fit aggregates global information over many candidate step sizes without incurring substantial computational costs.

Generality. The proposed method applies to arbitrary derivative orders, accuracy orders, and non-uniform stencils, requires no *a priori* knowledge of higher-order derivatives, and remains effective for moderately noisy functions. It allows the user to increase the reliability of the statistical fit by choosing a smaller ratio between two subsequent candidate step sizes.

Cost. Implementation with quasi-Newton optimisation of the two-parameter loss takes < 3 ms per derivative on a modern CPU once the values of f have been computed on a grid. Evaluation of f on a grid can be parallelised, speeding the algorithm up several times.

Practical guidance. Diagnostic plots of the empirical error curve and the fitted V-shape are strongly encouraged: they expose both potential plateaus and missing branches due to higher-order cancellations.

Limitations and future work. The assumption of the method is the $(d+a)$-times differentiability of f in the neighbourhood of the evaluation point; extending the approach to piecewise-smooth or stochastic objectives would require a more sophisticated set of rules. Also, while the paper focuses on scalar derivatives, a coordinate-wise application combined with simultaneous kink detection may yield competitive Jacobian estimates for functions not subject to explosive growth; the notion of step-size optimality in the multivariate setting remain open.

In summary, kink-based step-size selection provides a principled, accurate, and computationally feasible alternative to classical heuristics. Its ability to exploit the full shape of the error curve rather than single-point slope conditions makes it especially attractive for modern statistical and optimisation workloads in which accuracy outweighs a small increase in evaluation time.

References

Aitken, A. C. (1932). On interpolation by iteration of proportional parts, without the use of differences. *Proceedings of the Edinburgh Mathematical Society*, *3*(1), 56–76. https://doi.org/10.1017/S0013091500013808

Aitken, A. C. (1938). The application of quadratic extrapolation to the evaluation of derivatives, and to inverse interpolation. *Proceedings of the Royal Society of Edinburgh*, *58*, 161–175. https://doi.org/10.1017/S0370164600011093

Barton, R. R. (1992). Computing forward difference derivatives in engineering optimization. *Engineering Optimization*, *20*(3), 205–224. https://doi.org/10.1080/03052159208941281

Björck, Å., & Pereyra, V. (1970). Solution of Vandermonde systems of equations. *Mathematics of computation*, *24*(112), 893–903.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, *31*(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

Curtis, A. R., & Reid, J. K. (1974). The choice of step lengths when using differences to approximate Jacobian matrices. *IMA Journal of Applied Mathematics*, *13*(1), 121–126. https://doi.org/10.1093/imamat/13.1.121

Dumontet, J., & Vignes, J. (1977). Détermination du pas optimal dans le calcul des dérivées sur ordinateur. *RAIRO. Analyse numérique*, *11*(1), 13–25. https://doi.org/10.1051/m2an/1977110100131

Dumontet, J. (1973). *Algorithme de dérivation numérique: Étude théorique et mise en œuvre sur ordinateur* [Doctoral dissertation, Université de Paris VI].

Everett, J. D. (1901). On a new interpolation formula. *Journal of the Institute of Actuaries*, *35*, 452–458. https://doi.org/10.1017/S0020268100007149

Fornberg, B. (1988). Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, *51*(184), 699–706. https://doi.org/10.1090/s0025-5718-1988-0935077-0

Gill, P. E., Murray, W., Saunders, M. A., & Wright, M. H. (1983). Computing forward-difference intervals for numerical optimization. *SIAM Journal on Scientific and Statistical Computing*, *4*(2), 310–321. https://doi.org/10.1137/0904025

Jordan, C. (1928). Sur une formule d'interpolation. In N. Zanichelli (Ed.), *Atti del congresso internazionale dei matematici* (pp. 157–177).

Keller, H. B., & Pereyra, V. (1978). Symbolic generation of finite difference formulas. *Mathematics of Computation*, *32*, 955–971. https://doi.org/10.1090/S0025-5718-1978-0494848-1

Kostyrka, A. V. (2025). *pnd: Parallel numerical derivatives, gradients, Jacobians, and Hessians of arbitrary accuracy order* [R package version 0.1.0]. University of Luxembourg. https://doi.org/10.32614/CRAN.package.pnd

Lakin, W. D. (1986). Differentiating matrices for arbitrarily spaced grid points. *International Journal for Numerical Methods in Engineering*, *23*(2), 209–218. https://doi.org/10.1002/nme.1620230205

Lyness, J. N., & Moler, C. B. (1967). Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, *4*(2), 202–210. https://doi.org/10.1137/0704019

Mari, A., Bromley, T. R., & Killoran, N. (2021). Estimating the gradient and higher-order derivatives on quantum hardware. *Phys. Rev. A*, *103*, 012405. https://doi.org/10.1103/PhysRevA.103.012405

Mathur, R. (2012). *An analytical approach to computing step sizes for finite-difference derivatives* [Doctoral dissertation, University of Texas at Austin]. https://hdl.handle.net/2152/ETD-UT-2012-05-5275

Mathur, R. (2013). *Algorithm AAS 13-723: An analytical approach to computing step sizes for finite-difference derivatives*. Emergent Space Technologies, Inc. https://www.emergentspace.com/wp-content/uploads/stepsize.pdf

Oliver, J. (1980). The selection of interpolation points in the numerical approximation of second and third order derivatives. *Journal of Computational and Applied Mathematics*, *6*(3), 229–239. https://doi.org/10.1016/0771-050x(80)90030-3

Oliver, J., & Ruffhead, A. (1975). The selection of interpolation points in numerical differentiation. *BIT Numerical Mathematics*, *15*(3), 283–295. https://doi.org/10.1007/BF01933661

Prentice, J. S. C. (2011). Truncation and roundoff errors in three-point approximations of first and second derivatives. *Applied Mathematics and Computation*, *217*(9), 4576–4581. https://doi.org/https://doi.org/10.1016/j.amc.2010.11.008

Ridders, C. J. F. (1982). Accurate computation of f'(x) and f'(x) f"(x). *Advances in Engineering Software*, *4*(2), 75–76. https://doi.org/10.1016/S0141-1195(82)80057-0

Stepleman, R. S., & Winarsky, N. D. (1979). Adaptive numerical differentiation. *Mathematics of Computation*, *33*(148), 1257–1264. https://doi.org/10.1090/s0025-5718-1979-0537969-8