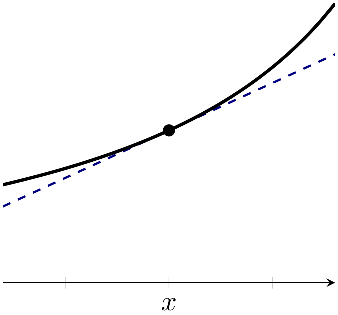
Numerical Differentiation

Contents

Introduction

Difference Quotients to Approximate Slopes



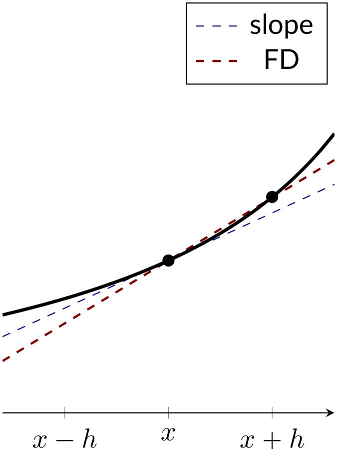
Let f be a smooth function. Analytically, the derivative is calculated by

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

slope of line connecting $(x, f(x))$ and $(x+h, f(x+h))$

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 $f'(x)$.

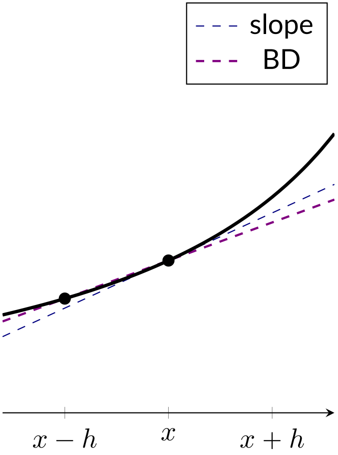
- 1st-order forward difference

difference accuracy type

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

step size

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

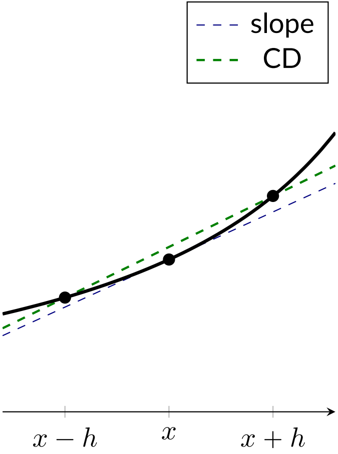
$$D_h^{[1f]}\{f\}(x) = \frac{f(x+h) - f(x)}{h}$$

- 1st-order backward difference

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

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

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

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

- 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

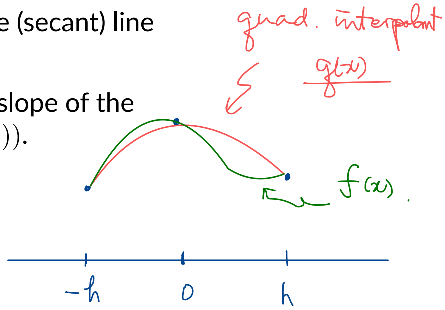
The terms *first-order* and *second-order* refer to how quickly the approximation converges to the actual value of $f'(x)$ as h approaches 0, not to the order of differentiation. More on this later.

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 $(h, f(h))$.

Side note (Lagrange polynomials)

- $l_1(x_1) = 1$, $l_1(x_2) = l_1(x_3) = 0$
- $l_2(x_1) = l_2(x_3) = 0$, $l_2(x_2) = 1$
- $l_3(x_1) = l_3(x_2) = 0$, $l_3(x_3) = 1$

↳ cf) Hat functions in PL interp.

Exercise 1. Show that the quadratic function

$x_1 = -h$

$x_2 = 0$

$x_3 = h$

$$q(x) = \underbrace{\frac{x(x-h)}{2h^2}}_{=l_1(x)} f(-h) - \underbrace{\frac{x^2-h^2}{h^2}}_{=l_2(x)} f(0) + \underbrace{\frac{x(x+h)}{2h^2}}_{=l_3(x)} f(h).$$

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

Ans

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

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

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

$$q'(0) = \frac{-\cancel{h} f(-h) + \cancel{h} f(h)}{2h^{\cancel{2}}} = \frac{f(h) - f(-h)}{2h} = D_h^{[2c]}\{f\}(0)$$

Interpolation and Difference Formulas (cont')

