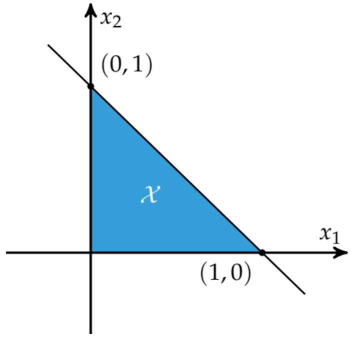
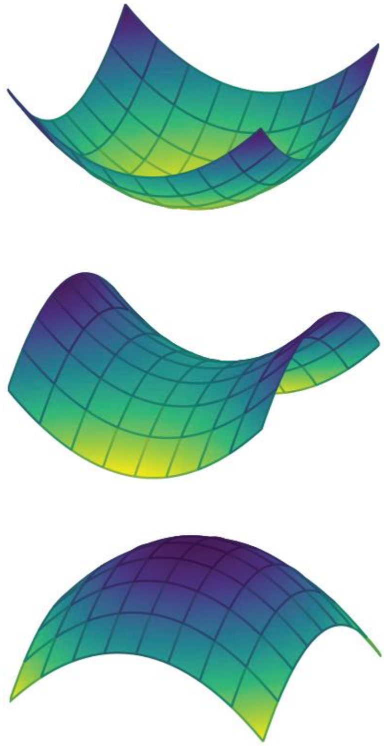
Tecnológico Nacional de México

Instituto Tecnológico de Orizaba/Celaya

Modern Computing Algorithms for Process Optimization with Julia Programming. Part I



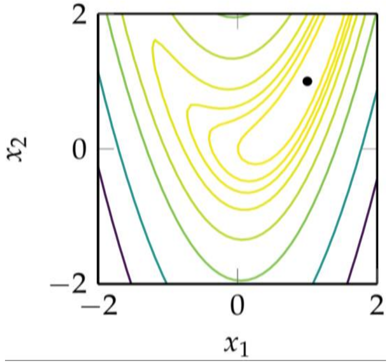
“2 - Derivatives and Gradients”



By:

Dr. Kelvyn Baruc Sánchez Sánchez

Postdoctoral Researcher/I.T. Celaya



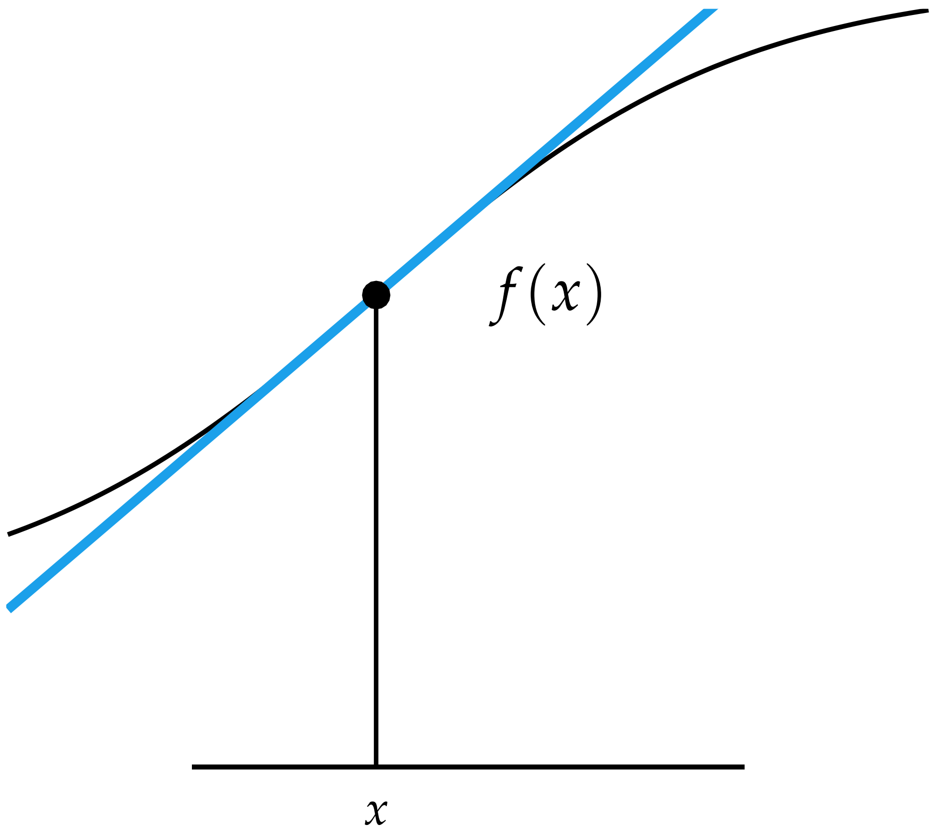
Derivatives

- Derivatives tell us which direction to search for a solution

Derivatives

- Slope of Tangent Line

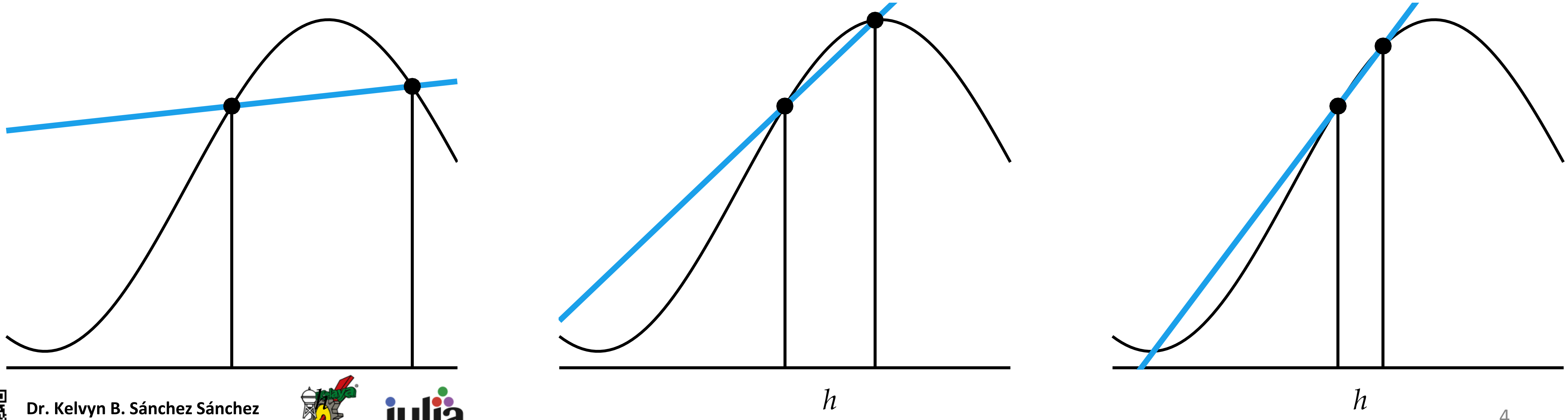
$$f'(x) \equiv \frac{df(x)}{dx}$$



Derivatives

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

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



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:

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

Derivatives

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.

