ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

METEOROLOGICAL DATA ASSIMILATION USING AN ADJOINT PROGRAM GENERATED BY AUTOMATIC DIFFERENTIATION

Yasuyoshi Horibata

Hosei University
3-7-2 kajino-cho, Koganei, Tokyo, Japan
e-mail: horibata@hosei.ac.jp

Keywords: Adjoint program, Automatic differentiation, Atmospheric flow simulation, TAPENADE, modified L-BFGS-B.

Abstract. *Meteorological data assimilation is formulated as a large-scale nonlinear optimization problem. In order to solve the problem, modified L-BFGS-B algorithm is used. At each search, the algorithm requires the gradient of a cost function. Automatic differentiation transforms a source computer program that computes a mathematical vector function into a new source program that computes derivatives of this function. A source computer program that simulates the atmospheric flow is differentiated in reverse mode by automatic differentiation tool TAPENADE, and the adjoint program is generated. The gradient is computed by using the adjoint program. Numerical experiments are presented for a three-dimensional downburst.*

1 INTRODUCTION

Numerical simulation of the atmospheric flow is an initial/boundary value problem: given an initial condition of the atmosphere, and appropriate surface and lateral boundary conditions, the model simulates the atmospheric evolution. Assimilation of meteorological observations is the process through which all the available information is used in order to estimate as accurately as possible the state of the atmospheric flow (an initial condition) [1]. The available information consists of the observation proper, and of the physical laws that govern the evolution of flow. In this paper, the assimilation is formulated as a nonlinear optimization problem; a cost function is minimized. At each iteration, an optimization algorithm requires the gradient of the cost function.

We developed a method for gradient computation by perturbing the initial condition for the discretized model equation [2, 3]. In this paper, however, we compute the gradient using an adjoint program.

The adjoint program is obtained by automatic differentiation (AD) tool. Automatic differentiation transforms a source computer program that computes a mathematical vector function into a new source program that computes derivatives of this function. A source computer program that simulates the atmospheric flow is differentiated by the AD tool TAPENADE [4]. The generated adjoint program is employed to compute the gradient of the cost function.

2 AUTOMATIC DIFFERENTIATION

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

$$P \quad is \quad \{I_1; I_2; \ldots I_p\} \tag{1}$$

$$F \quad = \quad f_p \circ f_{p-1} \circ \ldots \circ f_1 \tag{2}$$

Here each f_k is the elementary function implemented by instruction I_k.

The chain rule gives the Jacobian F' of F. Using the Jacobian, for a small perturbation δX in X, the corresponding perturbation $\delta Y = F(X + \delta X) - F(X)$ in Y is computed:

$$\delta Y = F' \delta X = f'_p(X_{p-1}) f'_{p-1}(X_{p-2}) \ldots f'_2(X_1) f'_1(X_0) \delta X \tag{3}$$

Here, X_k are the values of all variables after each instruction I_k; $X_0 = X$ and $X_k = f_k(X_{k-1})$.

The tangent program computes this perturbation; the program is generated by differentiating the source program in tangent mode. On the other hand, we define a scalar linear combination $Y^T W$ as the new result of the source program; W is the weighting vector. The gradient of $Y^T W$ is

$$F'^T(X) W = f_1'^T(X_0) f_2'^T(X_1) \ldots f_{p-1}'^T(X_{p-2}) f_p'^T(X_{p-1}) W \tag{4}$$

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

3 SOURCE PROGRAM

The model equation for the atmospheric flow [5] is written as

$$\frac{\partial \mathbf{x}}{\partial t} \quad = \quad \mathbf{F}(\mathbf{x}, \mathbf{y}) + \mathbf{D}(\mathbf{x}, \mathbf{y}) \tag{5}$$

$$\frac{\partial \mathbf{y}}{\partial t} = \mathbf{C}(\mathbf{x}, \mathbf{y}) \tag{6}$$

where

$$\mathbf{x} = (\theta, q_v, q_c, q_r, K_m) \qquad \mathbf{y} = (u, v, w, \pi) \tag{7}$$

Here q_v, q_c, and q_r are the mixing ratios of water vapor, cloud water, and rainwater, respectively; θ and K_m are the potential temperature and the SGS eddy coefficient for momentum, respectively; u, v, and w are the x , y, and z components of the wind velocity, respectively; and π is the deviation of the nondimensional pressure.

