Computational Physics (PHYS6350)

Lecture 7: Numerical Derivatives

$$\frac{\mathrm{d}f}{\mathrm{d}x} \simeq \frac{f(x+h)-f(x)}{h}$$

February 9, 2023

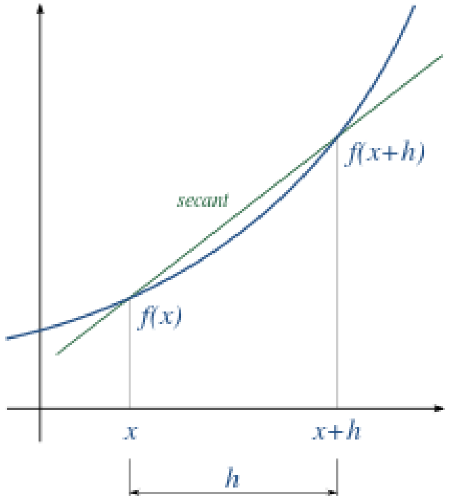
Instructor: Volodymyr Vovchenko (vvovchenko@uh.edu)

Course materials: https://github.com/vlvovch/PHYS6350-ComputationalPhysics

Numerical differentiation

Generic problem: evaluate

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}$$



We need numerical differentiation when

- Function f is known at a discrete set of points
- Too expensive/cumbersome to do directly
  - E.g. when f(x) itself is a solution to a complex web of non-linear equations, calculating f'(x) explicitly will require rewriting all the equations

References: Chapter 5 of *Computational Physics* by Mark Newman

Forward difference

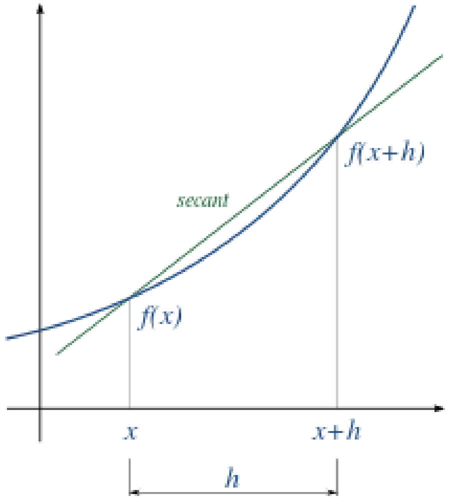
Simply approximate

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

by

$$\frac{\mathrm{d}f}{\mathrm{d}x} \simeq \frac{f(x+h) - f(x)}{h}$$

where h is finite



Taylor theorem:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \ldots$$

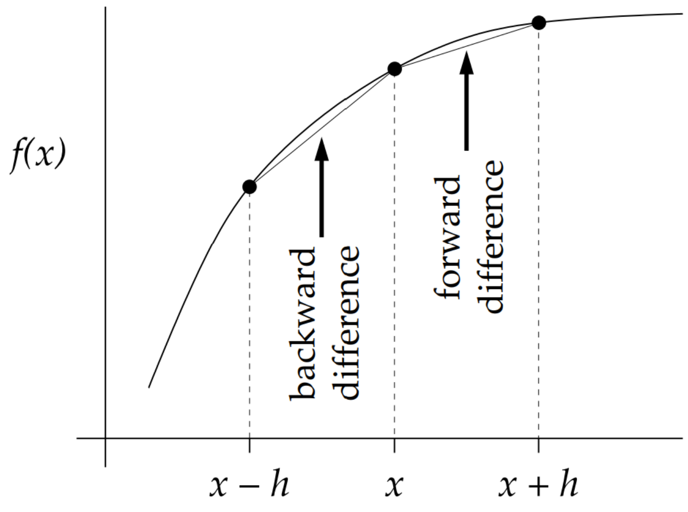
gives the approximation error estimate of

$$R_{\mathrm{forw}} = -\frac{1}{2} h f''(x) + \mathcal{O}(h^2)$$

Backward difference

Backward difference

$$\frac{\mathrm{d}f}{\mathrm{d}x} \simeq \frac{f(x) - f(x-h)}{h}$$



Taylor theorem:

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

gives the approximation error estimate of

$$R_{\text{back}} = \frac{1}{2} h f''(x) + \mathcal{O}(h^2)$$

Central difference

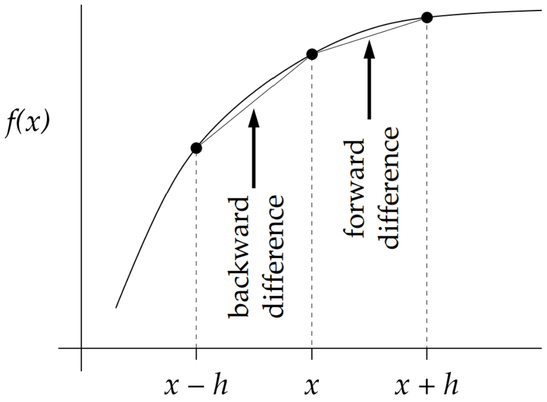
Recall the forward and backward difference and their errors

$$\frac{\mathrm{d}f}{\mathrm{d}x} \simeq \frac{f(x+h) - f(x)}{h}$$

$$R_{\text{forw}} = -\frac{1}{2} h f''(x) + \mathcal{O}(h^2)$$

$$\frac{\mathrm{d}f}{\mathrm{d}x} \simeq \frac{f(x) - f(x-h)}{h}$$

$$R_{\text{back}} = \frac{1}{2} h f''(x) + \mathcal{O}(h^2)$$



Taking the average of the two cancels out the *O(h)* error

central difference $$\frac{df}{dx} \simeq \frac{f(x+h) - f(x-h)}{2h}$$

Error estimate:

$$R_{\text{cent}} = -\frac{f'''(x)}{6} h^2 + \mathcal{O}(h^3)$$

High-order central difference

To improve the approximation error use more than two function evaluations, e.g.

$$\frac{df}{dx} \simeq \frac{Af(x-2h)+Bf(x-h)+Cf(x)+Df(x+h)+Ef(x+2h)}{h} + \mathcal{O}(h^4)$$

Determine A, B, C, D, E using Taylor expansion to cancel all terms up to h^4

$$\frac{df}{dx} \simeq \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12h} + \frac{h^4}{30} f^{(5)}(x)$$

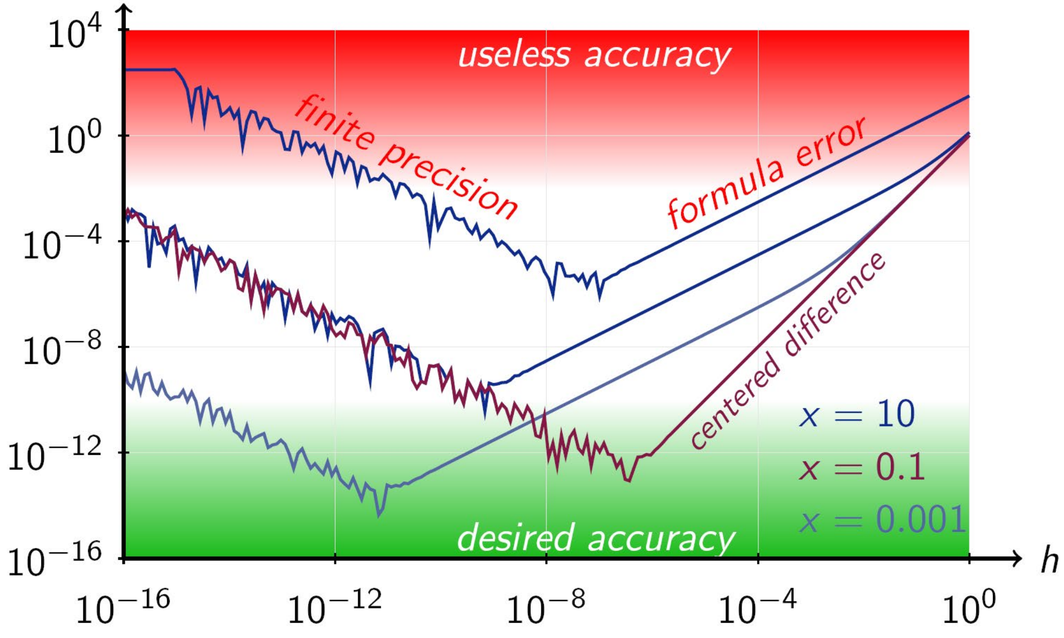
High-order terms:

Derivative	Accuracy	−5	−4	−3	−2	−1	0	1	2	3	4	5
1	2					−1/2	0	1/2				
	4				1/12	−2/3	0	2/3	−1/12			
	6			−1/60	3/20	−3/4	0	3/4	−3/20	1/60		
	8		1/280	−4/105	1/5	−4/5	0	4/5	−1/5	4/105	−1/280	

Balancing truncation and round-off errors

If h is too small, round-off errors become important

- cannot distringuish x and x+h and/or $f(x+h)$ and $f(x)$



Credit: Wikipedia

As a rule of thumb, if ε is machine precision and the truncation error is of order $O(h^n)$, then h should not be much smaller than $h \sim \sqrt[n+1]{\varepsilon}$

The higher the finite difference order is, the larger h should be

Balancing truncation and round-off errors

Let f(x) = exp(x)

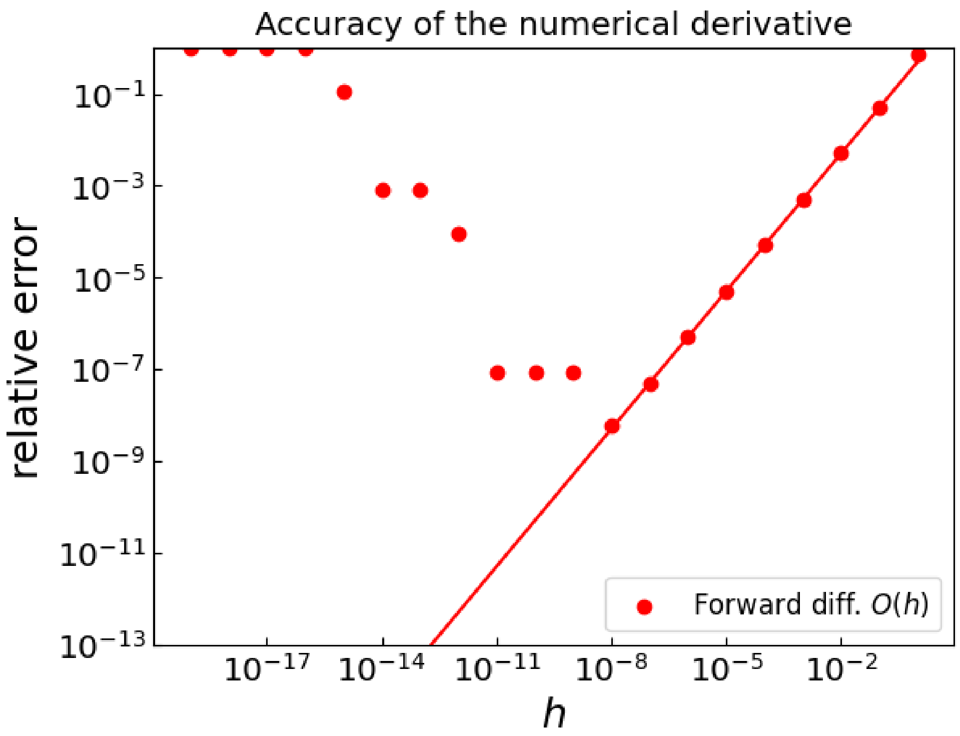
Calculate the derivatives at x = 0

```python
def f(x):
    return np.exp(x)

def df(x):
    return np.exp(x)
```

Forward difference O(h):

Optimal $h \sim \sqrt[2]{10^{-16}} \sim 10^{-8}$



Balancing truncation and round-off errors

Let f(x) = exp(x)

