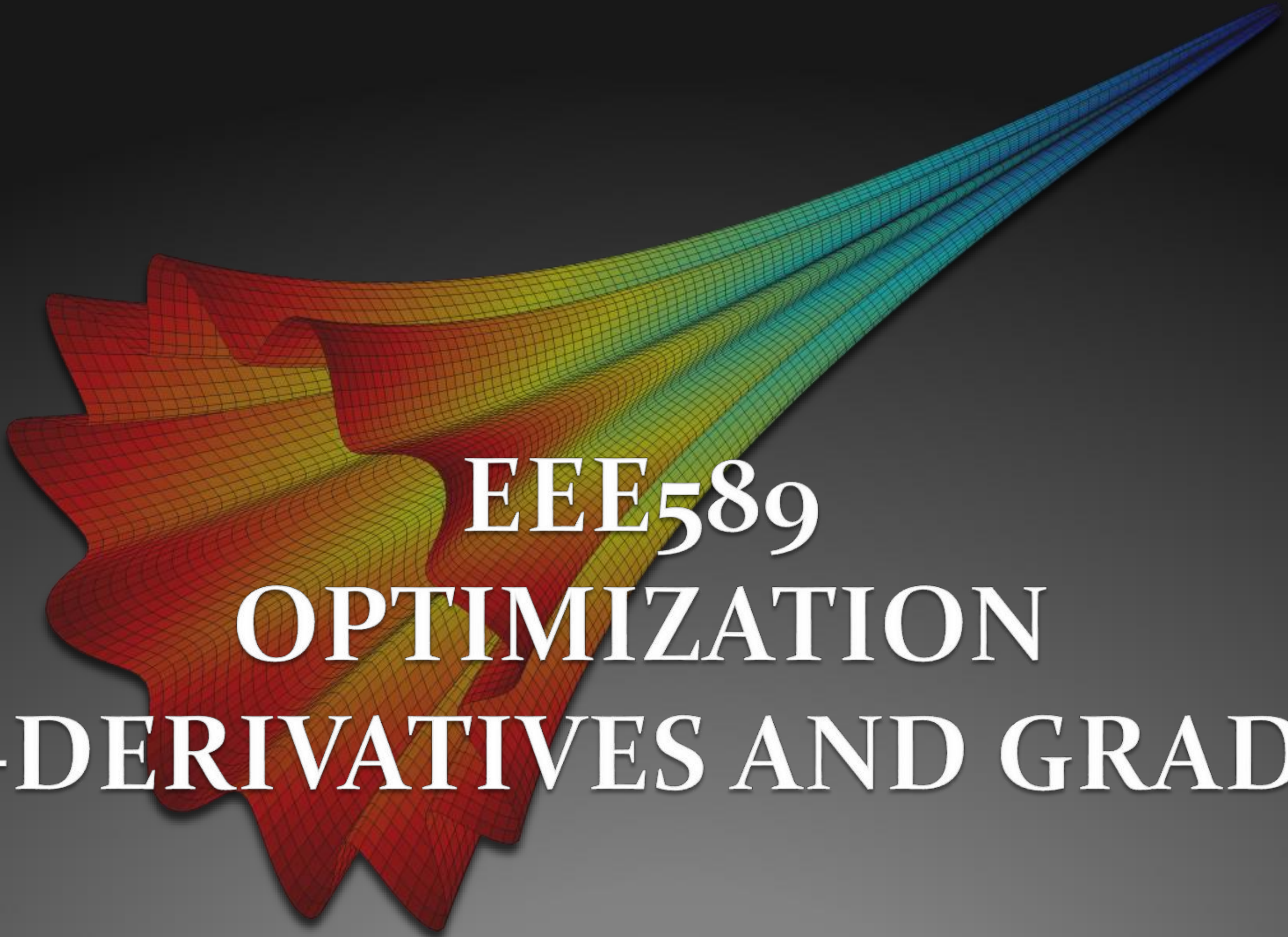


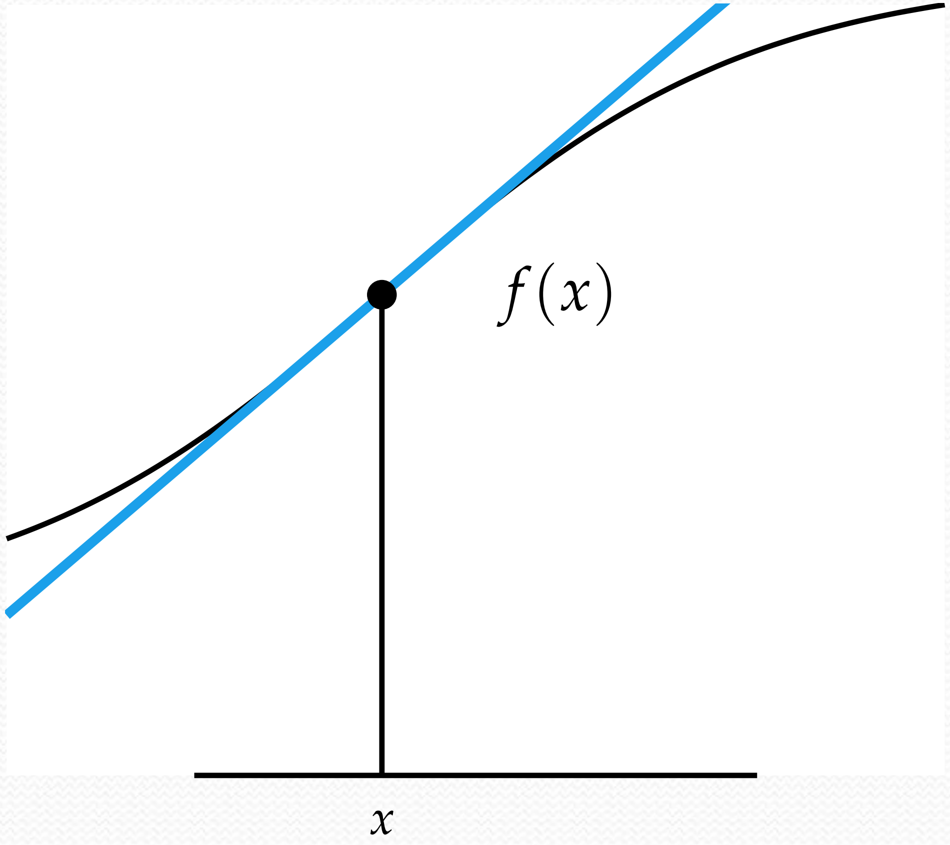
EEE589
OPTIMIZATION
CH II –DERIVATIVES AND GRADIENTS

Introduction

- Optimization is concerned with finding the design point that minimizes (or maximizes) an objective function.
- Knowing how the value of a function changes as its input is varied is useful because it tells us in which direction we can move to improve on previous points.
- The change in the value of the function is measured by the derivative in one dimension and the gradient in multiple dimensions.
- In this lecture, we will briefly review some essential elements from calculus.

Derivatives

- The *derivative* f ′(x) of a function f of a single variable x is the rate at which the value of f changes at x. It is often visualized using the tangent line to the graph of the function at x as shown in figure. The value of the derivative equals the slope of the tangent line.



