Differentiation

1 Preamble

In engineering analysis we must often evaluate the derivatives of functions in order to predict or optimize the performance of a system. In some cases the derivative directly serves as the quantity of interest; for example, the derivative of the density is related to the frequency of internal gravity waves in the ocean. In other cases, the derivative, or gradient, serves to guide a design process towards better performance. Unfortunately, in most cases we are not provided with a closed-form representation of the function, and instead we must base our prediction on limited (experimental) observations or (computational) evaluations. In this nutshell we answer the following question: if we can probe a function at some few input values near an input value of interest, how can we estimate the (say, first and second) derivatives of the function at the input value of interest? In this nutshell we consider one approach to this approximation problem: the venerable "finite difference" method.

We introduce the notion of finite difference approximation, and we present several important numerical differentiation schemes: approximation of the first derivative of a function by forward, backward, and centered difference formulas; approximation of the second derivative of a function by a centered difference formula. We derive error bounds for these finite difference approximations based on Taylor series expansions, and we discuss the issues of resolution and smoothness. Finally, we discuss the general concept of finite-precision arithmetic, and demonstrate the effect of truncation and round-off errors on convergence (and indeed, divergence).

Prerequisites: univariate differential and integral calculus; basic principles of numerical methods: discretization, convergence, convergence order, resolution, operation counts and FLOPs, asymptotic and big-O estimates; Interpolation: formulation, error analysis, and computational considerations.

2 Motivation: Example

Let us consider a concrete example of numerical differentiation. Oscillations in a statically stable ocean environment are characterized by the *Brunt-Väisälä frequency*,

$$N = \sqrt{\frac{g}{\rho}\frac{\partial \rho}{\partial x}},$$

where ρ is the density of the ocean water, g is the gravitational acceleration, and x is the depth measured from the free surface. (Depth shall always refer to vertical distance below the free surface.) We wish to approximate the frequency at a particular depth of interest, say $d = 200\text{m}$.

To accomplish our task, we are given Table 1, which provides the density of ocean water at seven different depths. (The density of ocean water depends on the salinity and temperature, which in turn depends on depth.) How can we exploit the data of Table 1 to approximate the derivative $\partial\rho/\partial x$ and subsequently the Brunt-Väisälä frequency at the depth of interest, $d = 200\text{m}$? Can we say anything about the accuracy of this estimate? What is the cost associated with the computation of this frequency estimate? The material provided in this nutshell will help you answer these questions.

depth (m)	0	100	200	300	400	500	600
density (kg/m^3)	1024.985	1025.375	1025.815	1026.271	1026.707	1027.086	1027.375

Table 1: Variation in the density of ocean water with depth.[1]

3 Forward Difference

We first consider arguably the simplest form of numerical differentiation: the forward difference formula. Here, we wish to approximate the derivative $f'(\bar{x}_0)$ based on function values $f(\bar{x}_0)$ and $f(\bar{x}_1)$ for $\bar{x}_0 < \bar{x}_1$. To approximate the derivative, we first recall the definition of the derivative:

$$f'(\bar{x}_0) \equiv \lim_{\delta\to 0} \frac{f(\bar{x}_0+\delta) - f(\bar{x}_0)}{\delta}.$$

We now approximate this derivative by replacing the limit $\delta \to 0$ — a differential — with a *finite difference*, $h \equiv \bar{x}_1 - \bar{x}_0 > 0$; the resulting approximation is the *forward difference formula*,

$$f'_h(\bar{x}_0) = \frac{f(\bar{x}_0+h) - f(\bar{x}_0)}{h} = \frac{f(\bar{x}_1) - f(\bar{x}_0)}{h}. \quad (1)$$

A geometric interpretation of the forward difference formula is provided in Figure 1. In general, a *finite difference formula* approximates the derivative of a function at a point of interest $\bar{x}_0$ based on the function value at $\bar{x}_0$ and selected additional points in the neighborhood of $\bar{x}_0$. The set of points which serve to evaluate the derivative is denoted the *finite difference stencil* or simply the *stencil*; the stencil for the forward difference formula is $\{\bar{x}_0, \bar{x}_1\}$.