This is the model in differential form. Once it has been discretized in space using finite differences, the model is written as a set of nonlinear-coupled ordinary differential equations. Next, once a time-difference scheme is used, it becomes a set of nonlinear-coupled difference equations.

Let $\mathbf{x}(t_0)$, $\mathbf{y}(t_0)$ denote the initial condition. The time-difference scheme integrates the model equation by marching outward from the initial condition. The initial condition defines a unique solution $\mathbf{x}(t)$, $\mathbf{y}(t)$ to the model equation.

4 COST FUNCTION

The cost function is taken as

$$J = \sum_{i=0}^{N} H[\mathbf{x}(t_i), \mathbf{y}(t_i)]\, dt \tag{8}$$

where $H[\mathbf{x}(t_i), \mathbf{y}(t_i)]$ is a scalar measuring the distance between $\mathbf{x}(t_i)$, $\mathbf{y}(t_i)$ and its observations available at time $t_i = i\Delta t$; Δt is the time interval. The available observations are assumed to be distributed over a limited time interval $[t_0, t_N]$. The constraint is the model equation. For a given initial condition and for the corresponding solution $\mathbf{x}(t_i)$, $\mathbf{y}(t_i)$ of the model, the cost function is evaluated. Thus, the cost function is regarded as a function of $\mathbf{x}(t_0)$, $\mathbf{y}(t_0)$.

Hence, assimilation of meteorological observations is formulated as a large-scale nonlinear optimization problem. This problem has thousands or millions of variables. Modified L-BFGS-B algorithm [6, 7] is employed in order to solve this problem.

5 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 source program [5]. Figure 1 and Figure 2 show the evolution of the rainwater and flow, respectively. Out of the simulation result, the x component of the wind velocity and the rainwater are assumed to be observations from a Doppler radar.

In order to estimate the initial condition used for the simulation, the modified L-BFGS-B algorithm solves the large-scale optimization problem. Starting from the initial guess, the L-BFGS-B algorithm repeats search until it reaches the optimal point. At each search, the gradient of the cost function is computed using the adjoint program [8]. Figure 3 shows the rainwater and flow fields of the starting point used in the optimization.

Figure 4 shows the convergence history of the optimization.

Figure 5 compares the rainwater field recovered by the optimization and the corresponding one of the initial condition in the $x - z$ cross section at $y = 0$. Figure 6 compares the flow field recovered by optimization and the corresponding one of the initial condition in the $x - z$ cross section at $y = 0$.