Derivatives

- We can use the derivative to provide a linear approximation of the function near x:

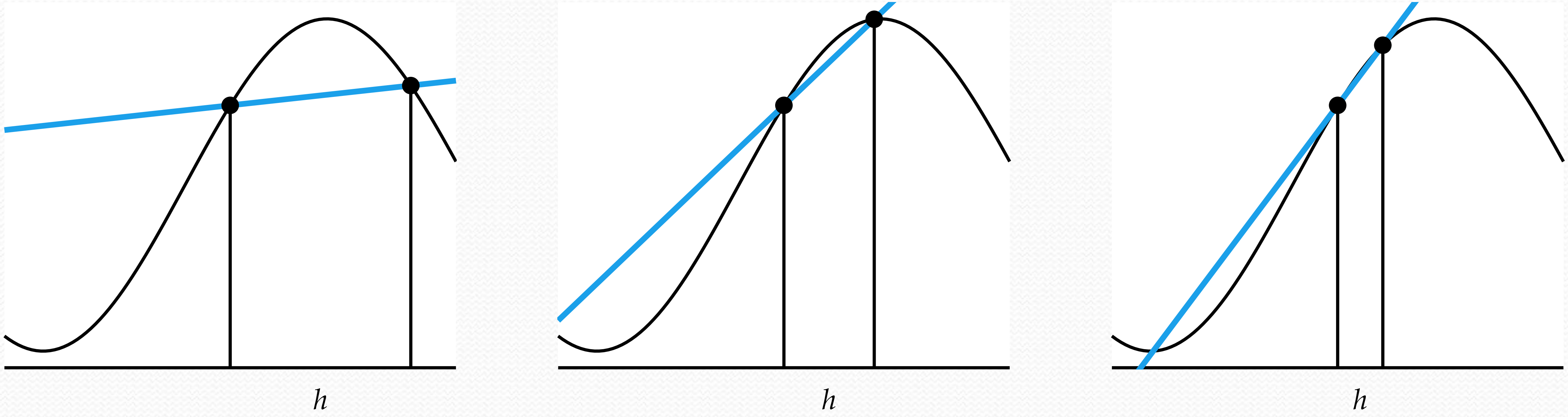
$$f(x+\Delta x) \approx f(x) + f'(x)\Delta x$$

- The derivative is the ratio between the change in f and the change in x at the point x:

$$f'(x) = \frac{\Delta f(x)}{\Delta x}$$

Derivatives

- $f'(x)$ is the change in $f(x)$ divided by the change in x as the step becomes infinitesimally small as illustrated by figure.



The tangent line is obtained by joining points with sufficiently small step differences.

Derivatives

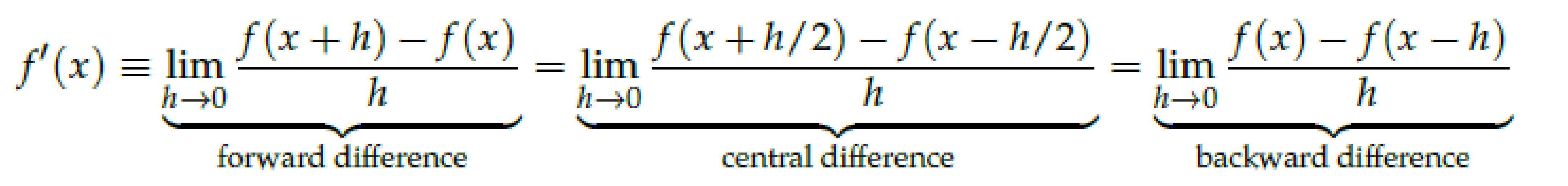
- The notation f ′(x) can be attributed to Lagrange. We also use the notation created by Leibniz,

$$f'(x) \equiv \frac{df(x)}{dx}$$

which emphasizes the fact that the derivative is the ratio of the change in f to the change in x at the point x.

Derivatives

- The limit equation defining the derivative can be presented in three different ways: the *forward difference*, the *central difference*, and the *backward difference*. Each method uses an infinitely small step size h:



$$f'(x) \equiv \underbrace{\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}}_{\text{forward difference}} = \underbrace{\lim_{h \to 0} \frac{f(x+h/2) - f(x-h/2)}{h}}_{\text{central difference}} = \underbrace{\lim_{h \to 0} \frac{f(x) - f(x-h)}{h}}_{\text{backward difference}}$$

- If f can be represented symbolically, *symbolic differentiation* can often provide an exact analytic expression for f ′ by applying derivative rules from calculus. The analytic expression can then be evaluated at any point x.

Derivatives in Multiple Dimensions

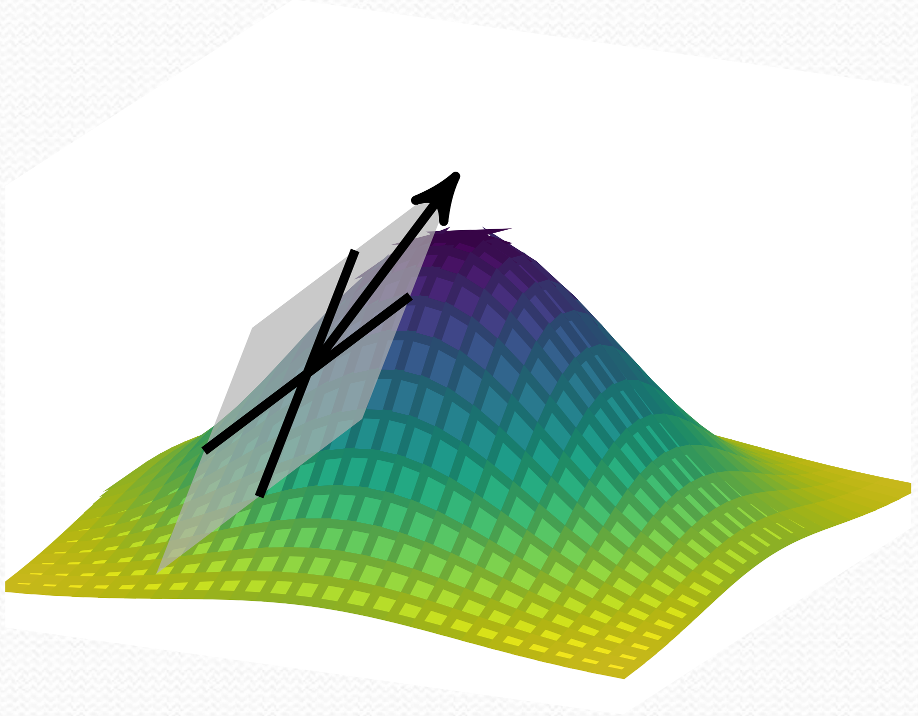
- The *gradient* is the generalization of the derivative to multivariate functions. It captures the local slope of the function, allowing us to predict the effect of taking a small step from a point in any direction.

- Recall that the derivative is the slope of the tangent line. The gradient points in the direction of steepest ascent of the tangent *hyperplane* as shown in figure.

- A hyperplane in an n-dimensional space is the set of points that satisfies

$$w1x1 + \cdot \cdot \cdot + wnxn = b$$

for some vector **w** and scalar b. A hyperplane has n – 1 dimensions.

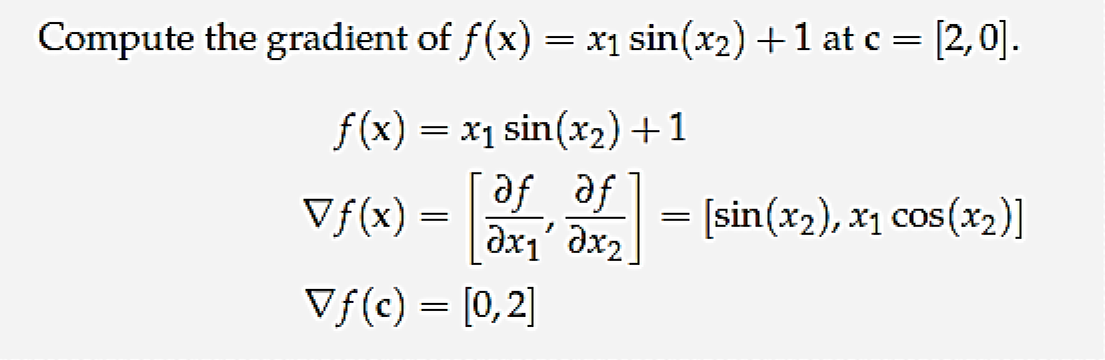


Gradient Vector

- The gradient of f at **x** is written ∇f (**x**) and is a vector. Each component of that vector is the *partial derivative* of f with respect to that component:

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \quad \frac{\partial f(\mathbf{x})}{\partial x_2}, \quad \ldots, \quad \frac{\partial f(\mathbf{x})}{\partial x_n}\right]$$

- We use the convention that vectors written with commas are column vectors. For example, we have [a, b, c] = [a b c]⊤. Example shows how to compute the gradient of a function at a particular point.



