

Numerical Differentiation

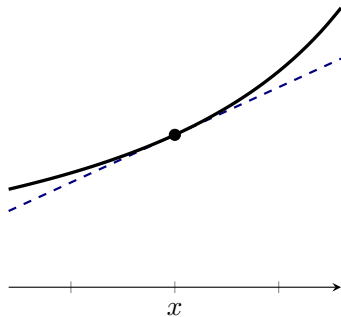
Contents

① Introduction

② Convergence of Difference Formulas

Introduction

Difference Quotients to Approximate Slopes



Let f be a smooth function. Analytically, the derivative is calculated by

$$D\{f\}(x) = f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h},$$

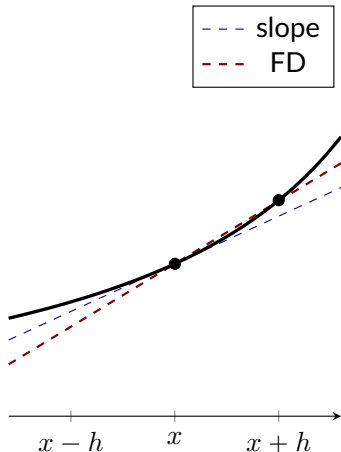
which represents the slope of the line tangent to the graph of f at x .

Difference Quotients to Approximate Slopes (cont')

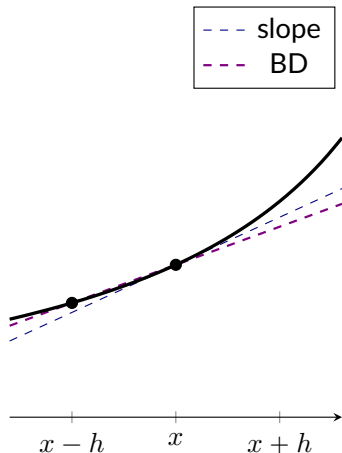
Since the definition relies on h approaching 0, choosing a small, fixed value for h approximates $f'(x)$.

- 1st-order forward difference

$$D_h^{[1f]}\{f\}(x) = \frac{f(x+h) - f(x)}{h}$$



Difference Quotients to Approximate Slopes (cont')



Since the definition relies on h approaching 0, choosing a small, fixed value for h approximates $f'(x)$.

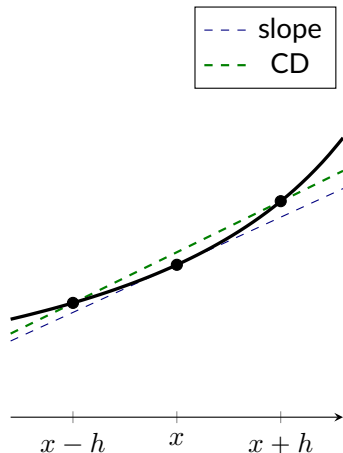
- 1st-order forward difference

$$D_h^{[1f]}\{f\}(x) = \frac{f(x+h) - f(x)}{h}$$

- 1st-order backward difference

$$D_h^{[1b]}\{f\}(x) = \frac{f(x) - f(x-h)}{h}$$

Difference Quotients to Approximate Slopes (cont')



Since the definition relies on h approaching 0, choosing a small, fixed value for h approximates $f'(x)$.

- 1st-order forward difference

$$D_h^{[1f]}\{f\}(x) = \frac{f(x+h) - f(x)}{h}$$

- 1st-order backward difference

$$D_h^{[1b]}\{f\}(x) = \frac{f(x) - f(x-h)}{h}$$

- 2nd-order centered difference

$$D_h^{[2c]}\{f\}(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Difference Quotients to Approximate Slopes (cont')

The three approximation formulas presented above are examples of so-called **finite difference formulas**.

Note

The terms *first-order* and *second-order* refer to how quickly the approximation converges to the actual value of $f'(x)$ as h approaches 0, not to the order of differentiation. More on this later.

Interpolation and Difference Formulas

For simplicity of notation, let's set $x = 0$.

Observe that

- The forward difference formula is simply the slope of the (secant) line through the two points $(0, f(0))$ and $(h, f(h))$.
- Similarly, the backward difference formula is simply the slope of the (secant) line through the two points $(0, f(0))$ and $(h, f(h))$.

Think:

- slope \Leftrightarrow derivative
- line through two points \Leftrightarrow interpolant

A natural extension of this perspective is to think of the centered difference formula as the derivative of the quadratic interpolant of the three points $(-h, f(-h))$, $(0, f(0))$, and $(h, f(h))$.

Interpolation and Difference Formulas (cont')

Exercise 1. Show that the quadratic function

$$q(x) = \frac{x(x-h)}{2h^2}f(-h) - \frac{x^2-h^2}{h^2}f(0) + \frac{x(x+h)}{2h^2}f(h).$$

interpolates $(-h, f(-h))$, $(0, f(0))$, and $(h, f(h))$.