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

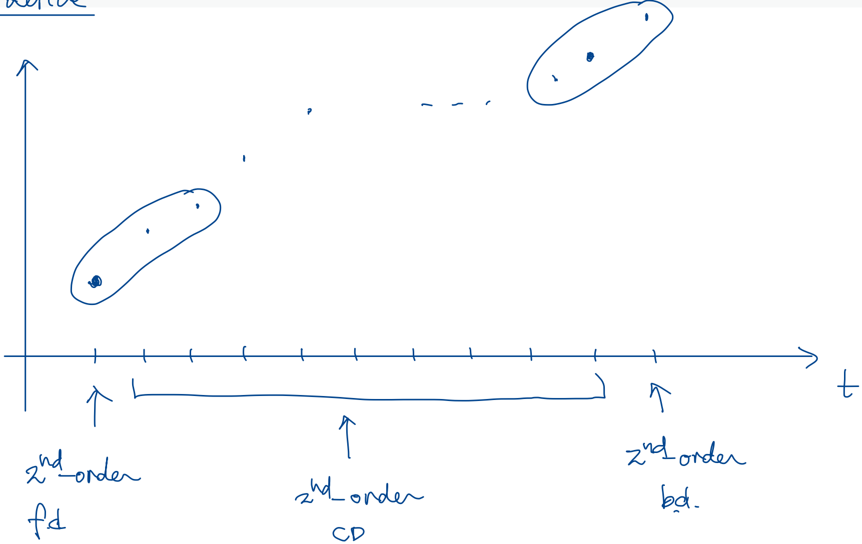
> *Interpolate the given function values, then differentiate the interpolant exactly.*

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

Common Difference Formulas

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

Practice



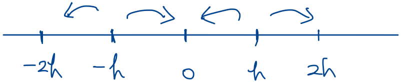
Higher Derivatives

For example, consider approximation of $f''(0)$.

<u>One way</u> (Repeated use of CD)

$$f''(0) \overset{\text{CD on } f'}{\approx} \frac{f'(h) - f'(-h)}{2h}$$

$$\overset{\text{CD on } f}{\approx} \frac{\dfrac{f(2h) - f(0)}{2h} - \dfrac{f(0) - f(-2h)}{2h}}{2h}$$



$$= \frac{f(2h) - 2f(0) + f(-2h)}{4h^2} \xrightarrow{h \to \frac{h}{2}} \boxed{f''(0) \approx \frac{f(h) - 2f(0) + f(-h)}{h^2}}$$

Better way (Interpolation-based)

Recall the quad. interpolant of $(0, f(0))$, $(\pm h, f(\pm h))$:

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

We use $q''(0)$ to approximate $f''(0)$:

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

Same as above!

Convergence of Difference Formulas

Convergence of Difference Formulas

- All finite difference formulas introduced converge as $h \to 0$.
- 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 $h \to 0$.
- In other words, we need to study how *quickly* the error $D^{[\cdot]}\{f\}(x) - f'(x)$ diminishes as $h \to 0$.
- The main tool for the analysis is the Taylor series.

First-Order Difference Formulas

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

$$\overbrace{D_h^{[1\mathrm{f}]}\{f\}(x)}^{\text{approx}} - \overbrace{f'(x)}^{\text{exact}} = \underbrace{\frac{1}{2} f''(x) h^{1}}_{\text{leading error}} + \underbrace{O(h^2)}_{\text{higher-order terms}}.$$

error

HOT

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

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

Second-Order Difference Formulas

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

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

$h^2,\quad h^4,\quad h^6,\quad h^8,\quad \ldots$

Derivation. Exercise.

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(x)}{6}h^3 + \cdots$$

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - \frac{f'''(x)}{6}h^3 + \cdots$$

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 nth-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 factoriztion.) 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.

2nd-order

$h \to h/2$

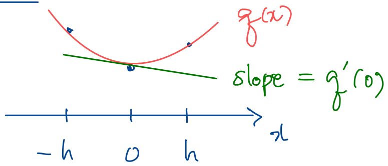
"error" $\to$ $\frac{\text{"error"}}{4}$

MATLAB Demo

Comparison of convergence.

Recap Centered Difference

Given: data



Want: approximate the rate of change of "y" values at $x=0$.

(i.e., $f'(0)$)

Interpolation-based approach

① Interpolate the given data using a quad. polynomial ($g(x)$)

② Differentiate $g(x)$.

$$f'(0) \approx g'(0)$$

Some (messy) useful detail

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

Cardinal functions

(Lagrange polynomials)

$$g'(0) = \frac{f(h) - f(-h)}{2h} = D_h^{[2c]}\{f\}(0)$$

Determining Optimal h

The difference formulas are inherently ill-conditioned as $h \to 0$ due to catastrophic cancellation when implemented in floating point arithmetic. As an example, consider the numerical evaluation[1] of the centered difference formula:

relative errors.