Compute the gradient of $f(\mathrm{x}) = x_1 \sin(x_2) + 1$ at $\mathrm{c} = [2, 0]$.

$$\begin{aligned} f(\mathrm{x}) &= x_1 \sin(x_2) + 1 \\ \nabla f(\mathrm{x}) &= \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right] = [\sin(x_2), x_1 \cos(x_2)] \\ \nabla f(\mathrm{c}) &= [0, 2] \end{aligned}$$

Hessian Matrix

- The *Hessian* of a multivariate function is a matrix containing all of the second derivatives with respect to the input. The second derivatives capture information about the local curvature of the function.

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ & \vdots & & \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{bmatrix}$$

Directional Derivative

- The *directional derivative* ∇**s** f (**x**) of a multivariate function f is the instantaneous rate of change of f (**x**) as **x** is moved with velocity **s**. The definition is closely related to the definition of a derivative of a univariate function:

$$\nabla_{\mathbf{s}} f(\mathbf{x}) \equiv \underbrace{\lim_{h\to 0} \frac{f(\mathbf{x}+h\mathbf{s}) - f(\mathbf{x})}{h}}_{\text{forward difference}} = \underbrace{\lim_{h\to 0} \frac{f(\mathbf{x}+h\mathbf{s}/2) - f(\mathbf{x}-h\mathbf{s}/2)}{h}}_{\text{central difference}} = \underbrace{\lim_{h\to 0} \frac{f(\mathbf{x}) - f(\mathbf{x}-h\mathbf{s})}{h}}_{\text{backward difference}}$$

- The directional derivative can be computed using the gradient of the function:

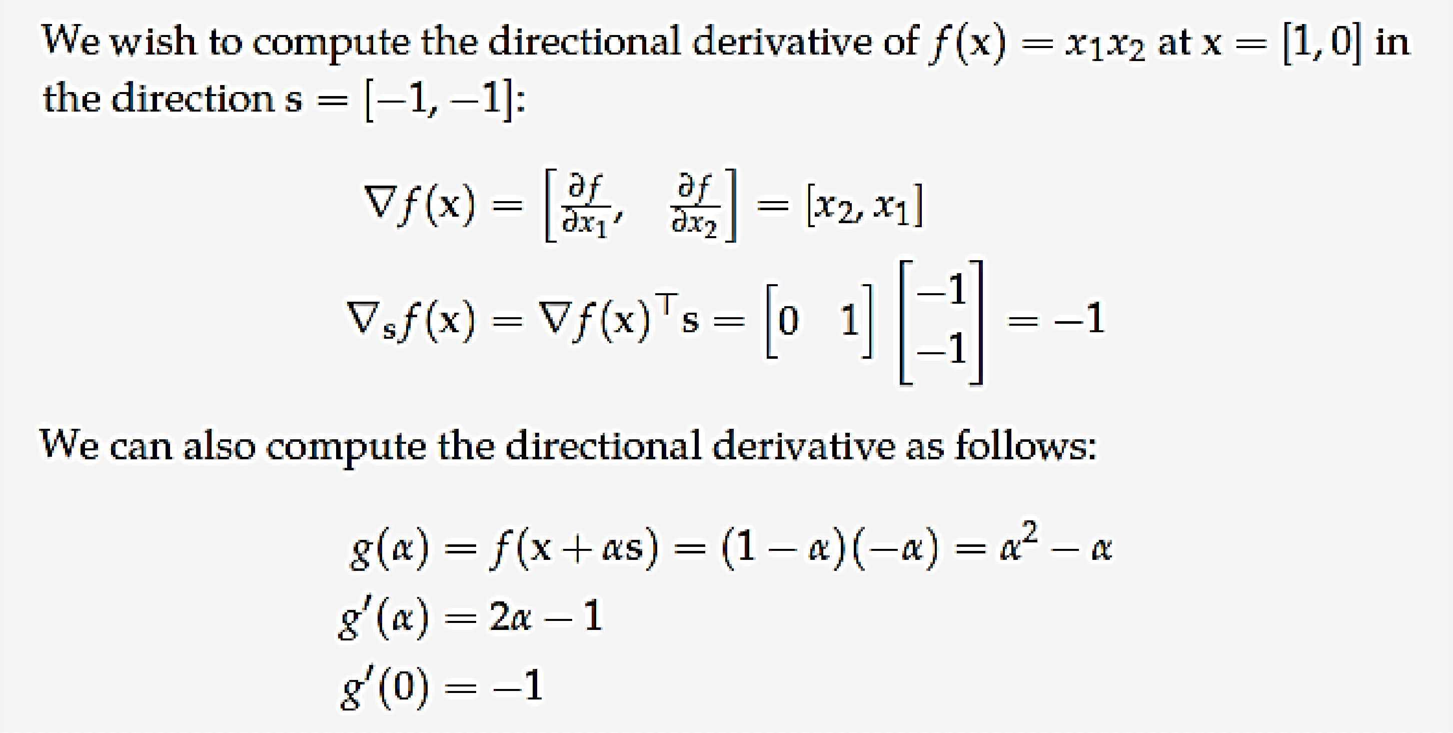
$$\nabla_{\mathbf{s}} f(\mathbf{x}) = \nabla f(\mathbf{x})^{\top} \mathbf{s}$$

Directional Derivative

- The directional derivative is highest in the gradient direction, and it is lowest in the direction opposite the gradient. This directional dependence arises from the dot product in the directional derivative’s definition and from the fact that the gradient is a local tangent hyperplane.

Directional Derivative

- Another way to compute the directional derivative $\nabla\mathbf{s}\ f\ (\mathbf{x})$ is to define $g(\alpha) \equiv f\ (\mathbf{x} + \alpha\mathbf{s})$ and then compute $g'(0)$, as illustrated in example.



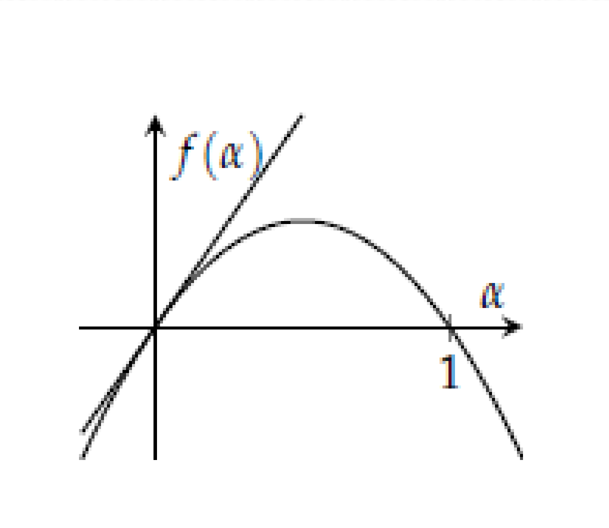
We wish to compute the directional derivative of $f(\mathrm{x}) = x_1 x_2$ at $\mathrm{x} = [1, 0]$ in the direction $\mathrm{s} = [-1, -1]$:

$$\nabla f(\mathrm{x}) = \left[\frac{\partial f}{\partial x_1}, \quad \frac{\partial f}{\partial x_2}\right] = [x_2, x_1]$$

$$\nabla_{\mathrm{s}} f(\mathrm{x}) = \nabla f(\mathrm{x})^\top \mathrm{s} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} = -1$$

We can also compute the directional derivative as follows:

$$\begin{aligned} g(\alpha) &= f(\mathrm{x} + \alpha\mathrm{s}) = (1 - \alpha)(-\alpha) = \alpha^2 - \alpha \\ g'(\alpha) &= 2\alpha - 1 \\ g'(0) &= -1 \end{aligned}$$



Numerical Differentiation

- The process of estimating derivatives numerically is referred to as *numerical differentiation*. Estimates can be derived in different ways from function evaluations.