Exercise 2. Show that $q'(0) = D_h^{[2c]}\{f\}(0)$.

Interpolation and Difference Formulas (cont')

In principle, once nodes are determined, a finite difference formula can be derived by:

Interpolate the given function values, then differentiate the interpolant exactly.

Some commonly used difference formulas are provided, without derivation, in the next slide.

Common Difference Formulas

Type	Order	Notation	Formula
Forward	1	$D_h^{[1f]}\{f\}(x)$	$\frac{f(x+h) - f(x)}{h}$
	2	$D_h^{[2f]}\{f\}(x)$	$\frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h}$
Backward	1	$D_h^{[1b]}\{f\}(x)$	$\frac{f(x) - f(x-h)}{h}$
	2	$D_h^{[2b]}\{f\}(x)$	$\frac{3f(x) - 4f(x-h) + f(x-2h)}{2h}$
Centered	2	$D_h^{[2c]}\{f\}(x)$	$\frac{f(x+h) - f(x-h)}{2h}$
	4	$D_h^{[4c]}\{f\}(x)$	$\frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}$

Higher Derivatives

Convergence of Difference Formulas

Convergence of Difference Formulas

- All finite difference formulas introduced converge as $h \rightarrow 0$.
- But there are difficulties in implementing this limiting calculation numerically, so we need to work with $h > 0$ which would yield acceptable accuracy.
- To address this kind of issues, we need to understand how the accuracy of difference formulas increases as $h \rightarrow 0$.
- In other words, we need to study how *quickly* the error $D^{[\cdot]} \{f\}(x) - f'(x)$ diminishes as $h \rightarrow 0$.
- The main tool for the analysis is the Taylor series.

First-Order Difference Formulas

The formula $D_h^{[1f]}\{f\}$ is said to be a **first-order** method because

$$D_h^{[1f]}\{f\}(x) - f'(x) = \underbrace{\frac{1}{2}f''(x)h}_{\text{leading error}} + O(h^2).$$

Derivation. Use the Taylor series of $D_h^{[1f]}\{f\}$ at x :

$$\begin{aligned} D_h^{[1f]}\{f\} - f'(x) &= \frac{f(x+h) - f(x)}{h} - f'(x) \\ &= \frac{1}{h} \left[\cancel{f(x)} + f'(x)h + \frac{f''(x)}{2}h^2 + O(h^3) - \cancel{f(x)} \right] - f'(x) \\ &= \left[\cancel{f'(x)} + \frac{f''(x)}{2}h + O(h^2) \right] - \cancel{f'(x)} \\ &= \frac{f''(x)}{2}h^2 + O(h^2). \end{aligned}$$

Second-Order Difference Formulas

The formula $D_h^{[2c]}\{f\}$ is said to be a **second-order** method because

$$D_h^{[2c]}\{f\}(x) - f'(x) = \underbrace{\frac{1}{6}f'''(x)h^2}_{\text{leading error}} + O(h^4).$$

Derivation. Exercise.

Remarks

- The difference $D_h^{[\cdot]} \{f\}(x) - f'(x)$ is called the **truncation error**. This term is derived from the idea that the finite difference formula has to truncate the series representation and thus cannot be exactly correct for all functions.
- When using a first-order formula, cutting h in half should reduce the error in the approximation of $f'(x)$ by about half, when h is sufficiently small.
- In general, when using an n th-order formula, reducing h by a factor of two should cut the error in $f'(x)$ estimate by a factor of about 2^n , when h is sufficiently small.
- The notation $O(f(n))$ is commonly used to describe the temporal or spatial complexity of an algorithm. In that context, a $O(n^2)$ algorithm is much worse than a $O(n)$ algorithm. (Remember flop counts for LU factorization.) However, when referring to error, a $O(h^2)$ algorithm is **better** than a $O(h)$ algorithm because it means that the accuracy improves faster as h decreases.

Comparison of convergence.

Determining Optimal h

The difference formulas are inherently ill-conditioned as $h \rightarrow 0$ due to catastrophic cancellation when implemented in floating point arithmetic. As an example, consider the numerical evaluation¹ of the centered difference formula:

$$\begin{aligned}\hat{D}_h^{[2c]}\{f\}(x) &= \frac{\hat{f}(x+h) - \hat{f}(x-h)}{2h} \\ &= \frac{f(x+h)(1+\epsilon_+) - f(x-h)(1+\epsilon_-)}{2h}, \quad \text{for some } |\epsilon_{\pm}| \leq \frac{1}{2} \boxed{\text{eps}} \\ &= \frac{f(x+h) - f(x-h)}{2h} + \frac{f(x+h)\epsilon_+ - f(x-h)\epsilon_-}{2h} \\ &= \left[f'(x) + \frac{1}{6} f'''(x) h^2 + O(h^4) \right] + \frac{(f(x) + O(h)) \epsilon_+ - (f(x) + O(h)) \epsilon_-}{2h} \\ &= f'(x) + \frac{1}{6} f'''(x) h^2 + \frac{\epsilon_+ - \epsilon_-}{2h} f(x) + O(\boxed{\text{eps}}) + O(h^4).\end{aligned}$$

So the error is bounded from above by:

$$\left| \hat{D}_h^{[2c]}\{f\}(x) - f'(x) \right| \leq \frac{1}{6} |f'''(x)| h^2 + \frac{\boxed{\text{eps}}}{2h} |f(x)| + O(\boxed{\text{eps}}).$$

¹For simplicity, we only consider round-off errors arising in the evaluation of f .

