Adjoint program generated by automatic differentiation of a meteorological simulation program, and its application to gradient computation

Yasuyoshi Horibata
Faculty of Science and Engineering, Hosei University, 3-7-2 kajino-cho, Koganei, Tokyo, Japan
horibata@hosei.ac.jp

Abstract
A source computer program that simulates the atmospheric flow is differentiated by Automatic Differentiation tool TAPENADE. Numerical experiments are presented for the adjoint program.

Keywords: Adjoint program, Automatic differentiation, Atmospheric flow simulation, Gradient

Introduction
Automatic Differentiation (AD) transforms a source computer program P that computes a mathematical vector function into a new source program that computes derivatives of this function. In this paper, a source computer program that simulates the atmospheric flow is differentiated by the AD tool TAPENADE [Hascoet and Pascual (2004)]. The generated adjoint program is employed to compute the gradient of a cost function.

Automatic Differentiation
Given a vector argument \( \mathbf{X} \in \mathbb{R}^n \), a source computer program P computes some vector function \( \mathbf{Y} = \mathbf{F}(\mathbf{X}) \in \mathbb{R}^m \). The AD tool generates a new source program that, given the argument \( \mathbf{X} \), computes some derivatives of \( \mathbf{F} \). P represents a sequence of instructions, which is identified with a composition of vector functions. Thus

\[
P = \{I_1; I_2; \ldots; I_p\}
\]

\[
\mathbf{F} = f_p \circ f_{p-1} \circ \ldots \circ f_1
\]

Here each \( f_i \) is the elementary function implemented by instruction \( I_i \).

The chain rule gives the Jacobian \( \mathbf{F}' \) of \( \mathbf{F} \). Using the Jacobian, for a small perturbation \( \delta \mathbf{X} \) in \( \mathbf{X} \), the corresponding perturbation \( \delta \mathbf{Y} = \mathbf{F}(\mathbf{X} + \delta \mathbf{X}) - \mathbf{F}(\mathbf{X}) \) in \( \mathbf{Y} \) is computed with quadratic errors:

\[
\delta \mathbf{Y} = \mathbf{F}'(\mathbf{X}) \times \delta \mathbf{X} = f'_p(\mathbf{X}_{p-1}) \times f'_{p-1}(\mathbf{X}_{p-2}) \times \ldots \times f'_1(\mathbf{X}_0) \times \delta \mathbf{X}
\]  \( (1) \)

The tangent program computes this perturbation; the program is generated by differentiating the source program in tangent mode. On the other hand, a scalar linear combination \( \mathbf{Y}^T \times \mathbf{\hat{Y}} \) is defined as the new result of the source program; \( \mathbf{\hat{Y}} \) is the weighting vector. The gradient of \( \mathbf{Y}^T \times \mathbf{\hat{Y}} \) is

\[
\mathbf{F}'^T(\mathbf{X}) \times \mathbf{\hat{Y}} = f'^T_p(\mathbf{X}_0) \times f'^T_{p-1}(\mathbf{X}_1) \times \ldots \times f'^T_{p-2}(\mathbf{X}_{p-2}) \times f'^T_p(\mathbf{X}_{p-1}) \times \mathbf{\hat{Y}}
\]  \( (2) \)

The adjoint program is generated by differentiating the source program in reverse mode, and computes the gradient.
Numerical Experiments

Numerical experiments are presented for a three-dimensional downburst. A downburst is a strong downdraft which induces an outburst of damaging winds on or near the ground. A downburst is simulated using the finite difference scheme [Horibata (2012)]. Then, the simulation program is differentiated in reverse mode by AD tool TAPENADE. The cost function is defined by

\[
J(X_0) = \sum_{i=0}^{N} c_q (q_{ri} - q_{ri}^{ob})^T (q_{ri} - q_{ri}^{ob}) + \sum_{j=0}^{N} c_u (u_{ji} - u_{ji}^{ob})^T (u_{ji} - u_{ji}^{ob})
\]

Here \(q_r\) and \(u\) are the mixing ratio of rainwater and the x component of the wind velocity, respectively; \(q_{ri}^{ob}\) and \(u_{ji}^{ob}\) are their observations, respectively. \(N\) is the number of time steps. The gradient of the cost function with respect to the field variables is computed by the adjoint program. Figures 1 and 2 compare the gradients with respect to rainwater field and the x-component field of the wind velocity with ones computed by the finite difference method, respectively.

![Figure 1](image1.png)

**Figure 1.** Gradient with respect to the rainwater field computed by the adjoint program (left) and the finite difference method (right) in the x-z cross section at y=0

![Figure 2](image2.png)

**Figure 2.** Gradient with respect to the x component field of the wind velocity computed by the adjoint program (left) and the finite difference method (right) in the x-z cross section at y=0

References

Hascoet, L. and Pascual, V. (2004) TAPENADE 2.1 user's guide, INRIA.