  - *Finite Difference Methods*

  - *Complex Step Method*

Finite Difference Methods

- As the name implies, *finite difference methods* compute the difference between two values that differ by a finite step size. They approximate the derivative definitions in equation using small differences:

$$f'(x) \approx \underbrace{\frac{f(x+h)-f(x)}{h}}_{\text{forward difference}} \approx \underbrace{\frac{f(x+h/2)-f(x-h/2)}{h}}_{\text{central difference}} \approx \underbrace{\frac{f(x)-f(x-h)}{h}}_{\text{backward difference}}$$

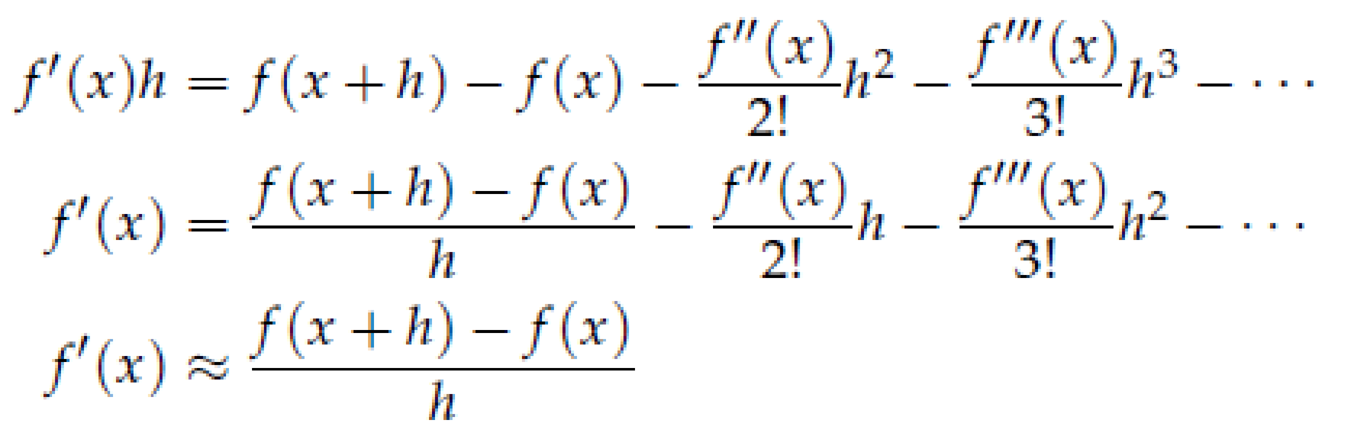
- Mathematically, the smaller the step size h, the better the derivative estimate. Practically, values for h that are too small can result in numerical cancellation errors. This effect is shown later.

Finite Difference Methods

- The finite difference methods can be derived using the Taylor expansion. We will derive the forward difference derivative estimate, beginning with the Taylor expansion of f about x:

$$f(x+h) = f(x) + \frac{f'(x)}{1!}h + \frac{f''(x)}{2!}h^2 + \frac{f'''(x)}{3!}h^3 + \cdots$$

We can rearrange and solve for the first derivative:



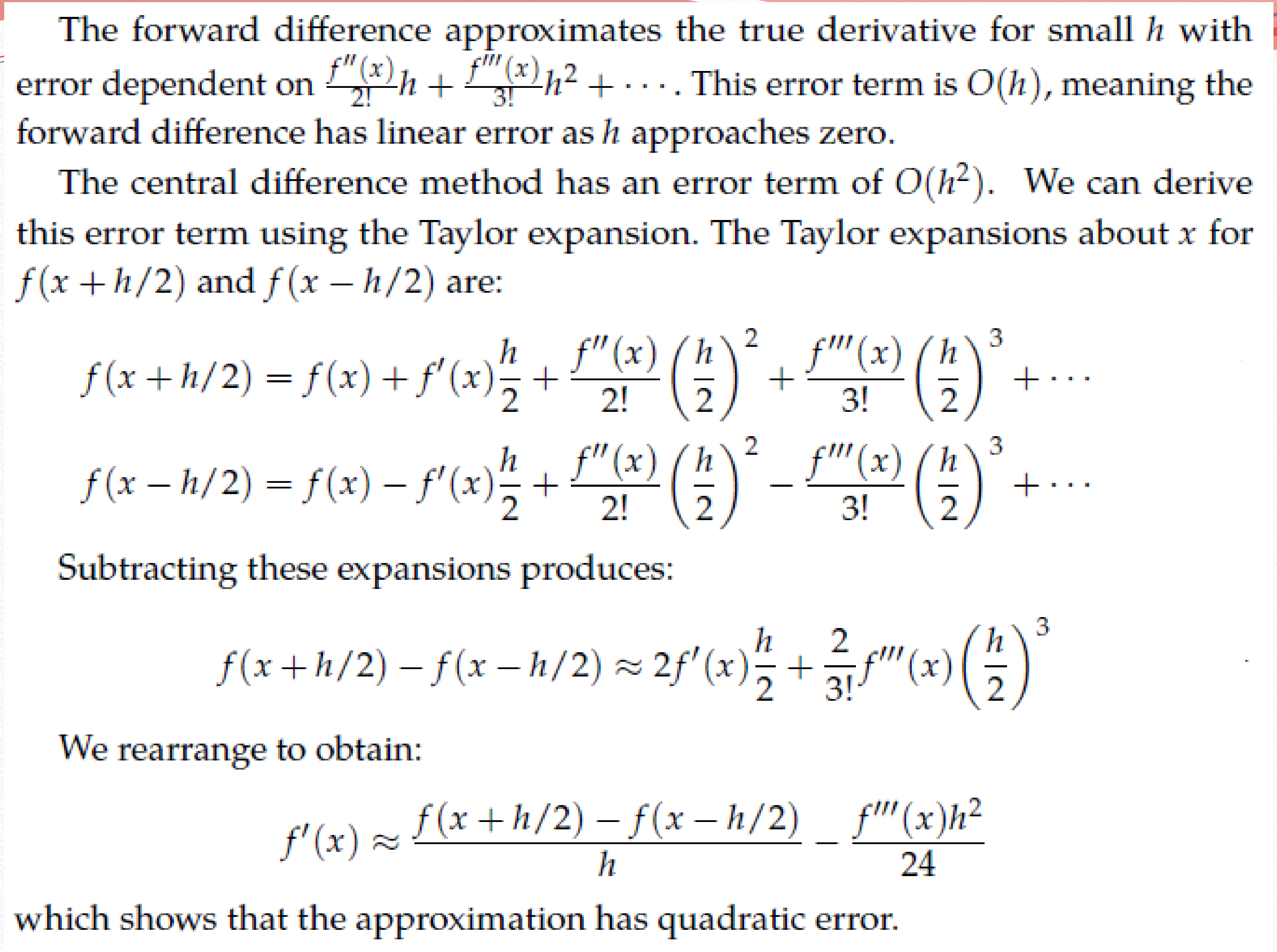
$$f'(x)h = f(x+h) - f(x) - \frac{f''(x)}{2!}h^2 - \frac{f'''(x)}{3!}h^3 - \cdots$$

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

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Finite Difference Methods

- *Error Analysis*
  - *Forward Difference: O(h)*
  - *Central Difference: O(h^2)*



The forward difference approximates the true derivative for small h with error dependent on $\frac{f''(x)}{2!}h + \frac{f'''(x)}{3!}h^2 + \cdots$. This error term is $O(h)$, meaning the forward difference has linear error as h approaches zero.

The central difference method has an error term of $O(h^2)$. We can derive this error term using the Taylor expansion. The Taylor expansions about x for $f(x+h/2)$ and $f(x-h/2)$ are:

$$f(x+h/2) = f(x) + f'(x)\frac{h}{2} + \frac{f''(x)}{2!}\left(\frac{h}{2}\right)^2 + \frac{f'''(x)}{3!}\left(\frac{h}{2}\right)^3 + \cdots$$

$$f(x-h/2) = f(x) - f'(x)\frac{h}{2} + \frac{f''(x)}{2!}\left(\frac{h}{2}\right)^2 - \frac{f'''(x)}{3!}\left(\frac{h}{2}\right)^3 + \cdots$$

Subtracting these expansions produces:

$$f(x+h/2) - f(x-h/2) \approx 2f'(x)\frac{h}{2} + \frac{2}{3!}f'''(x)\left(\frac{h}{2}\right)^3$$

We rearrange to obtain:

$$f'(x) \approx \frac{f(x+h/2) - f(x-h/2)}{h} - \frac{f'''(x)h^2}{24}$$

which shows that the approximation has quadratic error.