As a simple example, we apply the forward difference formula to the ocean density example. In particular, we wish to estimate the Brunt-Väisälä frequency at the depth of interest $d = 200\text{m}$.

[1]This table is derived from the representative ocean density profile provided in *Windows to the Universe*.

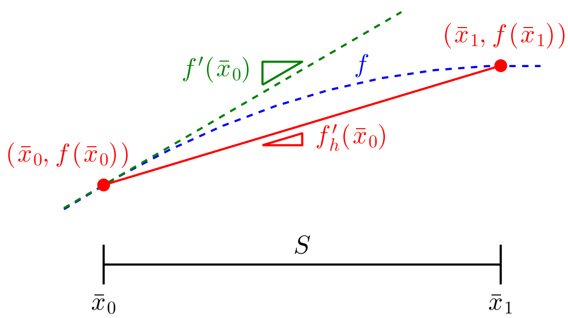

Figure 1: Illustration of the forward difference formula.

To approximate the derivative $\frac{\partial \rho}{\partial x}$ at depth of interest $d \equiv 200\text{m}$ we first choose the stencil $\{\bar{x}_0 \equiv 200\text{m}, \bar{x}_1 \equiv 300\text{m}\}$; the stencil must be a subset of the points available in Table 1. We then enlist the forward different formula (1) to obtain

$$f'_h(d = 200) = \frac{f(\bar{x}_1 \equiv 300) - f(\bar{x}_0 \equiv 200)}{300 - 200} = \frac{1026.271 - 1025.815}{100} = 0.456\,\text{kg/m}^2\ .$$

The approximation to the Brunt-Väisälä frequency at $d = 200\text{m}$ is then given by

$$N = \sqrt{\frac{g}{\rho}\frac{\partial \rho}{\partial x}} = \sqrt{\frac{g}{\rho(d \equiv 200)} f'_h(d \equiv 200)} = \sqrt{\frac{9.81}{1025.815} \cdot 0.456} = 6.60 \times 10^{-2}\,\text{s}^{-1}\ . \tag{2}$$

Note we evaluate the density in the denominator at the depth of interest, here $d = 200\text{m}$.

CYAWTP 1. Invoke the forward difference formula and the (depth, density) data of Table 1 to estimate the Brunt-Väisälä frequency at depths of interest $d = 300\text{m}$ and $d = 500\text{m}$, respectively.

We now proceed with the error analysis: we wish to understand how well the forward difference estimate $f'_h(\bar{x}_0)$ approximates the exact derivative $f'(\bar{x}_0)$. Towards this end, we first recall the Taylor series expansion with explicit remainder,

$$f(\bar{x}_1) = f(\bar{x}_0) + f'(\bar{x}_0)h + \frac{1}{2}f''(\xi)h^2\ ;$$

recall that the Mean-Value Theorem ensures the existence of a $\xi \in [\bar{x}_0, \bar{x}_1]$ for which equality holds (presuming f is twice differentiable). We now compare the forward difference approximation and the exact derivative at $\bar{x}_0$,

$$\begin{aligned} |f'(\bar{x}_0) - f'_h(\bar{x}_0)| &= \left| f'(\bar{x}_0) - \frac{f(\bar{x}_1) - f(\bar{x}_0)}{h} \right| = \left| f'(\bar{x}_0) - \frac{1}{h}\left(f'(\bar{x}_0)h + \frac{1}{2}f''(\xi)h^2 \right) \right| \\ &= \left| \frac{1}{2} f''(\xi) h \right| \le \frac{1}{2} \max_{x \in [\bar{x}_0, \bar{x}_1]} |f''(x)| h. \end{aligned} \tag{3}$$

The accuracy of the forward difference formula depends on two factors: the discretization resolution parameter h — the shorter the segment, the better the estimate; the second derivative of the function f — the slower the rate of the change of the derivative with x, the better the estimate. It is not surprising that $f'_h(\bar{x}_0)$ converges to $f'(\bar{x}_0)$ since the by construction $f'(\bar{x}_0)$ is the limit of $f'_h(\bar{x}_0)$ as h tends to zero. However, we also observe that the forward difference formula is *first-order accurate*: the order of the scheme is $p = 1$.

