Algorithmic Differentiation V

Second-Order Adjoints of Multivariate Scalar Functions

Uwe Naumann

Informatik 12:
Software and Tools for Computational Engineering (STCE)

RWTH Aachen University

Contents

Objective and Learning Outcomes

Tangents of Adjoints
- Derivation
- Implementation

Adjoints of Tangents
- Derivation
- Implementation

Adjoints of Adjoints
- Derivation
- Implementation

Summary and Next Steps

Outline

Objective and Learning Outcomes

Tangents of Adjoints
Derivation
Implementation

Adjoints of Tangents
Derivation
Implementation

Adjoints of Adjoints
Derivation
Implementation

Summary and Next Steps

Objective and Learning Outcomes

Objective

- Introduction to second-order adjoints of multivariate scalar functions and implementation with dco/c++

Learning Outcomes

- You will understand
  - second-order adjoints in tangent-of-adjoint, adjoint-of-tangent and adjoint-of-adjoint modes.
- You will be able to
  - implement second-order adjoints with dco/c++.

Outline

Objective and Learning Outcomes

Tangents of Adjoints
- Derivation
- Implementation

Adjoints of Tangents
- Derivation
- Implementation

Adjoints of Adjoints
- Derivation
- Implementation

Summary and Next Steps

A second derivative code

$$f_{(1)}^{(2)} : \boldsymbol{R}^n \times \boldsymbol{R}^n \times \boldsymbol{R} \times \boldsymbol{R} \rightarrow \boldsymbol{R} \times \boldsymbol{R} \times \boldsymbol{R}^{1\times n} \times \boldsymbol{R}^{1\times n},$$

generated by algorithmic differentiation in tangent-of-adjoint mode computes

$$\begin{pmatrix} y \\ y^{(2)} \\ \mathrm{x}_{(1)} \\ \mathrm{x}_{(1)}^{(2)} \end{pmatrix} = f_{(1)}^{(2)}\left(\mathrm{x}, \mathrm{x}^{(2)}, y_{(1)}, y_{(1)}^{(2)}\right) = \begin{pmatrix} f(\mathrm{x}) \\ f'(\mathrm{x}) \cdot \mathrm{x}^{(2)} \\ y_{(1)} \cdot f'(\mathrm{x}) \\ \mathrm{x}^{(2)\,T} \cdot y_{(1)} \cdot f''(\mathrm{x}) + y_{(1)}^{(2)} \cdot f'(\mathrm{x}) \end{pmatrix}.$$

Finite differences applied to adjoints yield approximate second-order adjoints.

The computational cost of accumulating the Hessian in either finite difference-of-adjoint or tangent-of-adjoint modes is $O(n) \cdot \text{Cost}(f)$.

Algorithmic differentiation of the first-order adjoint

$$\begin{pmatrix} y \\ x_{(1)}^T \end{pmatrix} = \begin{pmatrix} f(x) \\ f'(x)^T \cdot y_{(1)} \end{pmatrix}$$

in tangent mode (differentiation of $f'(x)^T : \boldsymbol{R}^n \to \boldsymbol{R}^n$) yields

$$\begin{pmatrix} y^{(2)} \\ x_{(1)}^{(2)\,T} \end{pmatrix} \equiv \frac{d\begin{pmatrix} y \\ x_{(1)}^T \end{pmatrix}}{d\begin{pmatrix} x \\ y_{(1)} \end{pmatrix}} \cdot \begin{pmatrix} x^{(2)} \\ y_{(1)}^{(2)} \end{pmatrix} = \begin{pmatrix} \frac{df(x)}{dx} \cdot x^{(2)} \left[+ \frac{df(x)}{dy_{(1)}} \cdot y_{(1)}^{(2)} = 0 \right] \\ \frac{d\left(f'(x)^T \cdot y_{(1)}\right)}{dx} \cdot x^{(2)} + \frac{d\left(f'(x)^T \cdot y_{(1)}\right)}{dy_{(1)}} \cdot y_{(1)}^{(2)} \end{pmatrix}$$