Complex Step Methods

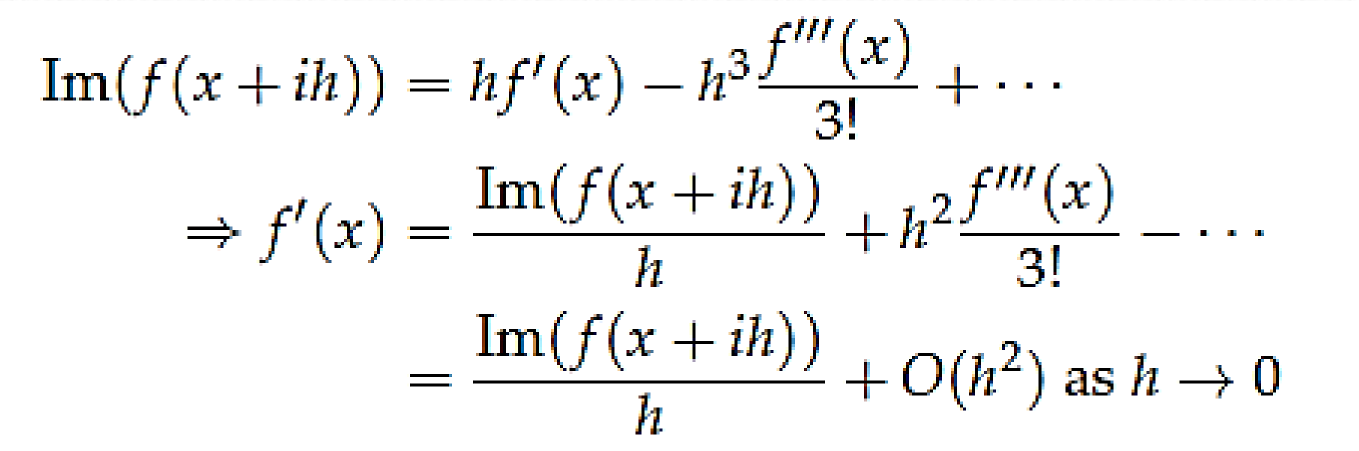
- We often run into the problem of needing to choose a step size h small enough to provide a good approximation but not too small so as to lead to numerical subtractive cancellation issues.
- The *complex step method* bypasses the effect of subtractive cancellation by using a single function evaluation.
- We evaluate the function once after taking a step in the imaginary direction.

Complex Step Methods

- *Taylor series expansion using imaginary step is*

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

- Taking only the imaginary component of each side produces a derivative approximation:



$$\begin{aligned} \mathrm{Im}(f(x+ih)) &= hf'(x) - h^3\frac{f'''(x)}{3!} + \cdots \\ \Rightarrow f'(x) &= \frac{\mathrm{Im}(f(x+ih))}{h} + h^2\frac{f'''(x)}{3!} - \cdots \\ &= \frac{\mathrm{Im}(f(x+ih))}{h} + O(h^2) \text{ as } h \to 0 \end{aligned}$$

Complex Step Methods

- The real part approximates f (x) to within O(h^2) as h → 0:

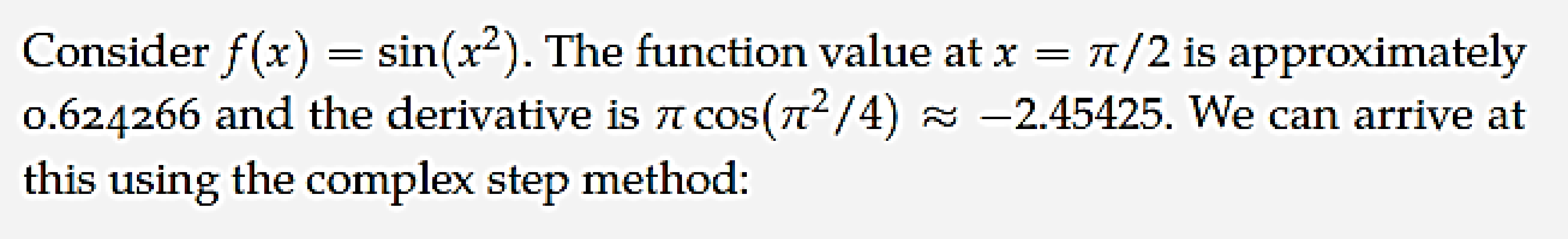
$$\mathrm{Re}(f(x+ih)) = f(x) - h^2\frac{f''(x)}{2!} + \dots$$

$$\Rightarrow f(x) = \mathrm{Re}(f(x+ih)) + h^2\frac{f''(x)}{2!} - \cdots$$

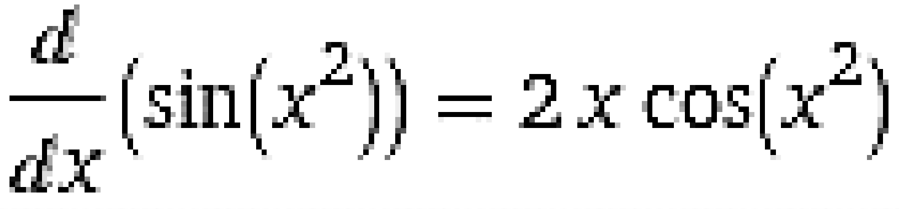
- Thus, we can evaluate both f (x) and f ′(x) using a single evaluation of f with complex arguments.

Complex Step Methods

- Example shows the calculations involved for estimating the derivative of a function at a particular point.



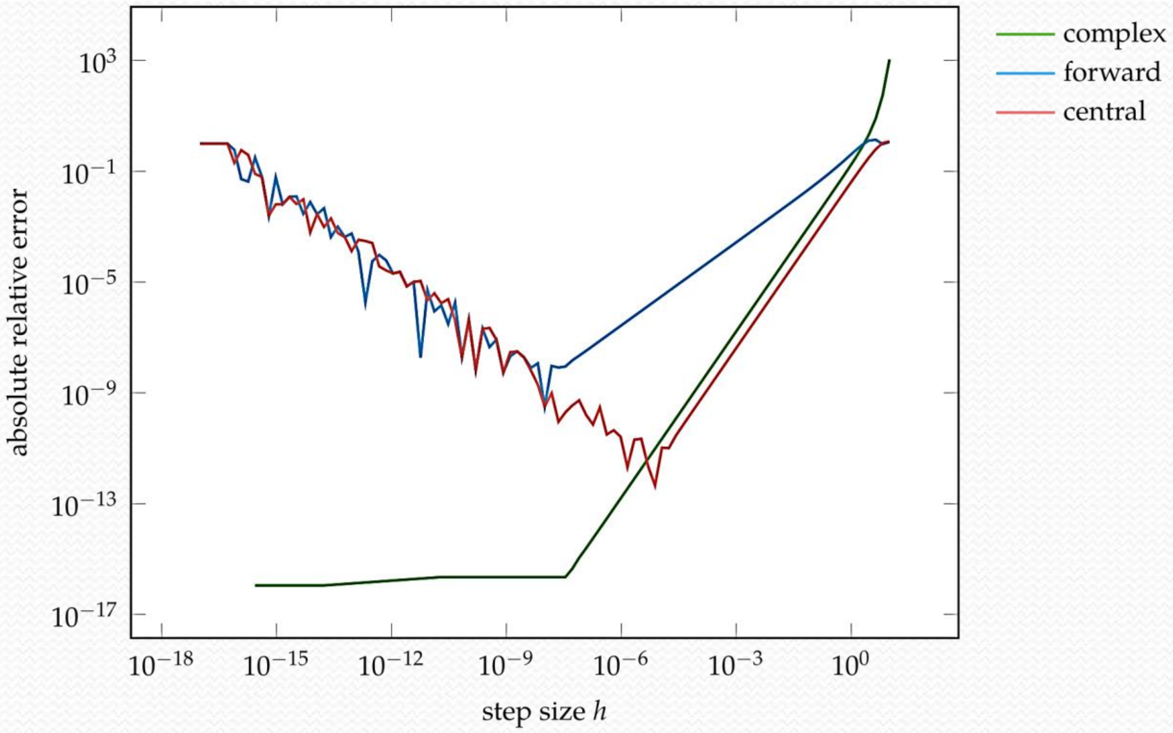
Consider $f(x) = \sin(x^2)$. The function value at $x = \pi/2$ is approximately 0.624266 and the derivative is $\pi \cos(\pi^2/4) \approx -2.45425$. We can arrive at this using the complex step method:



$$\frac{d}{dx}\left(\sin\left(x^2\right)\right) = 2\,x\cos\left(x^2\right)$$

Numerical Differentiation Error Comparison

- Figure compares the numerical error of the complex step method to the forward and central difference methods as the step size is varied.