Numerical Experiment 2. Invoke the differentiation GUI for forward difference approximation of the derivative $f'(x = 0)$ for the function $f(x) = \exp(x)$: visualize the geometric interpretation of the finite difference approximation; confirm empirically the convergence rate of the forward difference formula.

CYAWTP 3. Sketch the logarithmic convergence curve, $\log(|f'(\bar{x}_0) - f'_h(\bar{x}_0)|)$ as a function of $\log(1/h)$, and the associated logarithmic convergence asymptote, for forward difference approximation of the function $f = \sin(10\pi x)$ at $\bar{x}_0 = 0$. Estimate very roughly the value of the discretization parameter h at which the asymptotic regime is first apparent.

Numerical Experiment 4. Invoke the differentiation GUI to confirm your sketch and estimates of **CYAWTP 3**.

We now briefly comment on the cost associated with the forward difference formula. Given the pairs $(\bar{x}_0, f(\bar{x}_0))$ and $(\bar{x}_1, f(\bar{x}_1))$, the evaluation of the forward difference formula requires two subtractions ($\bar{x}_1 - \bar{x}_0$, and $f(\bar{x}_1) - f(\bar{x}_0)$) and one division ($(f(\bar{x}_1) - f(\bar{x}_0))/(\bar{x}_1 - \bar{x}_0)$); hence the evaluation of the derivative at $\bar{x}_0$ requires, in total, 3 FLOPs.

4 Generalization: Other Finite Difference Formulas

We now introduce a more general procedure to derive finite difference formulas. The key ingredient, just as in our develoment of quadrature rules, is interpolation. We first choose a stencil of input values; we next construct an interpolant of the function; we then evaluate the derivative of the interpolant at the point of interest. For the interpolant we adopt here the "point-based formulation" of interpolation: "what" shall be a low-order polynomial; "where" shall be the points of our stencil. (As always, the "what" and "where" must yield a unique interpolant.) In the context of differentiation, rather than denote our interpolation points $\bar{x}_1$, $\bar{x}_2, \ldots$, we instead we shift the indices such that $\bar{x}_0$ shall always be the input point of interest at which we wish to evaluate the derivative.

Forward difference formula. We first re-derive the forward difference formula but now by our "differentiation through interpolation" approach. We choose for our stencil $\{\bar{x}_0, \bar{x}_1\}$ and hence segment $S \equiv [\bar{x}_0, \bar{x}_1]$. We choose for our interpolant "what": linear, and (from the stencil) "where": $\bar{x}_0$, and $\bar{x}_1$. The linear interpolant is thus given by

$$(\mathcal{I}f)(x) = f(\bar{x}_0) + \frac{f(\bar{x}_1) - f(\bar{x}_0)}{\bar{x}_1 - \bar{x}_0}(x - \bar{x}_0) \quad \text{for } x \text{ in } S.$$

We now approximate the derivative of the function at $\bar{x}_0$, $f'(\bar{x}_0)$, as follows:

$$f'(\bar{x}_0) \approx (\mathcal{I}f)'(\bar{x}_0) = \frac{f(\bar{x}_1) - f(\bar{x}_0)}{\bar{x}_1 - \bar{x}_0} = f'_h(\bar{x}_0)\ ,$$

which is consistent with the formula we obtained earlier.