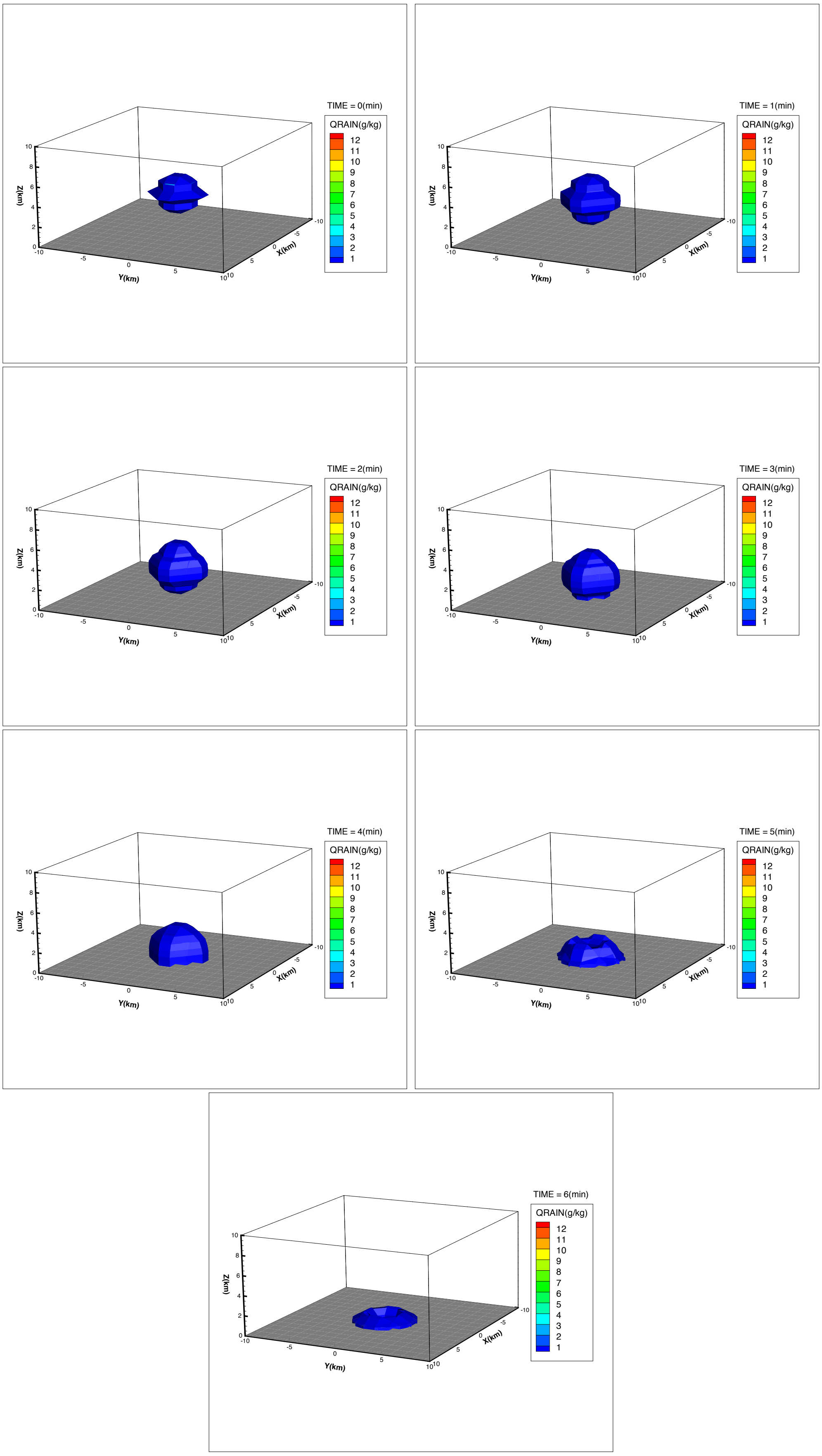

Figure 1: Evolution of the rainwater field after the downburst initiation.