$$= \begin{pmatrix} f'(x) \cdot x^{(2)} \\ f''(x) \cdot y_{(1)} \cdot x^{(2)} + f'(x)^T \cdot y_{(1)}^{(2)} \end{pmatrix}$$

implying $$\begin{pmatrix} y^{(2)} \\ x_{(1)}^{(2)} \end{pmatrix} = \begin{pmatrix} f'(x) \cdot x^{(2)} \\ x^{(2)\,T} \cdot y_{(1)} \cdot f''(x) + y_{(1)}^{(2)} \cdot f'(x) \end{pmatrix}.$$

Tangents of Adjoints

Implementation

```
#include "dco.hpp"
#include "Eigen/Dense"

template<typename T, int N>
void f_a(const Eigen::Matrix<T,N,1>& x_v, T& y_v, Eigen::Matrix<T,N,1>& dydx);

template<typename T, int N>
void f_a_t(const Eigen::Matrix<T,N,1>& x_v, T& y_v, Eigen::Matrix<T,N,1>& dydx_v,
    Eigen::Matrix<T,N,N>& ddydxx) {
  using DCO_T=typename dco::gt1s<T>::type;
  auto n=x_v.size();
  Eigen::Matrix<DCO_T,N,1> x(n), dydx(n); DCO_T y=0;
  for (auto i=0;i<n;i++) x(i)=x_v(i);
  for (auto i=0;i<n;i++) {
    dco::derivative(x(i))=1;
    f_a(x,y,dydx);
    for (auto j=0;j<n;j++) ddydxx(j,i)=dco::derivative(dydx(j));
    dco::derivative(x(i))=0;
  }
  for (auto j=0;j<n;j++) dydx_v(j)=dco::value(dydx(j));
  y_v=dco::value(y);
}
```

```
#include "dco.hpp"
#include "Eigen/Dense"
#include <limits>

template<typename T, int N>
void f_a(const Eigen::Matrix<T,N,1>& x_v, T& y_v, Eigen::Matrix<T,N,1>& dydx);

template<typename T, int N>
void f_a_cfd(Eigen::Matrix<T,N,1>& x, T& y, Eigen::Matrix<T,N,1>& dydx, Eigen::Matrix
  <T,N,N>& ddydxx) {
  auto n=x.size();
  for (auto i=0;i<n;i++) {
    T dx=fabs(x(i))<1 ? sqrt(std::numeric_limits<T>::epsilon())
       : sqrt(std::numeric_limits<T>::epsilon())*fabs(x(i));
    T yd;
    Eigen::Matrix<T,N,1> dydxp(n), dydxm(n);
    x(i)+=dx; f_a(x,yd,dydxp); x(i)-=2*dx; f_a(x,yd,dydxm); x(i)+=dx;
    for (auto j=0;j<n;j++) ddydxx(j,i)=(dydxp(j)-dydxm(j))/(2*dx);
  }
  f_a(x,y,dydx);
}
```

Outline

Objective and Learning Outcomes

Tangents of Adjoints
- Derivation
- Implementation

Adjoints of Tangents
- Derivation
- Implementation

Adjoints of Adjoints
- Derivation
- Implementation

Summary and Next Steps

A second derivative code

$$f_{(2)}^{(1)} : \mathbf{R}^n \times \mathbf{R}^n \times \mathbf{R} \times \mathbf{R} \to \mathbf{R} \times \mathbf{R} \times \mathbf{R}^{1\times n} \times \mathbf{R}^{1\times n}$$

generated by algorithmic differentiation in adjoint-of-tangent mode computes

$$\begin{pmatrix} y \\ y^{(1)} \\ \mathrm{x}_{(2)} \\ \mathrm{x}_{(2)}^{(1)} \end{pmatrix} = f_{(2)}^{(1)}\left(\mathrm{x}, \mathrm{x}^{(1)}, y_{(2)}, y_{(2)}^{(1)}\right) = \begin{pmatrix} f(\mathrm{x}) \\ f'(\mathrm{x}) \cdot \mathrm{x}^{(1)} \\ y_{(2)}^{(1)} \cdot \mathrm{x}^{(1)\,T} \cdot f''(\mathrm{x}) + y_{(2)} \cdot f'(\mathrm{x}) \\ y_{(2)}^{(1)} \cdot f'(\mathrm{x}) \end{pmatrix} .$$