Backward difference formula. To obtain the backward difference formula, we now choose for our stencil $\{\bar{x}_{-1}, \bar{x}_0\}$, $\bar{x}_{-1} < \bar{x}_0$, and hence segment $S \equiv [\bar{x}_{-1}, \bar{x}_0]$. We again choose a linear interpolant, which for our backward stencil is now given by

$$(\mathcal{I}f)(x) = f(\bar{x}_{-1}) + \frac{f(\bar{x}_0) - f(\bar{x}_{-1})}{\bar{x}_0 - \bar{x}_{-1}}(x - \bar{x}_{-1}) \quad \text{for } x \text{ in } S.$$

Finally, we take the derivative of this linear interpolant at $x = \bar{x}_0$ to obtain

$$f'_h(\bar{x}_0) = \frac{f(\bar{x}_0) - f(\bar{x}_{-1})}{\bar{x}_0 - \bar{x}_{-1}}.$$

as our backward difference formula. The error bound for the backward difference formula is identical to the error bound for the forward difference formula, (3), except that the maximum of $|f'|$ over $[\bar{x}_0, \bar{x}_1]$ is replaced by the maximum of $|f'|$ over $[\bar{x}_{-1}, \bar{x}_0]$. The operation count of the backward difference formula is identical to the operation count for the forward difference formula.

CYAWTP 5. Sketch a geometric interpretation, similar to Figure 1 for the forward difference formula, for the backward difference formula.

CYAWTP 6. Provide a lower bound and an upper bound for the Brunt-Väisälä frequency at $d = 100$m and also at $d = 500$m.

Centered difference formula. Here we choose the stencil $\{\bar{x}_{-1}, \bar{x}_0, \bar{x}_1\}$ for $\bar{x}_{-1} < \bar{x}_0 < \bar{x}_1$ and $\bar{x}_0 - \bar{x}_{-1} = \bar{x}_1 - \bar{x}_0 = h$; our segment is then given by $S \equiv [\bar{x}_{-1}, \bar{x}_1]$. We choose for our interpolant "where": quadratic polynomial, and (from the stencil) "what": $\bar{x}_{-1}, \bar{x}_0, \bar{x}_1$; recall that a quadratic polynomial is uniquely determined by three points. We may then write our interpolant as

$$(\mathcal{I}f)(x) = \frac{1}{2h^2} f(\bar{x}_{-1})(x - \bar{x}_0)(x - \bar{x}_1) - \frac{1}{h^2} f(\bar{x}_0)(x - \bar{x}_{-1})(x - \bar{x}_1) + \frac{1}{2h^2} f(\bar{x}_1)(x - \bar{x}_{-1})(x - \bar{x}_0) \ , \ x \in S \ . \qquad (4)$$

(Note that this interpolant as written is valid only for equispaced points.) We then differentiate (4) at $\bar{x}_0$ to obtain

$$f'_h(\bar{x}_0) = \frac{f(\bar{x}_1) - f(\bar{x}_{-1})}{2h} \ . \qquad (5)$$

which is denoted the *centered difference formula.* Note that this specific formula is derived for, and hence only applicable to, equispaced points.

CYAWTP 7. Sketch a geometric interpretation, similar to Figure 1 for the forward difference formula, of the centered difference formula.

CYAWTP 8. Derive the centered difference formula (5) from the expression (4) for the quadratic interpolant.

We may apply the Taylor series expansion to obtain an error bound for the centered difference approximation:

$$|f'(\bar{x}_0) - f'_h(\bar{x}_0)| \le \frac{1}{6} h^2 \max_{x \in [\bar{x}_{-1}, \bar{x}_1]} |f'''(x)|.$$

We observe that that the centered difference formula is *second*-order accurate; as the operation count is only 3 FLOPs — the same as for the first-order accurate forward and backward difference approximations — the centered difference approximation is a good buy. However, we note that, as always, higher order schemes require more smoothness: the error in the centered difference approximation depends on the *third* derivative of f.

Numerical Experiment 9. Invoke the differentiation GUI for centered difference approximation of the derivative $f(x) = x|x|$ at $x = 0$. What order convergence do you observe, and why?

CYAWTP 10. Which, if any, of the finite difference formulas above — forward difference, backward difference, centered difference — will differentiate exactly a linear function? How about a quadratic function? How about a cubic function?