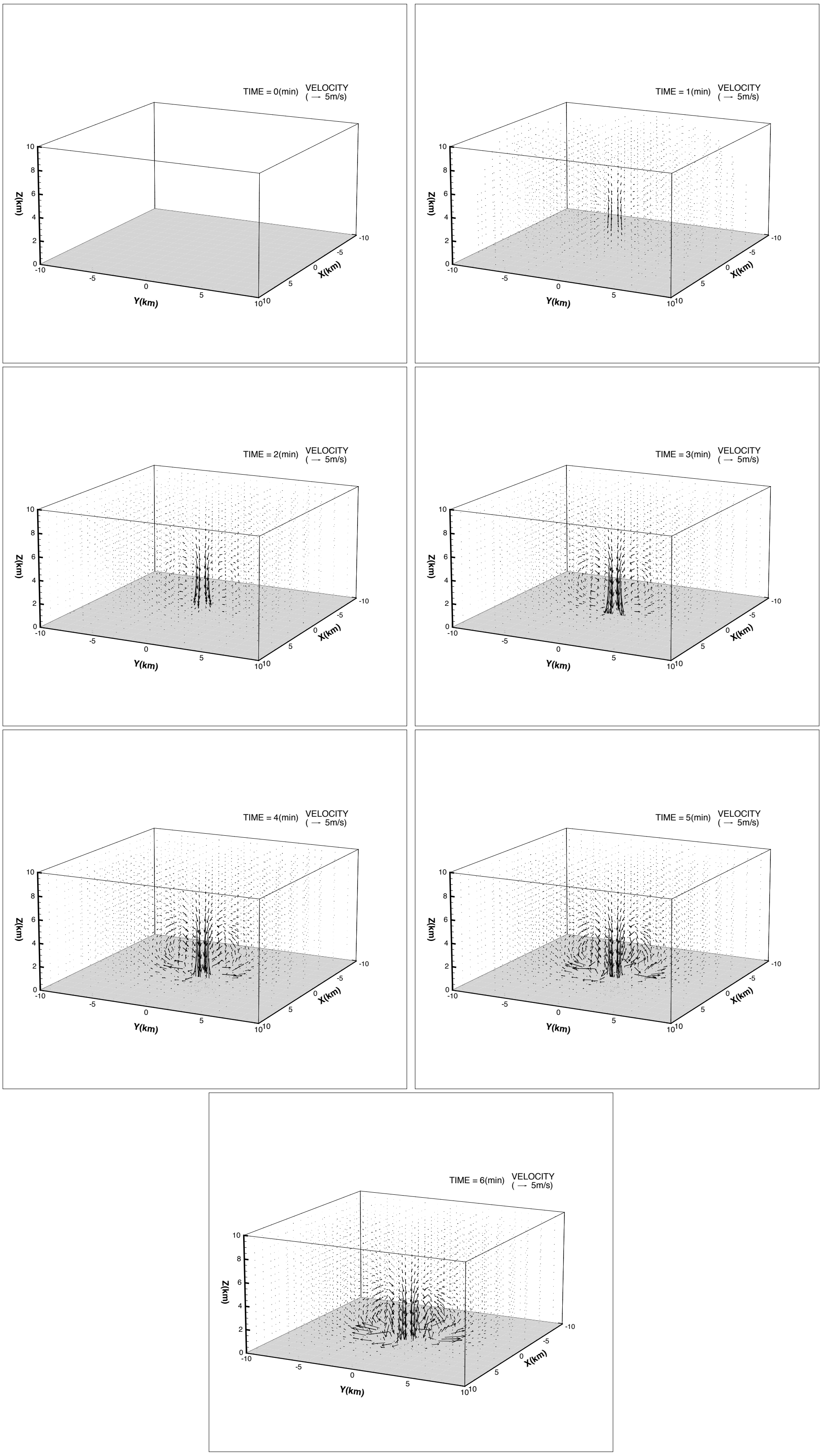

Figure 2: Evolution of the flow field after the downburst initiation.

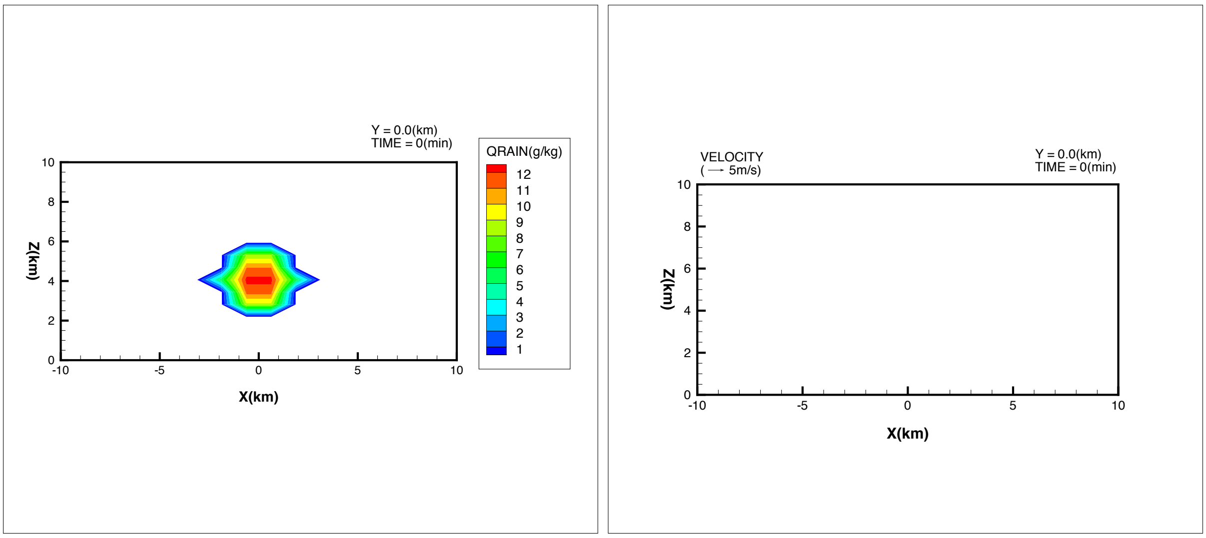

Figure 3: Starting point for the optimization; the rainwater (left) and flow (right) fields in the $x - z$ cross section at $y = 0$.