See example 2.1.ipynb

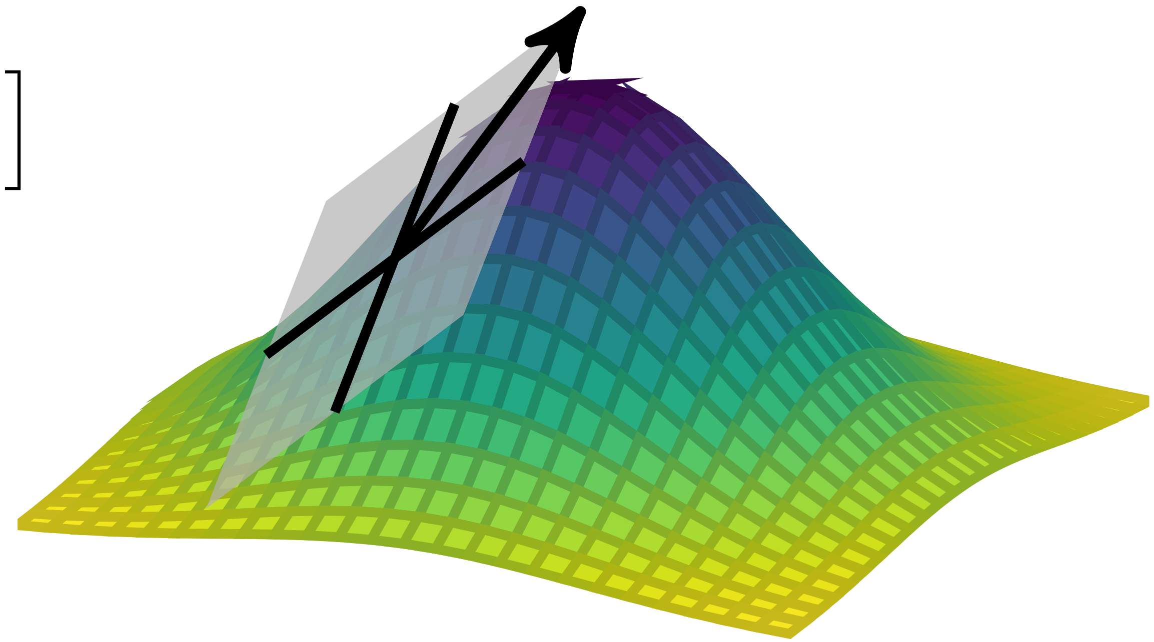
Derivatives in Multiple Dimensions

- Gradient Vector

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

- Hessian Matrix

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



See example 2.2.ipynb

Dr. Kelvyn B. Sánchez Sánchez
Instituto Tecnológico de Celaya

Derivatives in Multiple Dimensions

- *Directional Derivative*

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

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

$$= \underbrace{\lim_{h \to 0} \frac{f(\mathbf{x}) - f(\mathbf{x} - h\mathbf{s})}{h}}_{\text{backward difference}}$$

Derivatives in Multiple Dimensions

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

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

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

We can also compute the directional derivative as follows:

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

See example 2.3.ipynb

Numerical Differentiation

- *Finite Difference Methods*

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

- *Complex Step Method*

Numerical Differentiation: Finite Difference

- *Derivation from Taylor series expansion*

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

Numerical Differentiation: Finite Difference

- *Neighboring points are used to approximate the derivative*

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

- *h too small causes numerical cancellation errors*

Numerical Differentiation: Finite Difference

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

See example 2.4.ipynb

Numerical Differentiation: Complex Step

- *Taylor series expansion using imaginary step*

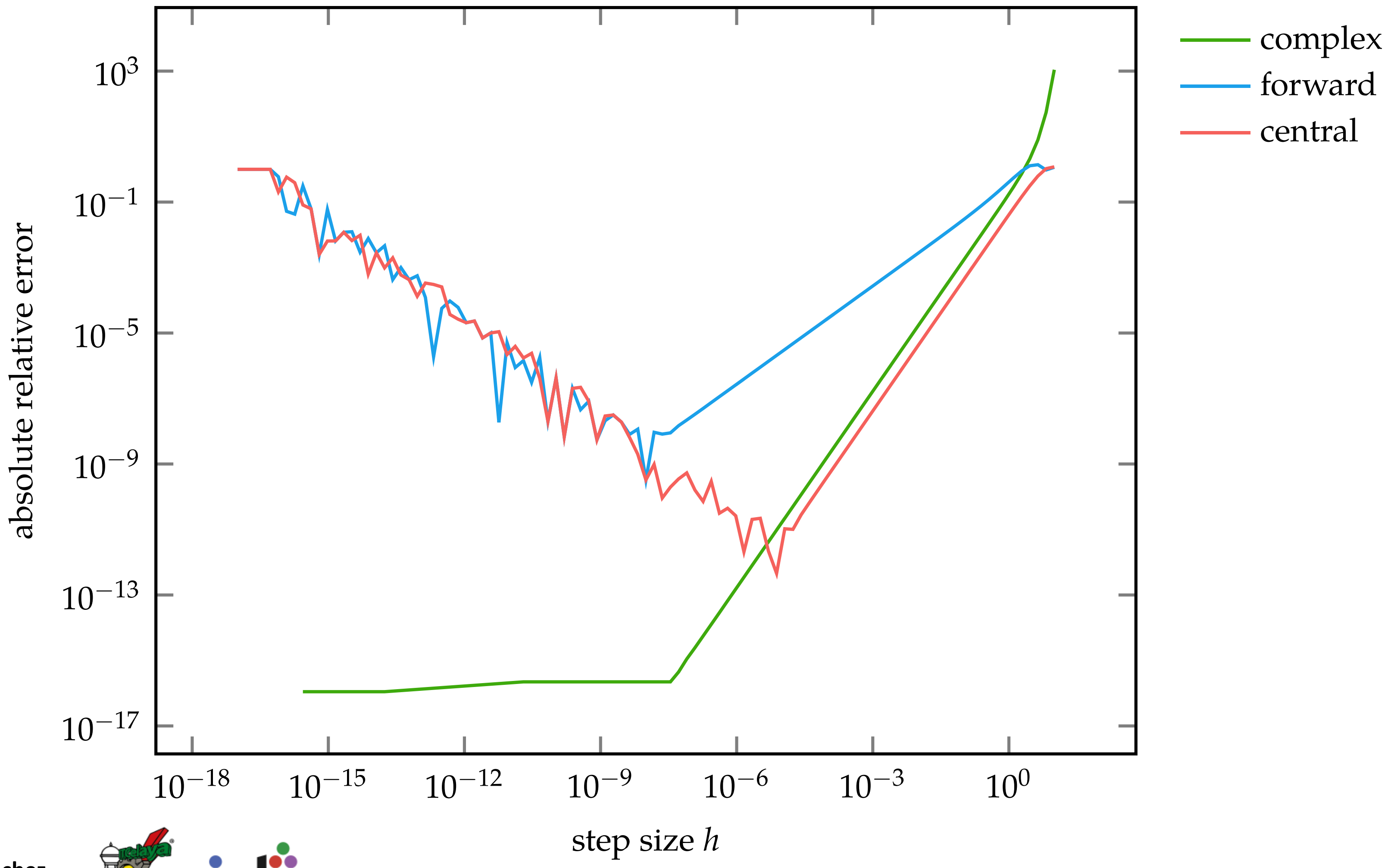
$$f(x+ih) = f(x) + ihf'(x) - h^2\frac{f''(x)}{2!} - ih^3\frac{f'''(x)}{3!} + \cdots$$