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

$$= \frac{f(x+h)(1+\epsilon_+) - f(x-h)(1+\epsilon_-)}{2h}, \quad \text{for some } |\epsilon_\pm| \leqslant \frac{1}{2}\boxed{\text{eps}} \approx 1.1 \times 10^{-16}$$

exact CD

$$= \boxed{\frac{f(x+h) - f(x-h)}{2h}} + \frac{f(x+h)\epsilon_+ - f(x-h)\epsilon_-}{2h}$$

Taylor expand: $f(x) \pm f'(x)h + \frac{f''(x)}{2}h^2 + \cdots$

$$= \left[f'(x) + \frac{1}{6}f'''(x)h^2 + O(h^4)\right] + \frac{(f(x) + O(h))\,\epsilon_+ - (f(x) + O(h))\,\epsilon_-}{2h}$$

$$= f'(x) + \frac{1}{6}f'''(x)h^2 + \frac{\epsilon_+ - \epsilon_-}{2h}f(x) + O(\boxed{\text{eps}}) + O(h^4).$$

$\frac{O(h)}{2h}\epsilon_\pm = O(1)\epsilon_\pm = O(\boxed{\text{eps}})$

So the error is bounded from above by:

$$\left|\widehat{D}_h^{[2c]}\{f\}(x) - f'(x)\right| \leqslant \frac{1}{6}\left|f'''(x)\right| h^2 + \frac{\boxed{\text{eps}}}{2h}\left|f(x)\right| + O(\boxed{\text{eps}}) + O(h^4).$$

[1] For 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:

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

$$g'(h) = 2\alpha h - \frac{\beta\boxed{\text{eps}}}{h^2} = 0$$

$$2\alpha h = \frac{\beta\boxed{\text{eps}}}{h^2}$$

$$2\alpha h^3 = \beta\boxed{\text{eps}}$$

$$h = \sqrt[3]{\frac{\beta\boxed{\text{eps}}}{2\alpha}}$$

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

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

$\approx 10^{-5}$ or 10^{-6}

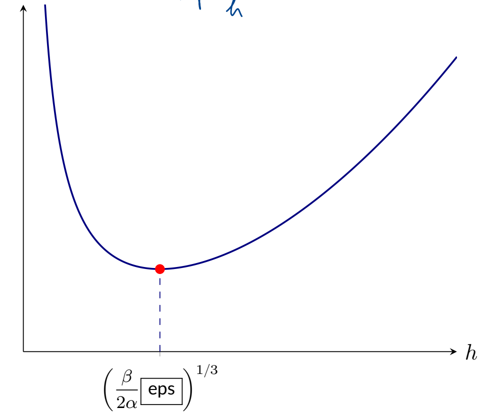
in which case

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

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

Determining Optimal h (cont')

$$g(h) = \alpha h^2 + \beta \frac{\boxed{\text{eps}}}{h}$$



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

where round-off ($\frac{1}{h}$) dominates | where truncation error (h^2) dominates

Richardson Extrapolation

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

leading error

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

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

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}.

or V_{2h}

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

Richardson Extrapolation: Illustration of Method

Write down

$$V_h = V + c_1 h^{p_1} + c_2 h^{p_2} + c_3 h^{p_3} + \cdots \tag{1}$$

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

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

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

Find a and b satisfying

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

Another Derivation of 2nd-Order Forward Difference

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

Recall:

$$\underbrace{D_h^{[1f]}\{f\}(x)}_{=V_h} = \underbrace{f'(x)}_{=V} + \underbrace{\frac{f''(x)}{2}}_{=c_1} h + \underbrace{\frac{f'''(x)}{6}}_{=c_2} h^2 + \cdots$$

Form a linear combo. of V_h and V_{2h} to approx. V w/ h^2-leading error.

$$2V_h = 2V + 2c_1 h + 2c_2 h^2 + \cdots$$

$$-)\ V_{2h} = V + 2c_1 h + 4c_2 h^2 + \cdots$$

$$\boxed{2V_h - V_{2h}} = V + 0 \cdot h + (-2c_2) h^2 + \cdots$$

new, better

The new algorithm is :

$$2V_h - V_{2h} = 2\,\frac{f(x+h) - f(x)}{h} - \frac{f(x+2h) - f(x)}{2h}$$

$$= \frac{4f(x+h) - 4f(x) - f(x+2h) + f(x)}{2h}$$

$$= \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} = D_h^{[2f]}\{f\}(x)$$

Confirmed !