An adjoint of a finite difference approximation of the first-order tangent yields an approximate second-order adjoint.

The computational cost of accumulating the Hessian in either adjoint-of-finite-difference or adjoint-of-tangent modes is $O(n) \cdot \mathrm{Cost}(f)$ ($O(n^2) \cdot \mathrm{Cost}(f)$ if implemented naively).

Algorithmic differentiation of the first-order tangent

$$\begin{pmatrix} y \\ y^{(1)} \end{pmatrix} = \begin{pmatrix} f(x) \\ f'(x) \cdot x^{(1)} \end{pmatrix}$$

in adjoint mode yields

$$\begin{pmatrix} x_{(2)} \\ x^{(1)}_{(2)} \end{pmatrix} \equiv \begin{pmatrix} y_{(2)} & y^{(1)}_{(2)} \end{pmatrix} \cdot \frac{d\begin{pmatrix} y \\ y^{(1)} \end{pmatrix}}{d\begin{pmatrix} x \\ x^{(1)} \end{pmatrix}} = \begin{pmatrix} y_{(2)} \cdot \frac{df(x)}{dx} + y^{(1)}_{(2)} \cdot \frac{d\left(f'(x)\cdot x^{(1)}\right)}{dx} \\ \left[y_{(2)} \cdot \frac{df(x)}{dx^{(1)}} = 0 + \right] y^{(1)}_{(2)} \cdot \frac{d\left(f'(x)\cdot x^{(1)}\right)}{dx^{(1)}} \end{pmatrix}$$

implying with $f'(x) \cdot x^{(1)} = {x^{(1)}}^T \cdot f'(x)^T$ (differentiation of $f'(x)^T$)

$$\begin{pmatrix} x_{(2)} \\ x^{(1)}_{(2)} \end{pmatrix} = \begin{pmatrix} y_{(2)} \cdot f'(x) + y^{(1)}_{(2)} \cdot {x^{(1)}}^T \cdot f''(x) \\ y^{(1)}_{(2)} \cdot f'(x) \end{pmatrix} .$$

While adjoints of tangents can be implemented with dco/c++ the mathematically equivalent[1] tangent-of-adjoint mode of algorithmic differentiation is typically preferred.

Implementation with dco/c++ exploits the seamless nesting of derivative types as dco::gt1s<dco::ga1s<**double**>::type>::type.

[1] ${x^{(2)}}^T \cdot y_{(1)} \cdot f''(x) = y_{(2)}^{(1)} \cdot {x^{(1)}}^T \cdot f''(x)$ for $y_{(1)} = y_{(2)}^{(1)}$ and $x^{(2)} = x^{(1)}$.

Outline

Objective and Learning Outcomes

Tangents of Adjoints
- Derivation
- Implementation

Adjoints of Tangents
- Derivation
- Implementation

Adjoints of Adjoints
- Derivation
- Implementation

Summary and Next Steps

A second derivative code

$$f_{(1,2)} : \boldsymbol{R}^n \times \boldsymbol{R}^n \times \boldsymbol{R} \times \boldsymbol{R} \rightarrow \boldsymbol{R} \times \boldsymbol{R}^{1\times n} \times \boldsymbol{R}^{1\times n} \times \boldsymbol{R},$$