Automatic Differentiation

- This section introduces algorithms for the numeric evaluation of derivatives of functions specified by a computer program. Key to these *automatic differentiation* techniques is the application of the chain rule:

$$\frac{d}{dx} f(g(x)) = \frac{d}{dx}(f \circ g)(x) = \frac{df}{dg}\frac{dg}{dx}$$

- A program is composed of elementary operations like addition, subtraction, multiplication, and division.

Automatic Differentiation

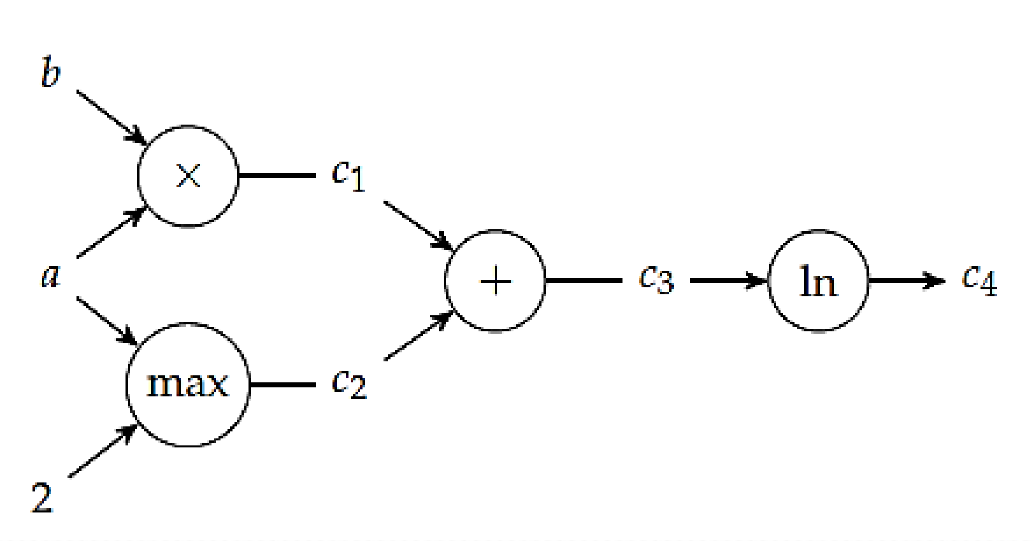
- Consider the function f (a, b) = ln(ab + max(a, 2)). If we want to compute the partial derivative with respect to **a** at a point, we need to apply the chain rule several times:

$$\frac{d}{da}(\max(a, 2)) = \begin{cases} 0 & a < 2 \\ 1 & a > 2 \\ \text{indeterminate} & \text{(otherwise)} \end{cases}$$

$$\begin{aligned}
\frac{\partial f}{\partial a} &= \frac{\partial}{\partial a}\ln(ab + \max(a,2)) \\
&= \frac{1}{ab + \max(a,2)}\frac{\partial}{\partial a}(ab + \max(a,2)) \\
&= \frac{1}{ab + \max(a,2)}\left[\frac{\partial(ab)}{\partial a} + \frac{\partial \max(a,2)}{\partial a}\right] \\
&= \frac{1}{ab + \max(a,2)}\left[\left(b\frac{\partial a}{\partial a} + a\frac{\partial b}{\partial a}\right) + \left((2 > a)\frac{\partial 2}{\partial a} + (2 < a)\frac{\partial a}{\partial a}\right)\right] \\
&= \frac{1}{ab + \max(a,2)}[b + (2 < a)]
\end{aligned}$$

Automatic Differentiation

- This process can be automated through the use of a *computational graph*. A computational graph represents a function where the nodes are operations and the edges are input-output relations. The leaf nodes of a computational graph are input variables or constants, and terminal nodes are values output by the function. A computational graph for ln(ab + max(a, 2)) is shown in figure.



Automatic Differentiation

- There are two methods for automatically differentiating f using its computational graph.

  - The *forward accumulation* method used by dual numbers traverses the tree from inputs to outputs.

  - *The reverse accumulation* requires a backwards pass through the graph.

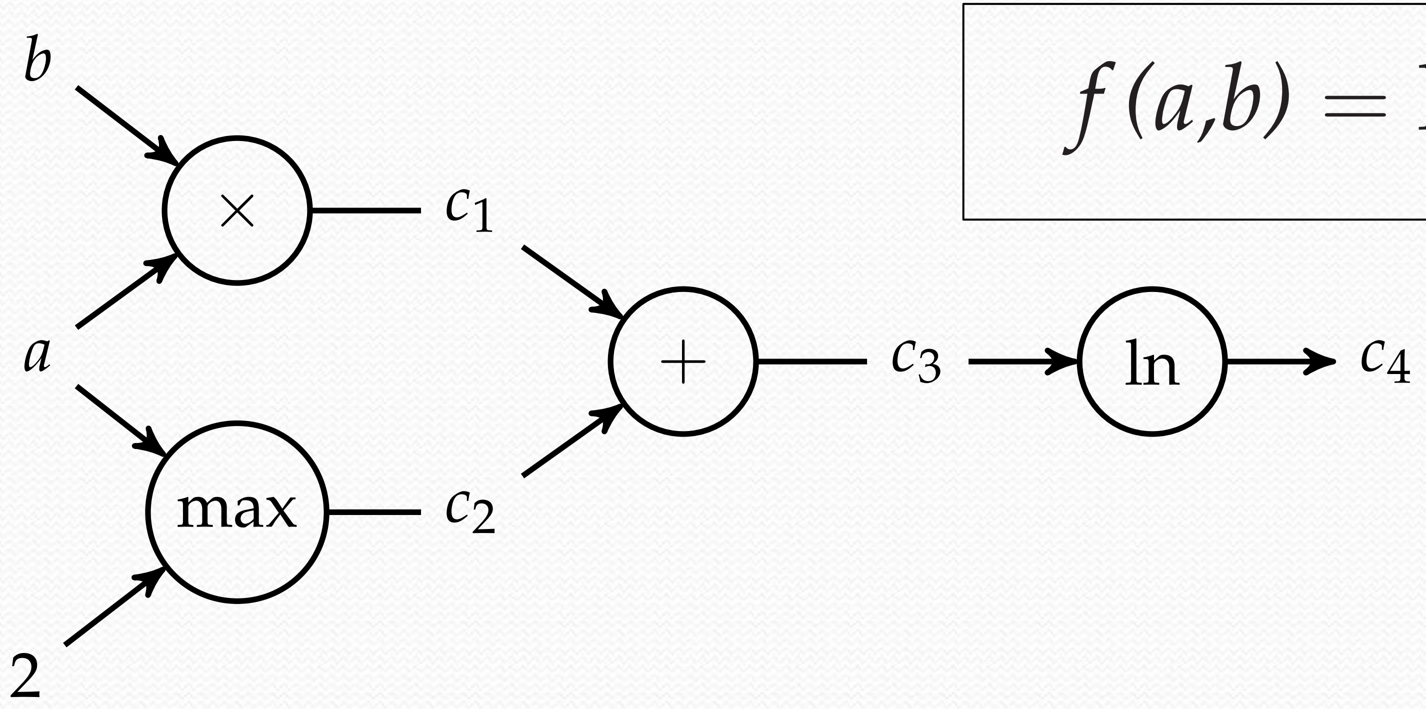
Forward Accumulation

- *Forward accumulation* will automatically differentiate a function using a single forward pass through the function's computational graph. The method is equivalent to iteratively expanding the chain rule of the inner operation:

$$\frac{df}{dx} = \frac{df}{dc_4}\frac{dc_4}{dx} = \frac{df}{dc_4}\left(\frac{dc_4}{dc_3}\frac{dc_3}{dx}\right) = \frac{df}{dc_4}\left(\frac{dc_4}{dc_3}\left(\frac{dc_3}{dc_2}\frac{dc_2}{dx} + \frac{dc_3}{dc_1}\frac{dc_1}{dx}\right)\right)$$

Forward Accumulation

- To illustrate forward accumulation, we apply it to the example function f (a, b) = ln(ab + max(a, 2)) to calculate the partial derivative at a = 3, b = 2 with respect to a.



$$\boxed{f(a,b) = \ln(\,ab + \max(a,2))}$$

Forward Accumulation

- The procedure starts at the graph’s source nodes consisting of the function inputs and any constant values. For each of these nodes, we note both the value and the partial derivative with respect to our target variable, as shown in figure.

