

Automatic Differentiation

A user-friendly, flexible interface for higher order derivatives

CS 207 Final Project

Isabelle Feldhaus, Shenghao Jiang, Robbert Struyven, William Wang

12 December 2018

Motivation

- Addressing the weaknesses of other techniques
 - *Symbolic differentiation*
 - Inefficient code that grows with program size and complexity
 - Challenge of converting computer program into a single expression
 - *Numerical differentiation*
 - Round-off or truncation errors
 - Problems calculating higher derivatives
 - Slow at computing partial derivatives with respect to many inputs
- AD applies the chain rule repeatedly to the sequence of elementary arithmetic operations in a function
 - Computed automatically and accurately

AD Package Features

- Forward mode automatic differentiation
- Reverse mode automatic differentiation
- Flexible function input
 - Mathematical operators
 - Text inputs
- Computation of higher order derivatives

Forward Mode AD Implementation

Reverse Mode AD Implementation

Flexible Function Input using a String Parser

- Interpreter feature allows for text representation of input function

Computing Higher Order Derivatives

Use Cases

Methods

- Real-parameter optimization
- Measuring the sensitivity of simulation to unknown parameters
- Physical modeling, parameter tuning
- Probabilistic inference
- Machine learning

Topics

- Atmospheric chemistry
- Breast cancer biostatistics analysis
- Chemical kinetics
- Semiconductor device simulation
- Water reservoir simulation
- Design optimization

Thank you!