Determining Optimal h (cont')

Ignoring the last two terms, the error bound consists of two parts:

$$|\text{error}| \lesssim \underbrace{\frac{1}{6} |f'''(x)| h^2}_{\text{truncation error}} + \underbrace{\frac{\boxed{\text{eps}}}{2h} |f(x)|}_{\text{round-off error}} = \alpha h^2 + \frac{\beta}{h} \boxed{\text{eps}} =: g(h).$$

Using calculus, one can find that $g(h)$ is minimized when

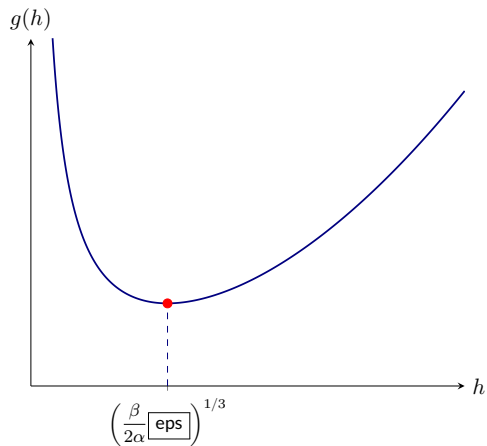
$$h = \left(\frac{\beta}{2\alpha} \boxed{\text{eps}} \right)^{1/3} = \left(\frac{3}{2} \left| \frac{f(x)}{f'''(x)} \right| \right)^{1/3} \boxed{\text{eps}}^{1/3} = O(\boxed{\text{eps}}^{1/3}),$$

in which case

$$\left| \hat{D}_h^{[2c]} \{f\}(x) - f'(x) \right| \lesssim \left(\frac{9}{32} |f^2(x) f'''(x)| \right)^{1/3} \boxed{\text{eps}}^{2/3} = O(\boxed{\text{eps}}^{2/3}).$$

Exercise. Repeat the analysis above to determine the optimal h for a first-order accurate difference formula. How about a general n th-order method?

Determining Optimal h (cont')



The effect of round-off error on numerical differentiation. For sufficiently small h , the error is dominated by rounding error.

Richardson Extrapolation

Set-up. Let V_h be a numerical approximation to the analytical value V such that

$$V_h = V + \underbrace{c_1 h^{p_1} + c_2 h^{p_2} + c_3 h^{p_3} + \cdots}_{\text{discretization error}},$$

where $p_1 < p_2 < p_3 < \cdots$ are positive integers and c_1, c_2, c_3, \dots are constant. Here h is some discretization of the analytical calculation, and as $h \rightarrow 0$, $V_h \rightarrow V$.

Goal. Construct a higher-order accurate method approximating V out of a lower-order accurate method V_h .

Procedure. (*Richardson extrapolation*) Form a suitable linear combination of V_h and $V_{h/2}$ approximating V which removes the leading error term h^{p_1} .

$$\text{Result: } \frac{2^{p_1}}{2^{p_1} - 1} V_{h/2} - \frac{1}{2^{p_1} - 1} V_h = V + O(h^{p_2})$$

Richardson Extrapolation: Illustration of Method

Write down

$$V_h = V + c_1 h^{p_1} + c_2 h^{p_2} + c_3 h^{p_3} + \dots \quad (1)$$

$$V_{h/2} = V + c_1 \left(\frac{h}{2}\right)^{p_1} + c_2 \left(\frac{h}{2}\right)^{p_2} + c_3 \left(\frac{h}{2}\right)^{p_3} + \dots \quad (2)$$

Multiplying (1) by a and (2) by b and summing up, we obtain

$$aV_h + bV_{h/2} = \underbrace{(a+b)}_{=1} V + c_1 \underbrace{\left[a + \frac{b}{2^{p_1}}\right]}_{=0} h^{p_1} + c_2 \left[a + \frac{b}{2^{p_2}}\right] h^{p_2} + \dots$$

Find a and b satisfying

$$\begin{cases} a + b = 1 \\ a + \frac{b}{2^{p_1}} = 0 \end{cases} \implies \begin{cases} a = -\frac{1}{2^{p_1} - 1} \\ b = \frac{2^{p_1}}{2^{p_1} - 1} \end{cases}$$

Another Derivation of 2nd-Order Forward Difference

Exercise. Derive $D_h^{[2f]}\{f\}(x)$ using Richardson extrapolation on $V_h = D_h^{[1f]}\{f\}(x)$.