$b = 2$

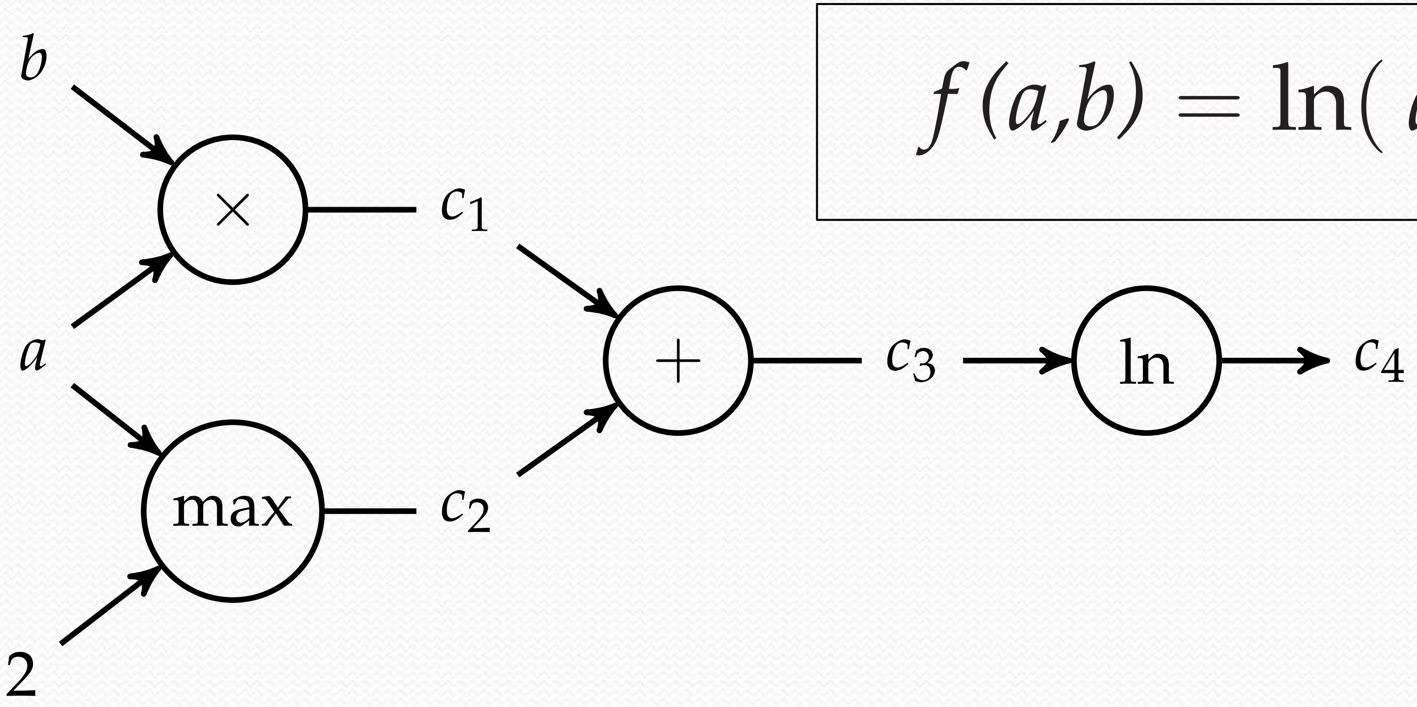
$\dot{b} = 0$

$\frac{\partial b}{\partial a}$

$a = 3$

$\dot{a} = 1$

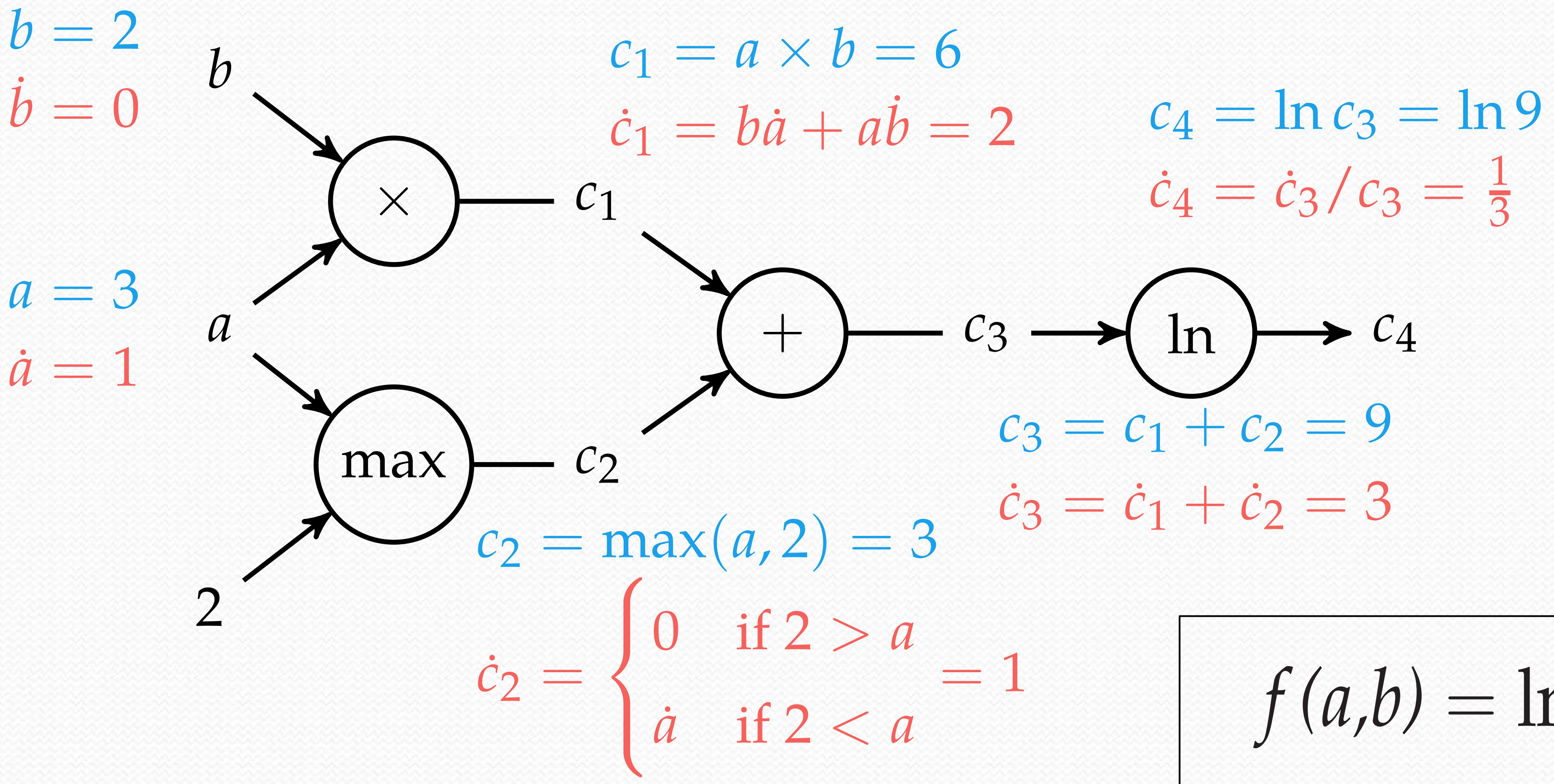
$\frac{\partial a}{\partial a}$



$$\boxed{f(a,b) = \ln(ab + \max(a, 2))}$$

Forward Accumulation

- Next we proceed down the tree, one node at a time, choosing as our next node one whose inputs have already been computed. We can compute the value by passing through the previous nodes' values, and we can compute the local partial derivative with respect to a using both the previous nodes' values and their partial derivatives. The calculations are shown in figure.



