# Add Initial Integration of Clad with Enzyme

Manish Kausik H, GSoC 2022

Mentors: Vassil Vassilev, David Lange, William Moses

# Contents

# Clad and Enzyme

- Both are libraries to perform Automatic Differentiation
- Clad:
  - A plugin to the Clang compiler
  - Specific to C++ Language
  - Works on the frontend - Modifies the AST
  - Has: Forward mode, Reverse Mode, Hessian, Jacobian, Error Estimation, Numerical Diff
  - Has support for Object Oriented Constructs
- Enzyme:
  - Works on the Backend - AD on the LLVM IR
  - Applicable to multiple languages
  - Has: Forward mode, Reverse Mode
  - Focus on Interoperability

# Clad API

```
#include "clad/Differentiator/Differentiator.h"
#include <iostream>

double foo(double x) { return x * x; }

int main() {
  // Call clad to generate the derivative of foo wrt x.
  auto foo_dx = clad::differentiate(foo, "x");

  // Call clad to generate the gradient of foo
  auto foo_grad = clad::gradient(foo);
}
```

# Enzyme API

```
#include <iostream>
extern double __enzyme_autodiff(void*, double);
double foo(double x) { return x * x; }
double dfoo(double x) {
    // This returns the derivative of square or 2 * x
    return __enzyme_autodiff((void*) foo, x);
}

int main() {
    for(double i=1; i<5; i++){
        printf("foo(%f)=%f, dfoo(%f)=%f",i,foo(i),i,dfoo(i));
     }
}
```

# Integrating Enzyme Reverse Mode with Clad

1. Identifying a request for using Enzyme with Clad (PR #460)

2. Integrating Enzyme as a static library in Clad (PR #466)

3. Generating code for Enzyme Reverse mode with clad (PR #486)

4. Verifying Enzyme generated derivatives with clad(PR #488)

# Identifying a request for using Enzyme with Clad (PR #460)

```
clad::gradient(f) //Normal Calling convention
clad::gradient<clad::opts::use_enzyme>(f) //Calling Convention for using Enzyme within Clad
```

This was done by:

1. Making clad::opts as an enum with the entry use_enzyme
2. Introducing a new integer type template argument that captures this option
3. Reflect this in DiffRequest when we visit the clad::gradient call expression

## Integrating Enzyme as a static library in Clad (PR #466)

1. Added a CMake flag “-DENABLE_ENZYME_BACKEND”.

2. Had to schedule the Enzyme passes in the right order so that results are correct

3. Tested the integrated enzyme module on some basic functions, just to check if its correctly integrated

Generating code for Enzyme Reverse mode with clad (PR #486)

# Verifying Enzyme Results with Clad (PR #486)

# Benchmarking: Enzyme vs Clad

```
3: ----------------------------------------------------------------------------------------------
3: Benchmark                                                  Time             CPU   Iterations
3: ----------------------------------------------------------------------------------------------
3: BM_ReverseModeAddArrayAndMultiplyWithScalarsExecute      163 ns          163 ns      4175844
3: BM_ReverseModeAddArrayAndMultiplyWithScalarsExecuteEnzyme 18.7 ns         18.7 ns     37734184
3: BM_ReverseModeSumExecute                                 51.3 ns         51.3 ns     13539251
3: BM_ReverseModeSumExecuteWithEnzyme                       13.0 ns         13.0 ns     50688863
3: BM_ReverseModeProductExecute                              102 ns          102 ns      6913341
3: BM_ReverseModeProductExecuteEnzyme                       27.5 ns         27.5 ns     25498045
3: BM_ReverseGaus                                            209 ns          209 ns      3402252
3: BM_ReverseGausEnzyme                                     83.1 ns         83.1 ns      8296798
3/4 Test #3: clad-EnzymeCladComparison ........   Passed    6.92 sec
```

Thank You!