$$f'(x) = \frac{\mathrm{Im}(f(x+ih))}{h} + O(h^2) \ \text{ as } h \to 0$$

$$f(x) = \mathrm{Re}(f(x+ih)) + O(h^2)$$

See example 2.5 & 2.6.ipynb

Numerical Differentiation Error Comparison



See example 2.7.ipynb

Dr. Kelvyn B. Sánchez Sánchez
Instituto Tecnológico de Celaya

Automatic Differentiation

- Evaluate a function and compute partial derivatives simultaneously using the chain rule of differentiation

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

Automatic Differentiation

- Forward Accumulation is equivalent to expanding a function using the chain rule and computing the derivatives inside-out
- Requires *n*-passes to compute *n*-dimensional gradient
- Example

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

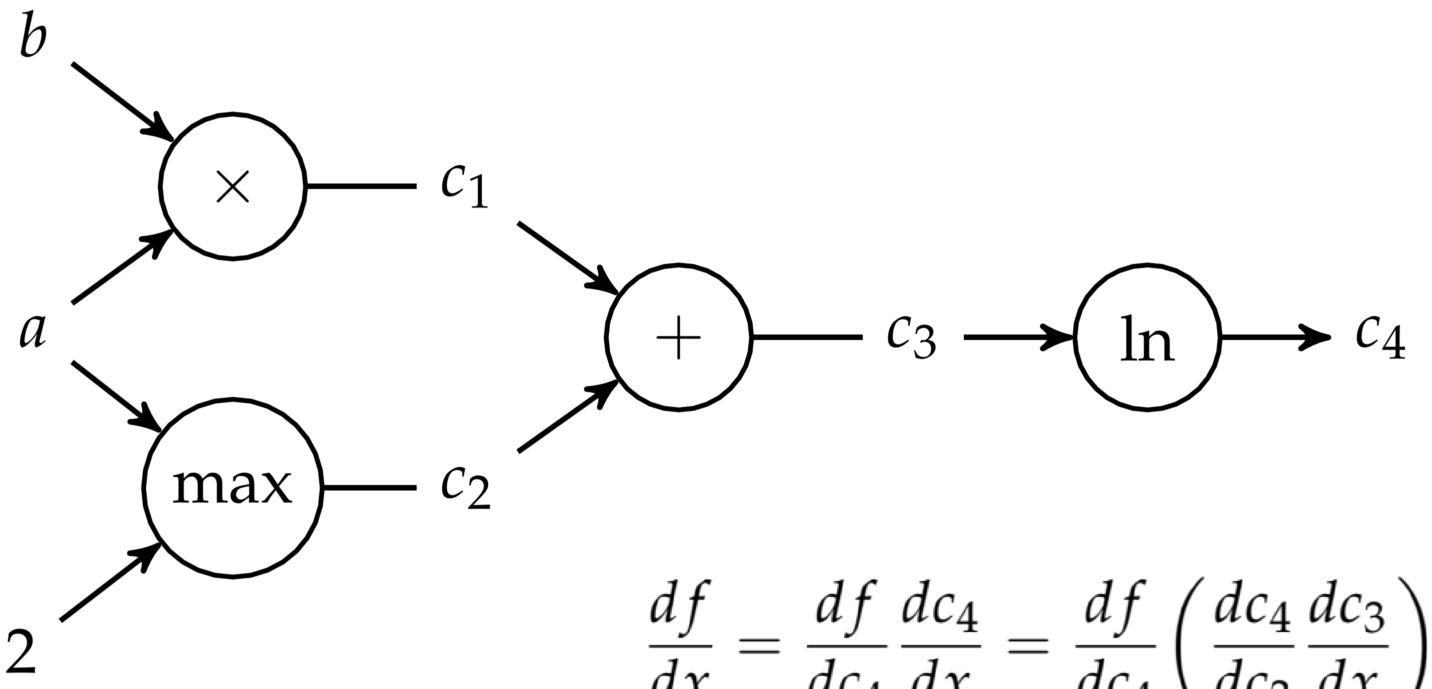
Automatic Differentiation

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

Automatic Differentiation

- Forward Accumulation

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



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

Automatic Differentiation