$$b = 2$$
$$\dot{b} = 0$$
$$a = 3$$
$$\dot{a} = 1$$
$$c_1 = a \times b = 6$$
$$\dot{c}_1 = b\dot{a} + a\dot{b} = 2$$
$$c_2 = \max(a, 2) = 3$$
$$\dot{c}_2 = \begin{cases} 0 & \text{if } 2 > a \\ \dot{a} & \text{if } 2 < a \end{cases} = 1$$
$$c_3 = c_1 + c_2 = 9$$
$$\dot{c}_3 = \dot{c}_1 + \dot{c}_2 = 3$$
$$c_4 = \ln c_3 = \ln 9$$
$$\dot{c}_4 = \dot{c}_3 / c_3 = \frac{1}{3}$$

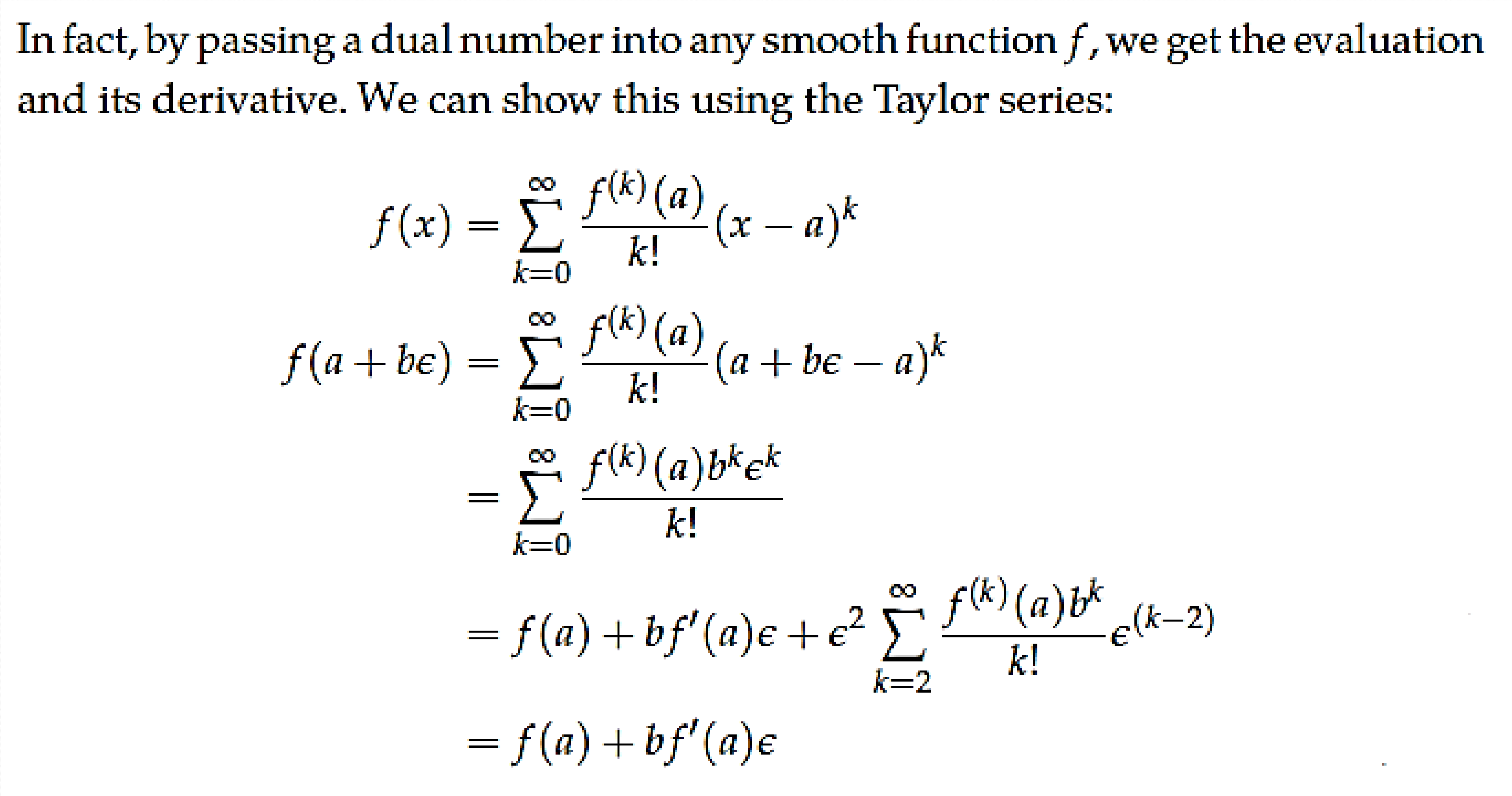
$$f(a,b) = \ln(ab + \max(a, 2))$$

Forward Accumulation

- We end up with the correct result, f (3, 2) = ln 9 and ∂ f /∂a = 1/3. This was done using one pass through the computational graph.
- This process can be conveniently automated by a computer using a programming language which has overridden each operation to produce both the value and its derivative. Such pairs are called *dual numbers*.
- Dual numbers can be expressed mathematically by including the abstract quantity ϵ, where ϵ^2 is defined to be 0. Like a complex number, a dual number is written a + bϵ where a and b are both real values. We have:

$$(a + b\epsilon) + (c + d\epsilon) = (a + c) + (b + d)\epsilon$$
$$(a + b\epsilon) \times (c + d\epsilon) = (ac) + (ad + bc)\epsilon$$

Forward Accumulation



In fact, by passing a dual number into any smooth function f, we get the evaluation and its derivative. We can show this using the Taylor series:

$$
\begin{aligned}
f(x) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!}(x-a)^k \\
f(a+b\epsilon) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!}(a+b\epsilon-a)^k \\
&= \sum_{k=0}^{\infty} \frac{f^{(k)}(a)b^k\epsilon^k}{k!} \\
&= f(a)+bf'(a)\epsilon+\epsilon^2\sum_{k=2}^{\infty} \frac{f^{(k)}(a)b^k}{k!}\epsilon^{(k-2)} \\
&= f(a)+bf'(a)\epsilon
\end{aligned}
$$

Reverse Accumulation

- Forward accumulation requires n passes in order to compute an n-dimensional gradient.
- *Reverse accumulation* requires only a single run in order to compute a complete gradient but requires two passes through the graph: a *forward pass* during which necessary intermediate values are computed and a *backward pass* which computes the gradient.
- Reverse accumulation is often preferred over forward accumulation when gradients are needed, though care must be taken on memory constrained systems when the computational graph is very large.

Reverse Accumulation

- Like forward accumulation, reverse accumulation will compute the partial derivative with respect to the chosen target variable but iteratively substitutes the outer function instead:

$$\frac{df}{dx} = \frac{df}{dc_4}\frac{dc_4}{dx} = \left(\frac{df}{dc_3}\frac{dc_3}{dc_4}\right)\frac{dc_4}{dx} = \left(\left(\frac{df}{dc_2}\frac{dc_2}{dc_3} + \frac{df}{dc_1}\frac{dc_1}{dc_3}\right)\frac{dc_3}{dc_4}\right)\frac{dc_4}{dx}$$

- This process is the reverse pass, the evaluation of which requires intermediate values that are obtained during a forward pass.

Reverse Accumulation

- Reverse accumulation can be implemented through *operation overloading* in a similar manner to the way dual numbers are used to implement forward accumulation.
- Two functions must be implemented for each fundamental operation: a forward operation that overloads the operation to store local gradient information during the forward pass and a backward operation that uses the information to propagate the gradient backwards.

Summary

- Derivatives are useful in optimization because they provide information about how to change a given point in order to improve the objective function

- For multivariate functions, various derivative-based concepts are useful for directing the search for an optimum, including the gradient, the Hessian, and the directional derivative

- One approach to numerical differentiation includes finite difference approximations

Summary

- Complex step method can eliminate the effect of subtractive cancellation error when taking small steps, resulting in high quality gradient estimates.

- Analytic differentiation methods include forward and reverse accumulation on computational graphs