5 Finite Difference Formula for the Second Derivative

We may also develop finite difference approximations for the *second* derivative of a function f. Indeed, we may follow the same procedure as for first derivatives with one crucial difference: we select a stencil; we next construct an interpolant of f; we then differentiate — this time *twice* — the interpolant at the point of interest. Clearly since we must differentiate twice we must choose an interpolant which is at least quadratic — otherwise our approximate derivative is identically zero — which in turn demands at least a three-point stencil for uniqueness.

In fact, we have just described the most common finite difference approximation of the second derivative of a function f. We choose the stencil $\{\bar{x}_{-1}, \bar{x}_0, \bar{x}_1\}$ for $\bar{x}_{-1} < \bar{x}_0 < \bar{x}_1$ and $\bar{x}_0 - \bar{x}_{-1} = \bar{x}_1 - \bar{x}_0 = h$; our segment is then given by $S \equiv [\bar{x}_{-1}, \bar{x}_1]$. We choose for our interpolant "where": quadratic polynomial, and (from the stencil) "what": $\bar{x}_{-1}, \bar{x}_0, \bar{x}_1$. We thus arrive at (4), just as in our derivation of the centered difference approximation of the first derivative of f. But now, for the second derivative approximation, we differentiate the interpolant twice to obtain

$$f''_h(\bar{x}_0) \equiv (\mathcal{I}f)''(\bar{x}_0) \equiv \frac{f(\bar{x}_{-1}) - 2f(\bar{x}_0) + f(\bar{x}_1)}{h^2} \approx f''(\bar{x}_0) \; . \tag{6}$$

Note that whereas for first derivative approximations h^1 appears in the denominator, for second derivative approximations h^2 appears in the denominator, consistent with the differentials of the Liebniz notation for derivatives.

The error of the centered difference formula for the second derivative may be bounded as

$$|f''(\bar{x}_0) - f''_h(\bar{x}_0)| \leq \frac{1}{12} h^2 \max_{x \in [\bar{x}_{-1}, \bar{x}_1]} |f^{(4)}(x)| \; , \tag{7}$$

where $f^{(4)}$ denotes the fourth derivative of the function; the approximation is *second-order* accurate. Note that this specific formula is derived for, and hence only applicable to, equispaced points, and in this case it is not possible to develop a strictly second-order accurate scheme if $\bar{x}_0 - \bar{x}_{-1} \neq \bar{x}_1 - \bar{x}_0$.

CYAWTP 11. Derive the second-derivative approximation (7) from the expression (4) for the quadratic interpolant.

There are many applications of this second-derivative formula, for example in the numerical treatment of second-order ordinary or partial differential equations. We consider here another application: *a posteriori* error estimation. Most of the estimates we present in these nutshells are of the *a priori* variety, provided before we obtain our numerical result; *a priori* estimates are not quantitative, in the sense that the bound references the (unknown) exact solution. In constrast, *a posteriori* error estimates are provided after we obtain our numerical result; *posteriori* error estimates are thus quantitative, and can serve both to assess the accuracy and adapt the discretization (for example, to choose smaller segments in certain parts of the interval).

We give here a simple example. We are given a discretization of $[a, b]$ in the form of equispaced segment endpoints $x_1 \equiv a, x_2, \ldots, x_N \equiv b$. We are provided with function values $f(x_i), 1 \leq i \leq N$; we wish to predict $f(x)$ at any point in $[a, b]$. We first construct a (global) piecewise-linear interpolant, $\mathcal{I}f$. Next, to quantitatively understand the accuracy of our interpolant, we evaluate $f''_h(x_i), 2 \leq i \leq N-1$, according to the formula (6), and compute — motivated by the *a priori* error bound for piecewise-linear interpolation presented in the Interpolation nutshell — $\hat{e}_i \equiv (h^2)/8 \cdot |f''_h(x_i)|, 2 \leq i \leq N-1$. Finally, we may take (say) the maximum of $\hat{e}_i$ and $\hat{e}_{i+1}$ as an *a posteriori* error estimator for the interpolation error over segment $S_i, 2 \leq i \leq N-2$. (For our error estimators for S_1 and S_{N-1} we take $\hat{e}_2$ and $\hat{e}_{N-1}$, respectively.)