- Forward Accumulation

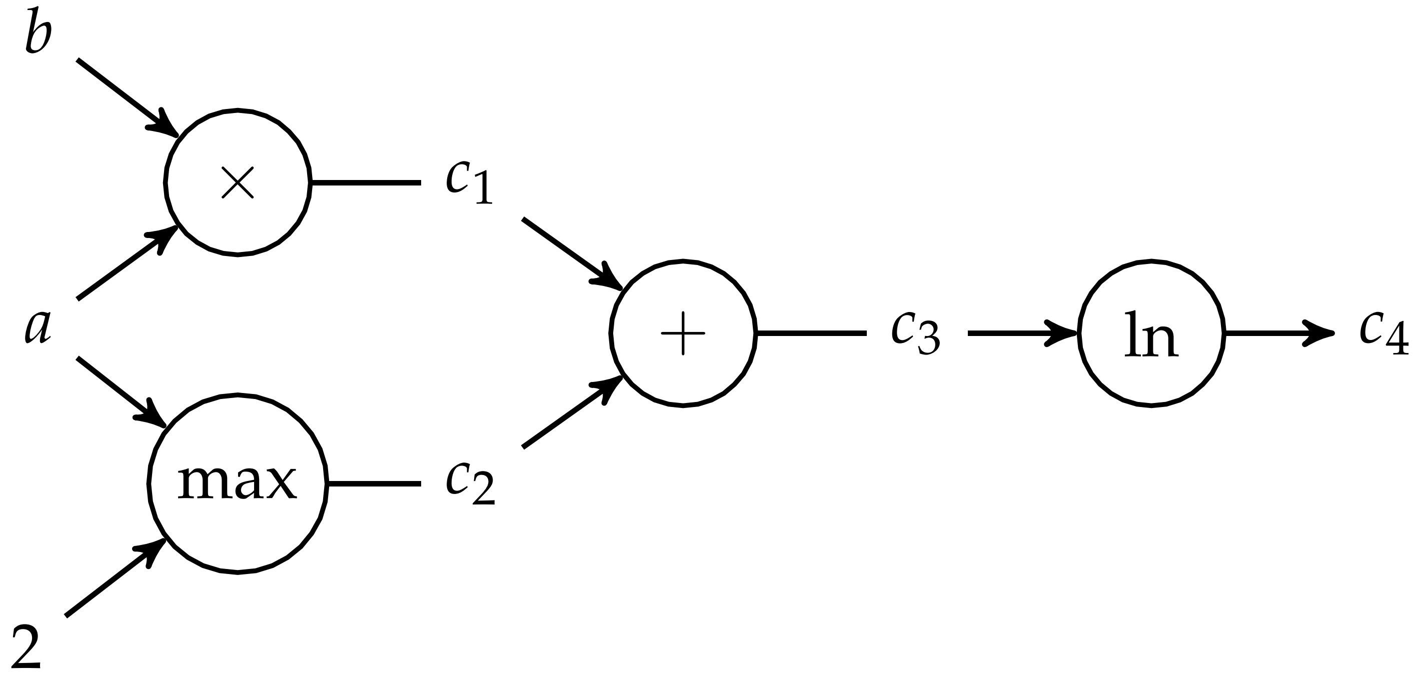
$$f(a,b) = \ln(\, ab + \max(a,2))$$

$b = 2$

$\dot{b} = 0$

$a = 3$

$\dot{a} = 1$



Dr. Kelvyn B. Sánchez Sánchez
Instituto Tecnológico de Celaya

Automatic Differentiation

- Forward Accumulation

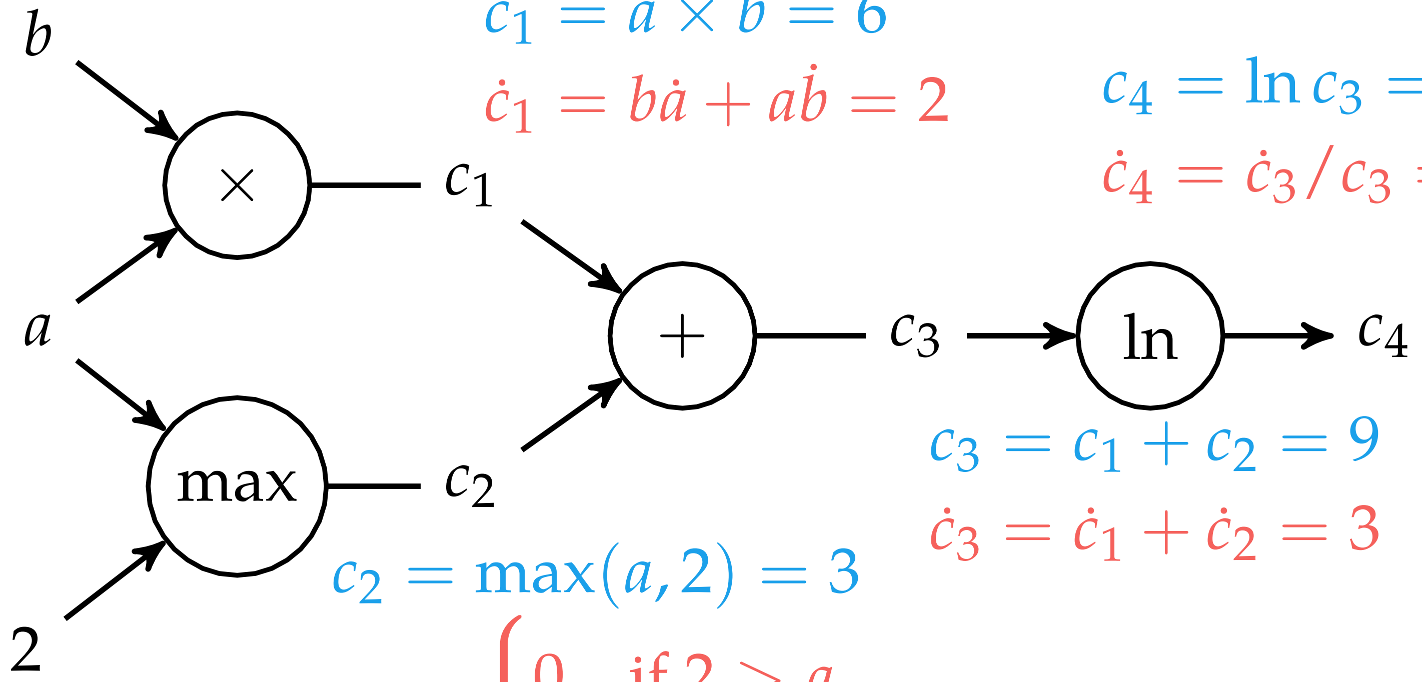
$$f(a,b) = \ln(ab + \max(a,2))$$

$b = 2$

$\dot{b} = 0$

$a = 3$

$\dot{a} = 1$



$c_1 = a \times b = 6$

$\dot{c}_1 = b\dot{a} + a\dot{b} = 2$

$c_2 = \max(a, 2) = 3$

$$\dot{c}_2 = \begin{cases} 0 & \text{if } 2 > a \\ \dot{a} & \text{if } 2 < a \end{cases} = 1$$

$c_3 = c_1 + c_2 = 9$

$\dot{c}_3 = \dot{c}_1 + \dot{c}_2 = 3$

$c_4 = \ln c_3 = \ln 9$

$\dot{c}_4 = \dot{c}_3 / c_3 = \frac{1}{3}$

Automatic Differentiation

- Reverse accumulation is performed in single run using two passes over an *n*-dimensional function (forward and back)
- Note: this is central to the backpropagation algorithm used to train neural networks
- Many open-source software implementations are available

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