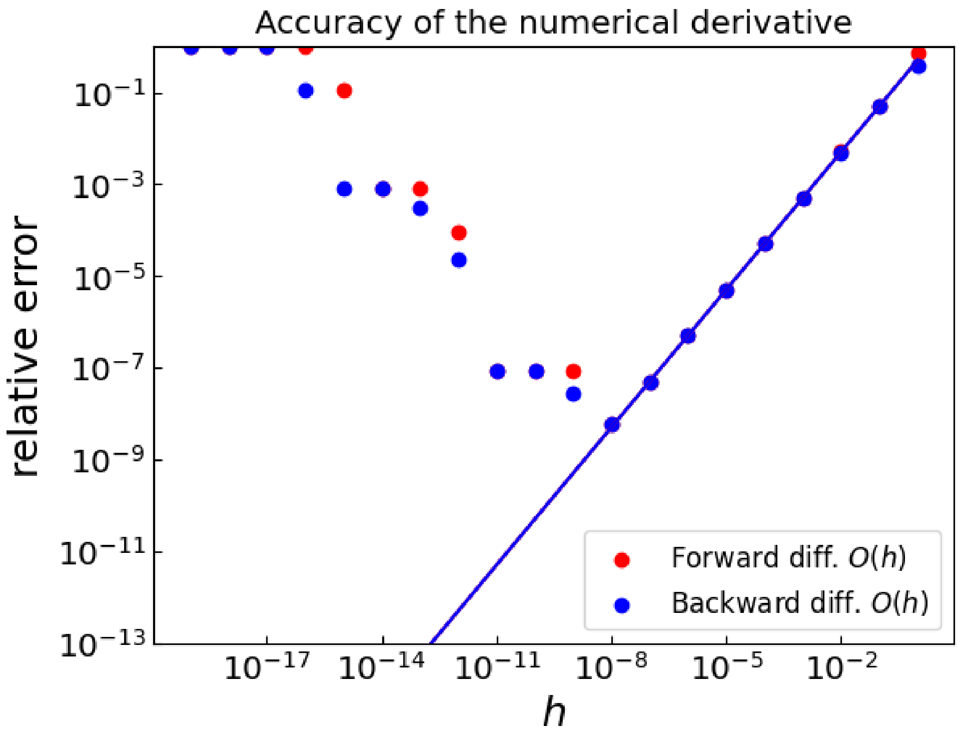
Calculate the derivatives at x = 0

```python
def f(x):
    return np.exp(x)

def df(x):
    return np.exp(x)
```

Backward difference O(h):

Optimal $h \sim \sqrt[2]{10^{-16}} \sim 10^{-8}$



Balancing truncation and round-off errors

Let f(x) = exp(x)

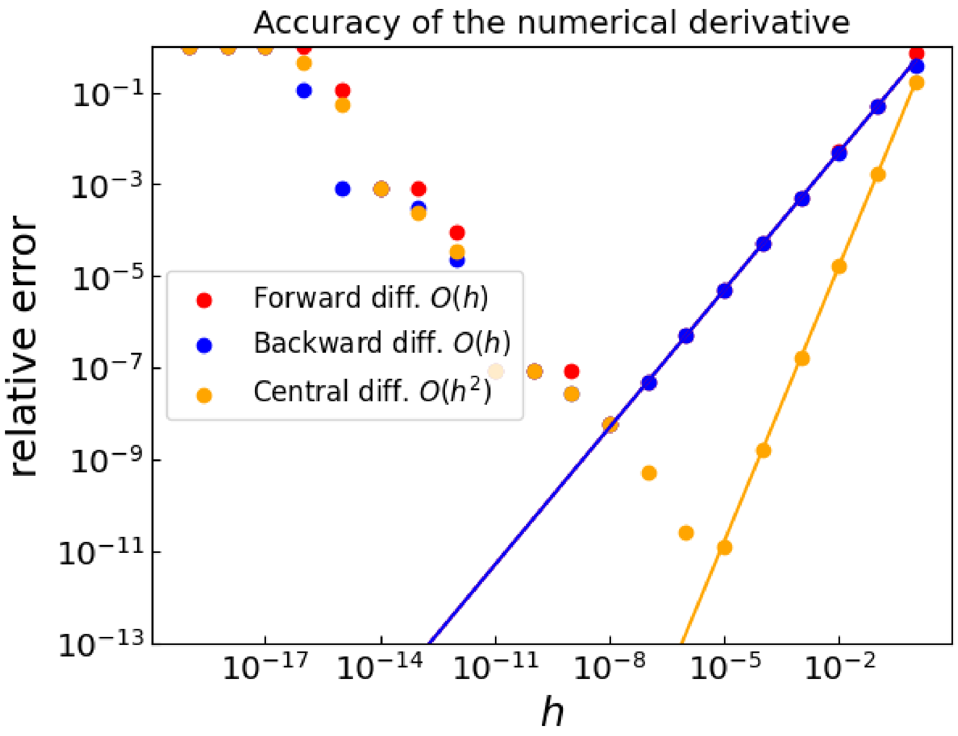
Calculate the derivatives at x = 0

```python
def f(x):
    return np.exp(x)

def df(x):
    return np.exp(x)
```