6 Finite-Precision Arithmetic

In a digital computer, a floating-point number is represented as a mantissa — with 1 non-zero digit before the decimal, and n digits after the decimal — and an exponent. (In reality, the representation is binary, not digital, but the arguments are similar.) Representation of a number such as e thus involves a *truncation error* or *representation error*: of the infinite number of non-repeating digits, we retain only the first n digits. We denote by $\epsilon \equiv 10^{-n}$ the *machine precision*.

CYAWTP 12. Consider the two numbers $x = 1.000000$ and $y = 1.000001$. Calculate $z = x + y$; how many significant figures does z retain? Now calculate $w = y - z$; how many significant figures does w retain?

In some cases, truncation error can corrupt our numerical approximation. As a simple example, we consider the forward difference formula (1). As h approaches ϵ (machine precision), the convergence as h decreases will be replaced by divergence: the $1/h$ in the denominator of the difference formula will amplify the truncation error; for $h < \epsilon$ (roughly) we can no longer distinguish $f(x_0)$ and $f(x_1)$ in the numerator and (1) will evaluate to zero — hence we obtain a *fixed order-unity* error. In the appendix we analyze the convergence of finite difference approximations (of first derivatives) in the presence of finite-precision effects: we can predict the value of the discretization parameter h (as a function of ϵ) at which divergence is first observed; we can further demonstrate that the error which can be achieved in the presence of finite-precision effects is lower for higher-order schemes. We confirm these effects empirically in Figure 2.

7 Perspectives

We have only here provided a first look at the topic of numerical differentiation. A more in-depth study may be found in *Math, Numerics, and Programming (for Mechanical Engineers)*, M Yano, JD Penn, G Konidaris, and AT Patera, available on MIT OpenCourseWare, which adopts similar notation to these nutshells and hence can serve as a companion reference. For an even more comprehensive view from both the computational and theoretical perspectives we recommend *Numerical Mathematics*, A Quarteroni, R Sacco, F Saleri, Springer, 2000.

Of the many further topics of interest, perhaps the most important is the treatment of noisy data. In this nutshell we consider noise-free function evaluations. However, our analysis of finite-precision arithmetic effects suggests that differentiation — unlike integration — can be very sensitive to noise. In a subsequent nutshell we consider an alternative approach to numerical differentiation which is more suitable for noisy data: least-squares projection.

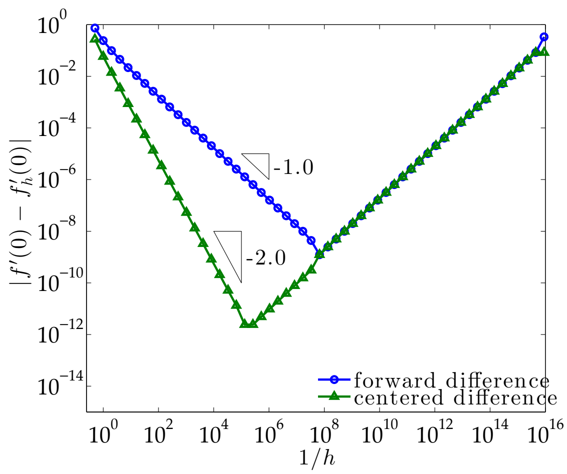

Figure 2: The error in the finite difference approximation of the first derivative of $f(x) = \exp(x)/3$ at $x = 0$ in finite-precision ($\epsilon \approx 1.0 \times 10^{-16}$) arithmetic. We observe the anticipated convergence result (of infinite-precision arithmetic) until h is on the order of $\epsilon^{1/(1+p)}$ where p is the order of the scheme.

