Add Support for differentiating functor objects in Clad

Parth Arora
Mentors: Vassil Vassilev, David Lange

What is Clad?

- Clad is an automatic differentiation clang plugin for C++. It can differentiate mathematical functions represented as C++ functions.

- For each function that is to be differentiated, Clad creates another function that computes its derivative.

Project overview

- Add support for differentiating functor objects and lambda expressions in both forward and reverse mode automatic differentiation.
- Extend Clad by adding support for differentiating more C++ syntax and constructs.
- Make Clad more robust by adding an automatic testing framework for testing reverse mode AD using forward mode AD.

Add Support for differentiating functor objects and lambda expressions

- Differentiating functors means differentiating the call operator (*operator()*) member function defined by the functor type and executing the differentiated function using a reference to the functor.
- All Clad differentiation functions support differentiating functor objects and lambda expressions.
- Functor objects of template class types are also supported.

```
// An object of class type that has defined call operator ('operator()')
class Experiment;

Experiment E;

// both ways are equivalent
auto d_E = clad::differentiate(&E, "i");    // passing functor by pointer
auto d_ERef = clad::differentiate(E, "i");  // passing functor by reference
```

Extend clad by adding support for differentiating more C++ syntax and constructs.

- Added support for differentiating *while* and *do-while* statements in the forward and reverse mode automatic differentiation.

- Added support for differentiating *switch* statement in the forward mode automatic differentiation

Automatic testing of the reverse mode AD using the forward mode AD

- Optionally enabled, using a compile-time flag, assert-based testing of the reverse mode AD using the forward mode AD.

- Verifies at runtime if the produced derivative result is consistent using both forward and reverse mode AD. If verification fails, then aborts the program with 'Assertion Failed' message.

Thank you