

The Manual of ReverseAD

Mu Wang

September 19, 2016

Abstract

(TODO)

Contents

Preliminaries of Automatic Differentiation

(TODO)

Chapter 1

Basic Usage of ReverseAD

1.1 Download And Installation

1.2 The One Minute Example

```
1 #include <memory>
2 #include <iostream>
3
4 #include "reversead/reversead.hpp"
5
6 using ReverseAD::adouble;
7 using ReverseAD::TrivialTrace;
8 using ReverseAD::BaseReverseHessian;
9 using ReverseAD::DerivativeTensor;
10 using ReverseAD::trace_on;
11 using ReverseAD::trace_off;
12
13 template <typename T>
14 T foo(T x1, T x2) {
15     return pow(x1+1, 2) + pow(x1*x1 - x2, 2);
16 }
17
18 int main() {
19     adouble x1, x2;
20     adouble y;
```

```

21 double vy;
22 trace_on<double>(); // begin tracing
23 x1 <=<= 2.0; // independent variable #0
24 x2 <=<= 3.0; // independent variable #1
25 y = foo<adouble>(x1, x2); // function evaluation
26 y >=>= vy; // dependent variable
27 std::shared_ptr<TrivialTrace<double>> trace =
28     trace_off<double>(); // end tracing
29 std::cout << "y = " << vy << std::endl;
30
31 std::unique_ptr<BaseReverseHessian<double>> hessian(
32     new BaseReverseHessian<double>(trace));
33 std::shared_ptr<DerivativeTensor<size_t, double>> tensor = hessian
34     ->compute(2,1);
35
36 // retrieve results
37 size_t size;
38 size_t** tind;
39 double* values;
40 // adjoints : dep[0].order[1]
41 tensor->get_internal_coordinate_list(0, 1, &size, &tind, &values);
42 std::cout << "size of adjoints = " << size << std::endl;
43 for (size_t i = 0; i < size; i++) {
44     std::cout << "A[" << tind[i][0] << "] = " << values[i] << std::
45     endl;
46 }
47 // hessian : dep[0].order[2]
48 tensor->get_internal_coordinate_list(0, 2, &size, &tind, &values);
49 std::cout << "size of hessian = " << size << std::endl;
50 for (size_t i = 0; i < size; i++) {
51     std::cout << "H[" << tind[i][0] << ", " << tind[i][1]
52     << "] = " << values[i] << std::endl;
53 }
54 }

```

Listing 1.1 One Minute Example

1.3 Additional Options and Examples

1.3.1 Disk Tape

1.3.2 Forward-over-Reverse Mode

Chapter 2

Internal Implementation

2.1 Function Trace

```
1 private:
2     std::shared_ptr<VirtualTape<opbyte>> op_tape;
3     std::shared_ptr<VirtualTape<locint>> loc_tape;
4     std::shared_ptr<VirtualTape<Base>> val_tape;
5     std::shared_ptr<VirtualTape<Base>> param_tape;
6     std::shared_ptr<VirtualTape<double>> coval_tape;
```


2.2 Active Type

2.3 Indexing Scheme

2.4 Derivative Computation Classes

2.4.1 Adjoint, Hessian and Third Order

2.4.2 Generic Reverse Mode

2.4.3 Generating Flat Code

2.5 Forward-over-Reverse Mode

2.6 Time-Step Functions