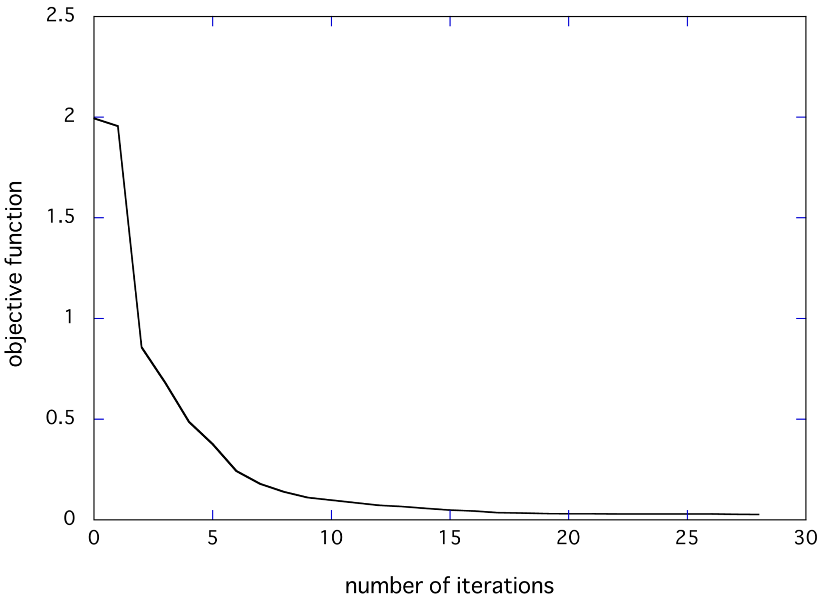

Figure 4: Convergence history of the optimizatiot.

6 CONCLUSIONS

- Meteorological data assimilation is formulated as a non-linear optimization problem.
- A source computer program that simulates the atmospheric flow is differentiated by automatic differentiation tool TAPENADE.
- The adjoint program generated by TAPENADE is used to compute the gradient of a cost function
- Numerical experiments are done for a three-dimensional downburst.

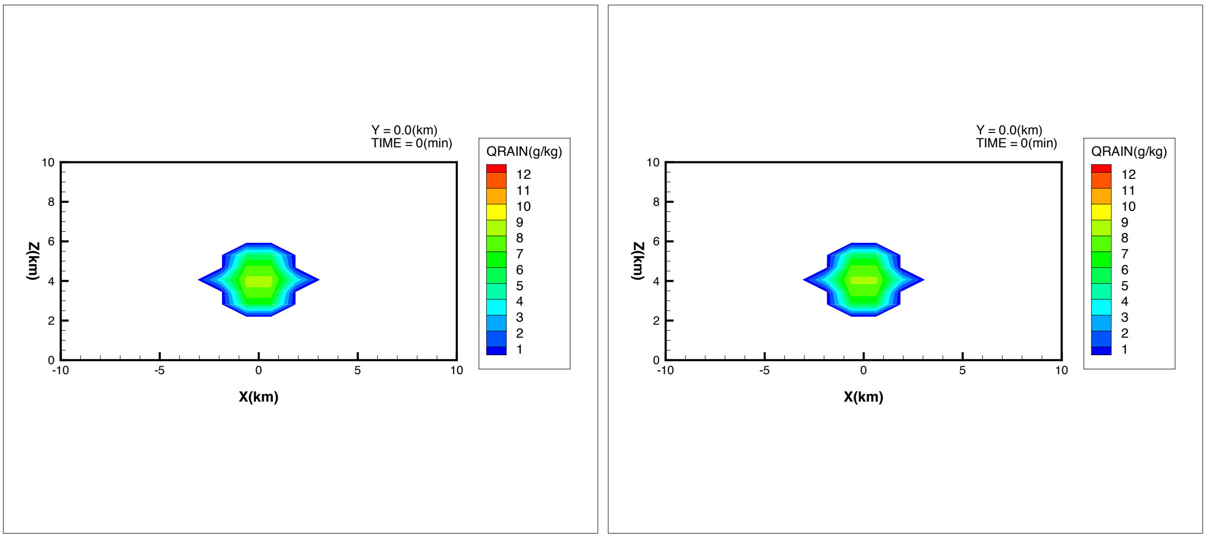

Figure 5: The rainwater field recovered by the optimization (left) and the corresponding one of the initial condition for the simulation (right) in the $x - z$ cross section at $y = 0$.