Appendix A Finite Difference in Finite-Precision Arithmetic

We have so far assumed exact arithmetics. In practice, a *floating-point number* is represented in the computer as a mantissa — with 1 non-zero digit before the decimal, and n digits after the decimal — and an exponent. (In reality, the representation is binary, not base-10, but the arguments are similar.) Thus, in a computer, a real number x is approximated by its *floating-point approximation* $\texttt{fl}(x)$, which satisfies

$$|x - \texttt{fl}(x)| \le \epsilon |x|,$$

where ϵ is the *machine precision.* In other words, for any real number x, we can find $\texttt{fl}(x)$ such that $x = \texttt{fl}(x) + \tilde{\epsilon}|x|$ for some $|\tilde{\epsilon}| \le \epsilon$. The floating-point number system used in a typical modern computer carries $n \approx 16$ significant digits, and hence $\epsilon \approx 10^{-16}$. We note that the floating-point number system can represent any real number quite accurately, and hence the errors arising from finite-precision arithmetics can often be ignored.

However, it is certainly possible to observe the effect of finite-precision arithmetics, in particular when the difference between two nearly identical numbers plays an important role in a calculation. As a simple example, we consider the forward-difference formula (1). The function values $f(\bar{x}_0)$ and $f(\bar{x}_1)$ in the numerator are represented in the floating-point number system as $\texttt{fl}(f(\bar{x}_0)) = f(\bar{x}_0) + \tilde{\epsilon}_0$ and $\texttt{fl}(f(\bar{x}_1)) = f(\bar{x}_1) + \tilde{\epsilon}_1$ for $\tilde{\epsilon}_0 \le \epsilon$ and $\tilde{\epsilon}_1 \le \epsilon$. (For simplicity we assume the rest of the arithmetic is carried out exactly; of course in a computer all intermediate steps must also adhere to the floating-point number system.) The error in our forward difference approximation is then

$$\begin{aligned}
f'(\bar{x}_0) - f'_{h,\mathrm{fp}}(\bar{x}_0) &= \left| f'(\bar{x}_0) - \frac{\texttt{fl}(f(\bar{x}_1)) - \texttt{fl}(f(\bar{x}_0))}{h} \right| \\
&= \left| f'(\bar{x}_0) - \frac{f(\bar{x}_1) + \tilde{\epsilon}_1 f(\bar{x}_1) - f(\bar{x}_0) - \tilde{\epsilon}_0 f(\bar{x}_0)}{h} \right| \\
&= \left| f'(\bar{x}_0) - \frac{f(\bar{x}_1) - f(\bar{x}_0)}{h} - \frac{\tilde{\epsilon}_1 f(\bar{x}_1) - \tilde{\epsilon}_0 f(\bar{x}_0)}{h} \right| \\
&= \left| f'(\bar{x}_0) - f'_h(\bar{x}_0) - \frac{\tilde{\epsilon}_1 f(\bar{x}_1) - \tilde{\epsilon}_0 f(\bar{x}_0)}{h} \right| \\
&\le \left| f'(\bar{x}_0) - f'_h(\bar{x}_0) \right| + \left| \frac{\tilde{\epsilon}_1 f(\bar{x}_1) - \tilde{\epsilon}_0 f(\bar{x}_0)}{h} \right| \\
&\le \frac{1}{2} h \max_{x \in [\bar{x}_0, \bar{x}_1]} |f''(x)| + 2 \frac{\epsilon}{h} \max_{x \in [\bar{x}_0, \bar{x}_1]} |f(x)|; \qquad (8)
\end{aligned}$$

in the last step, the bound for the first term follows from our forward difference error bound (3) and the bound for the second term follows from $\tilde{\epsilon}_1 \le \epsilon$ and $\tilde{\epsilon}_2 \le \epsilon$. The finite-precision forward difference error consists of two contributions: the first is the standard finite-difference error; the second is the error due to the finite-precision arithmetic. In particular, note that the second error *diverges* as $h \to \epsilon$.

