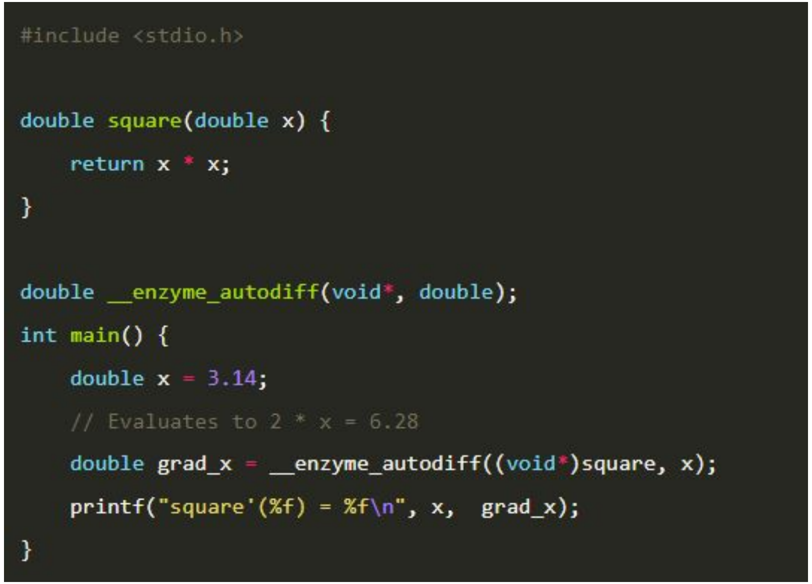
# A Deep Dive into Enzyme

Zack McKevitt

# Enzyme

- Automatic Differentiation Framework
- Compiler that takes existing code and computes derivative of said function
- Built on LLVM
- Large open source project with 16 contributors and ~700 stars
- Pre publication paper: https://arxiv.org/pdf/2010.01709.pdf



```
#include <stdio.h>

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

double __enzyme_autodiff(void*, double);
int main() {
    double x = 3.14;
    // Evaluates to 2 * x = 6.28
    double grad_x = __enzyme_autodiff((void*)square, x);
    printf("square'(%f) = %f\n", x,  grad_x);
}
```

# Zygote

- Zygote is an extension to the Julia programming language to support automatic differentiation
- Much older, repository dates back to 2011
- Large open source repository (80+ contributors and 1.2k+ stars)
- Exclusive to the julia programming language
- Paper: https://arxiv.org/abs/1810.07951

# Initial Project Direction

- Compare Enzyme performance and implementation to the older Zygote
- Analyze compiler techniques used by Enzyme to perform automatic differentiation at the systems level
- Analyze strengths of Enzyme over Zygote and why certain optimizations work better/worse
- Open to any suggestions!

# Motivation

- Compilers are fun!
- Learn LLVM API and how to use it for numerical analysis
- Learn a little bit of Rust (some of Enzyme repo is in Rust)
- Collect empirical data to determine which tool is better in different circumstances

# Repositories

Enzyme: https://github.com/EnzymeAD/Enzyme

Zygote: https://github.com/FluxML/Zygote.jl