



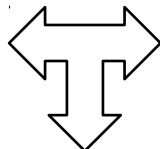
# Tool Development for Supporting Algorithmic Differentiation

## 1. Motivation

In many scientific applications derivatives are needed for optimizations or data assimilation to match physical observations and model tuning. In this context, algorithmic differentiation (AD) allows for the semantic augmentation of codes yielding the ability to compute derivatives. Typically in C++, the double data type is replaced by a user-defined AD type which overloads all operators and mathematical functions. The code structure, thus, stays mostly unchanged.

The new type, in addition to the normal function value, also calculates the derivative value:

```
1 double f(double x) {
2   double y = x * x + x;
3   return y;
4 }
```



```
1 #include "adouble.h"
2 adouble f(adouble x) {
3   adouble y = x * x + x;
4   return y;
5 }
```

```
1 class adouble {
2   double value;
3   double derivative;
4 public:
5   adouble(double);
6   adouble(const adouble&);
7   adouble operator*(const adouble&);
8   adouble operator+(const adouble&);
9 };
```

While this looks straightforward, the semantic augmentation can lead to many compiler errors and often requires substantial effort to integrate into existing code bases.

## Contact

Alexander Hück, M. Sc.  
S1 | 22 Room 412  
Tel. +49 6151 16-7 55 77  
[alexander.hueck@sc.tu-darmstadt.de](mailto:alexander.hueck@sc.tu-darmstadt.de)



Date  
06/08/2017

## 2. Research Questions

- There are different techniques to compute derivatives with AD: How can they be combined to efficiently compute derivatives? (Runtime analysis, performance modelling)
- Supporting the semantic augmentation through automatic code changes to avoid compiler errors. (Source code analysis and transformations)
- There are many different AD implementations using operator overloading with different performance characteristics:
  1. Is a source translator tool to port between the different implementations feasible?
  2. Is it possible to write a template-based wrapper library for a AD tool abstraction layer?

Furthermore, if interested, other programming languages, e.g. Rust, can be explored.