We can readily show that the value of h that minimizes the bound (8) is

$$h^* = \epsilon^{1/2} \sqrt{\frac{4 \max_{x \in [\bar{x}_0, \bar{x}_1]} |f(x)|}{\max_{x \in [\bar{x}_0, \bar{x}_1]} |f''(x)|}}$$

and the bound associated with the optimal h is

$$\left|f'(\bar{x}_0) - f'_{h^*,\text{fp}}(\bar{x}_0)\right| \le \epsilon^{1/2}\sqrt{4\left(\max_{x\in[\bar{x}_0,\bar{x}_1]}|f(x)|\right)\left(\max_{x\in[\bar{x}_0,\bar{x}_1]}|f''(x)|\right)}.$$

For the forward difference formula evaluated in finite-precision arithmetics, h that leads to the minimum error bound is $\mathcal{O}(\epsilon^{1/2})$ and the minimum error bound is $\mathcal{O}(\epsilon^{1/2})$.

We may analyze the finite-precision centered difference error in a similar manner:

$$\begin{aligned} |f'(\bar{x}_0) - f'_{h,\text{fp}}(\bar{x}_0)| &= \left| f'(\bar{x}_0) - \frac{\texttt{fl}(f(\bar{x}_1)) - \texttt{fl}(f(\bar{x}_{-1}))}{2h} \right| \\ &\le |f'(\bar{x}_0) - f'_h(\bar{x}_0)| + \left| \frac{\tilde{\epsilon}_1 f(\bar{x}_1) - \tilde{\epsilon}_{-1} f(\bar{x}_{-1})}{2h} \right| \\ &\le \frac{1}{6}h^2 \max_{x\in[\bar{x}_{-1},\bar{x}_1]} |f''(x)| + \frac{\epsilon}{h} \max_{x\in[\bar{x}_{-1},\bar{x}_1]} |f(x)|; \end{aligned}$$

the finite difference error converges at the rate of h^2, and the finite precision error diverges at the rate of h^{-1}. The value of h that minimizes the bound is

$$h^* = \epsilon^{1/3}\left(\frac{3\max_{x\in[\bar{x}_0,\bar{x}_1]}|f(x)|}{\max_{x\in[\bar{x}_0,\bar{x}_1]}|f'''(x)|}\right)^{1/3}$$

and the bound associated with the optimal h is

$$\begin{aligned} &\left|f'(\bar{x}_0) - f'_{h^*,\text{fp}}(\bar{x}_0)\right| \\ &\le \epsilon^{2/3}\left(\frac{1}{24}\left(\max_{x\in[\bar{x}_0,\bar{x}_1]}|f(x)|\right)\left(\max_{x\in[\bar{x}_0,\bar{x}_1]}|f'''(x)|\right)^2 + \frac{1}{3}\left(\max_{x\in[\bar{x}_0,\bar{x}_1]}|f(x)|\right)^2\left(\max_{x\in[\bar{x}_0,\bar{x}_1]}|f'''(x)|\right)\right)^{1/3}. \end{aligned}$$

For the centered difference formula evaluated in finite-precision arithmetic, h that leads to the minimum error bound is $\mathcal{O}(\epsilon^{1/3})$ and the minimum error bound is $\mathcal{O}(\epsilon^{2/3})$. Note that, for respective optimal h, the centered difference formula attains smaller error (bound) than the forward difference formula: $\mathcal{O}(\epsilon^{2/3})$ vs. $\mathcal{O}(\epsilon^{1/2})$.

We can in fact generalize the analysis to show the following for an order p scheme: (*i*) the optimal h, h^*, is $\mathcal{O}(\epsilon^{\frac{1}{p+1}})$; (*ii*) the error bound for the optimal h is $\mathcal{O}(\epsilon^{\frac{p}{p+1}})$; (*iii*) the error converges as h^p for $h > h^*$; (*iv*) the error diverges as h^{-1} for $h < h^*$. The behavior for the forward difference formula ($p = 1$) and the centered difference formula ($p = 2$) is captured in Figure 2.