Central difference O(h²):

Optimal $h \sim \sqrt[3]{10^{-16}} \sim 10^{-5}$



Balancing truncation and round-off errors

Let f(x) = exp(x)

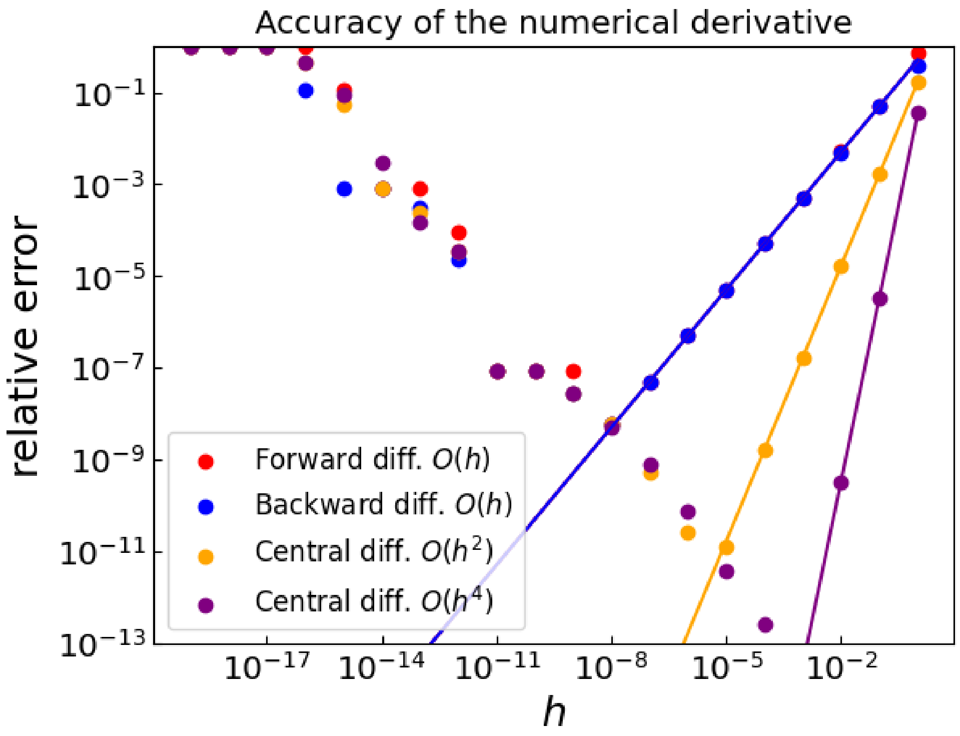
Calculate the derivatives at x = 0

```python
def f(x):
    return np.exp(x)

def df(x):
    return np.exp(x)
```

Central difference O(h⁴):

Optimal $h \sim \sqrt[5]{10^{-16}} \sim 10^{-3}$



High-order derivatives

Central difference

$$\frac{df}{dx}(x) \simeq \frac{f(x+h/2) - f(x-h/2)}{h}$$

Now apply central difference again to *f'(x+h/2)* and *f'(x-h/2)*

$$\begin{aligned} f''(x) &\simeq \frac{f'(x+h/2) - f'(x-h/2)}{h} \\ &= \frac{[f(x+h) - f(x)]/h - [f(x) - f(x-h)]/h}{h} \\ &= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \end{aligned}$$