generated by algorithmic differentiation in adjoint-of-adjoint mode computes

$$\begin{pmatrix} y \\ \mathbf{x}_{(1)} \\ \mathbf{x}_{(2)} \\ y_{(1,2)} \end{pmatrix} = f_{(1,2)}\left(\mathbf{x}, \mathbf{x}_{(1,2)}, y_{(1)}, y_{(1,2)}\right) = \begin{pmatrix} f(\mathbf{x}) \\ y_{(1)} \cdot f'(\mathbf{x}) \\ y_{(2)} \cdot f'(\mathbf{x}) + \mathbf{x}_{(1,2)}^T \cdot y_{(1)} \cdot f''(\mathbf{x}) \\ f'(\mathbf{x}) \cdot \mathbf{x}_{(1,2)} \end{pmatrix}$$

The computational cost of accumulating the Hessian in adjoint-of-adjoint mode is $O(n) \cdot \text{Cost}(f)$.

Algorithmic differentiation of the first-order adjoint

$$\begin{pmatrix} y \\ x_{(1)}^T \end{pmatrix} = \begin{pmatrix} f(x) \\ f'(x)^T \cdot y_{(1)} \end{pmatrix}$$

in adjoint mode (differentiation of $x_{(1)}^T \in \boldsymbol{R}^n$ instead of $x_{(1)} \in \boldsymbol{R}^{1 \times n}$) yields

$$\begin{pmatrix} x_{(2)} \\ y_{(1,2)} \end{pmatrix} \equiv \begin{pmatrix} y_{(2)} & x_{(1,2)}^T \end{pmatrix} \cdot \frac{d \begin{pmatrix} y \\ x_{(1)}^T \end{pmatrix}}{d \begin{pmatrix} x \\ y_{(1)} \end{pmatrix}} = \begin{pmatrix} y_{(2)} \cdot \frac{df(x)}{dx} + x_{(1,2)}^T \cdot \frac{d\left(f'(x)^T \cdot y_{(1)}\right)}{dx} \\ \left[y_{(2)} \cdot \frac{df(x)}{dy_{(1)}} = 0 \right] + x_{(1,2)}^T \cdot \frac{d\left(f'(x)^T \cdot y_{(1)}\right)}{dy_{(1)}} \end{pmatrix}$$

implying with $f'(x)^T \cdot y_{(1)} = y_{(1)} \cdot f'(x)^T$ and $x_{(1,2)}^T \cdot f'(x)^T = f'(x) \cdot x_{(1,2)}$

$$\begin{pmatrix} x_{(2)} \\ y_{(1,2)} \end{pmatrix} = \begin{pmatrix} y_{(2)} \cdot f'(x) + x_{(1,2)}^T \cdot y_{(1)} \cdot f''(x) \\ f'(x) \cdot x_{(1,2)} \end{pmatrix} .$$

While adjoints of adjoints can be implemented with dco/c++ the mathematically equivalent[2] tangent-of-adjoint mode of algorithmic differentiation is typically preferred.

Implementation with dco/c++ exploits the seamless nesting of derivative types as dco::ga1s<dco::ga1s<**double**>::type>::type.

[2] ${x^{(2)}}^T \cdot y_{(1)} \cdot f''(x) = {x_{(1,2)}}^T \cdot y_{(1)} \cdot f''(x)$ for $x^{(2)} = x_{(1,2)}$.

Outline

Software and Tools for Computational Engineering | RWTH AACHEN UNIVERSITY

Objective and Learning Outcomes

Tangents of Adjoints
- Derivation
- Implementation

Adjoints of Tangents
- Derivation
- Implementation

Adjoints of Adjoints
- Derivation
- Implementation

Summary and Next Steps

Summary

- Introduction to second-order adjoints of multivariate scalar functions in tangent-of-adjoint, adjoint-of-tangent and adjoint-of-adjoint modes and implementation with dco/c++

Next Steps

- Download and inspect sample code.
- Run you own experiments.
- Continue the course to find out more ...