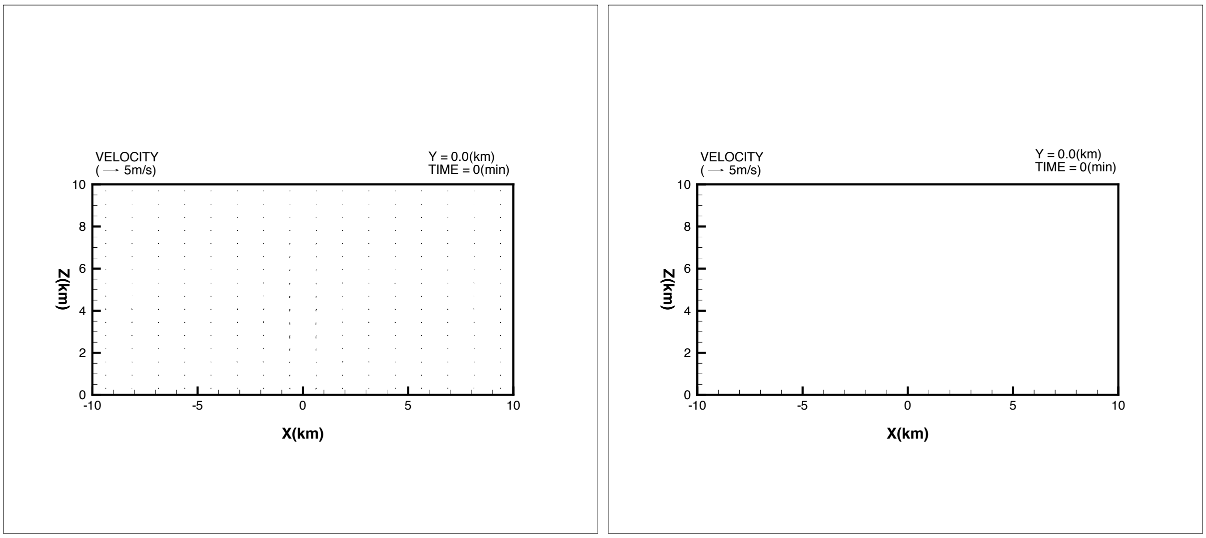

Figure 6: The flow field recovered by the optimization (left) and the corresponding one of the initial condition for the simulation (right) in the $x - z$ cross section at $y = 0$.

- The recovered fields agree reasonably with the initial condition used for observation generation.

REFERENCES

[1] O. Talagrand, Assimilation of Observations, An Introduction, *J. Met. Soc. Japan*, Special Issue **75**, 1B, pp. 191-209, 1997.

[2] Y. Horibata, Gradient Computation of a Nonlinear Programming Problem for Meteorological Assimilation, *Proc. of European Conference on Computational Fluid Dynamics*, ECCOMAS CFD 2006, Egmond aan Zee, Netherlands, Paper 405, 2006.

[3] Y. Horibata, Comparison of algorithms for solving large-scale optimization problems with bound constraints for meteorological data assimilation, *Proceedings of the 6th European Congress on Computational Methods in Applied Sciences and Engineering*, ECCOMAS 2012, Vienna, Austria, Paper 2811, 2012.

[4] L. Hascoet, V. Pascual, TAPENADE 2.1 user's guide, INRIA, France, 2004.

[5] Y. Horibata, H. Oikawa, Numerical Simulation of Downbursts Using a Terrain-following Coordinate Transformation, *Proc. of the Second European Computational Fluid Dynamics Conference*, Stuttgart, Germany, pp. 1002-1009, 1994.

[6] C. Zhu, R. H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-B, FORTRAN subroutines for large scale bound constrained optimization, *ACM Transactions on Mathematical Software*, **23**, pp. 550-560, 1997.

[7] J.L. Morales, J. Nocedal, L-BFGS-B: Remark on Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization, *ACM Transactions on Mathematical Software*, **23**, Article 7, 2011.

[8] Y. Horibata, Adjoint program generated by automatic differentiation of a meteorological simulation program, and its application to gradient computation, *Proceedings of the 6th International Conference on Computational Methods*, 14th - 17th July 2015, Auckland, G.R. Liu and Raj Das, Paper ID 861, ScienTech Publisher, New Zealand, 2015.