Introduction to Algorithmic Differentiation

We Previously Covered Two Methods of Computing Derivatives

- finite-difference approximations, and;
- complex-step approximations.

Algorithmic Differentiation Is Possible If You Have the Source Code

Algorithmic differentiation (AD) systematically differentiates source code line-by-line.

- AD is not symbolic differentiation: we do not get an explicit expression for the function's derivative.
- For a given set of input (design) variables, `x`, AD code produces the derivative of the outputs with respect to those inputs at the given values: we get numbers not an expression.

AD is also known as Automatic Differentiation.

There Are Two “Modes” of Algorithmic Differentiation

forward mode: Also called the tangent mode. Computes directional derivatives of the form $(\nabla f)^T p$.

reverse mode: Also called the adjoint mode. Computes (weighted) gradients (∇f).

I Will Use a Simple Function to Illustrate Both Modes of AD

Consider the following function and its Matlab implementation

$$f(x_1, x_2) = x_1^2 + x_2 \sin\left(x_1^2\right).$$

```
function [f] = func(x1, x2)
% compute a simple function value
v1 = x1.^2;
v2 = x2.*sin(v1);
f = v1 + v2;
end
```