General formula (to order h^2)

$$f^{(n)}(x) = \frac{1}{h^n} \sum_{k=0}^{n} (-1)^{k+n} \binom{n}{k} f(x+kh)$$

Second derivative

```python
def d2f_central(f,x,h):
    return (f(x+h) - 2*f(x) + f(x-h)) / (h**2)
```

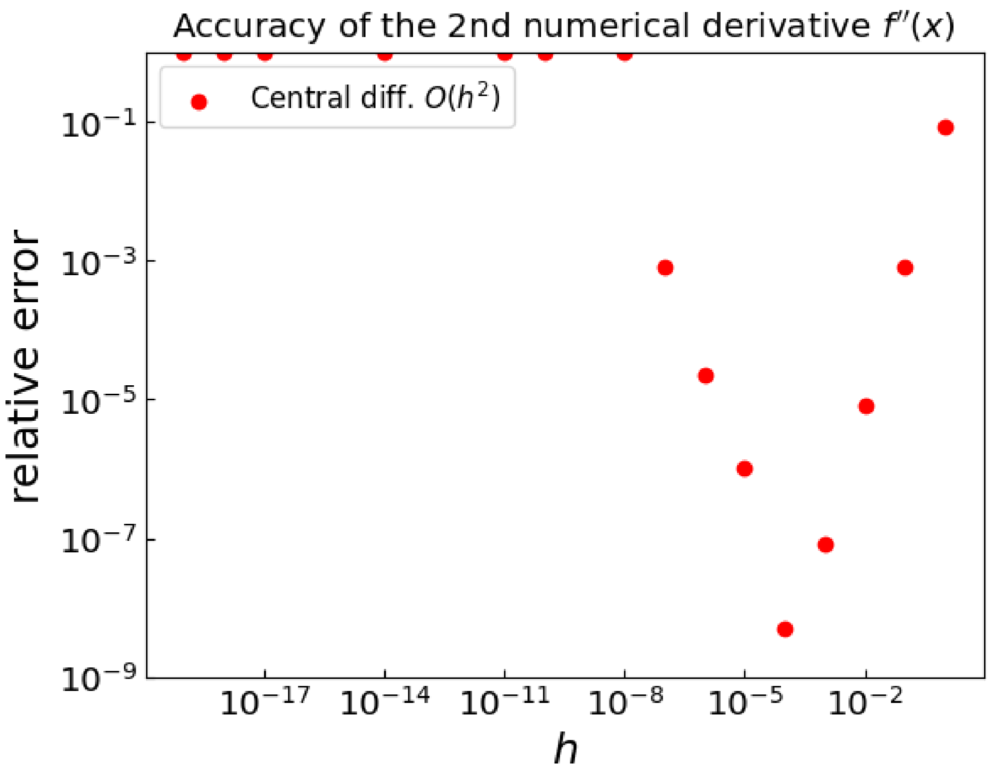
f(x) = exp(x)

```python
def f(x):
    return np.exp(x)

def df(x):
    return np.exp(x)

def d2f(x):
    return np.exp(x)
```

Optimal $h \sim \sqrt[4]{10^{-16}} \sim 10^{-4}$



Partial derivatives

Let us have *f(x,y)*
Use central difference to calculate first-order derivatives

$$\frac{\partial f}{\partial x} = \frac{f(x+h/2,y) - f(x-h/2,y)}{h}$$
$$\frac{\partial f}{\partial y} = \frac{f(x,y+h/2) - f(x,y-h/2)}{h}$$

Reapply the central difference to calculate $\partial^2 f(x,y)/\partial x \partial y$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{f(x+h/2,y+h/2) - f(x-h/2,y+h/2) - f(x+h/2,y-h/2) + f(x-h/2,y-h/2)}{h^2}$$

Summary: Numerical differentiation

- Forward/backward differences
  - Useful when we are given a grid of function values
  - Have limited accuracy (linear in h)

- Central difference
  - More precise than forward/backward differences (quadratic in h)
  - Gives f'(x) estimate at the midpoint of function evaluation points

- Higher-order formulas are obtained by using more than two function evaluations
  - Can be used when limited number of function evaluations available

- Straightforwardly extendable to high-order and partial derivatives

- Balance between truncation and round-off error has to be respected
  - h should not be taken too small