ISSN No. 0737-5352-23

DATA ASSIMILATION AND THE ADJOINT METHOD: An Example

Gerald D. Taylor
Department of Mathematics
and
Center for Geosciences

Research Supported by U.S. Army Research Office
under Grant No. DAAL03-86-K-0175

CIRA Cooperative Institute for Research in the Atmosphere

Colorado State University

DATA ASSIMILATION AND THE ADJOINT METHOD:
An Example

by

Gerald D. Taylor

Department of Mathematics
and
Center for Geosciences
Colorado State University
Fort Collins, Colorado 80523

This research was supported by ARO
under Grant No. DAAL03-86-K-0175 through the
U.S. Army Center for Geosciences and CIRA

May 1991

Abstract

The following are copies of slides of a talk presenting a method of four dimensional data assimilation based on the adjoint method via considering a simple example.

DATA ASSIMILATION AND THE ADJOINT METHOD

The following are copies of the slides of a talk presenting some facts concerning the adjoint method via considering a simple example.

Data assimilation, the effective integration of observations into predictive dynamical equations has been a major topic of investigation for some 30 years in meteorology.

* Using optimization theory, Le Dimet and Talagrand (1986) and Talagrand and Courtier (1987a, 1987b, 1990) have proposed an approach for this problem called the adjoint method.

* Define a real valued function measuring the "distance" between the model solution corresponding to a given initialization and the available observations. The goal is to select an initialization that minimizes this distance.

* Using the adjoint method, the gradient of this distance function with respect to the initial condition can be calculated so that a gradient based minimization algorithm can be applied.

* Thus, using this initialization the model will predict values that most closely fit the known observational data.

* The current wisdom is that this procedure is most effective when applied to the actual predictive equations used in the dynamical model.

* Since these codes are (usually) based upon some sort of discretization, the above minimization problem becomes the minimization of a function of several variables and the adjoint method is an application of the chain rule for calculating the gradient of this function.

As a simple example of this approach consider the following continuous model: Given $u_0(z,0)$, *find* $u(z,t)$, for $0 \le z \le L$ and $t > 0$, such that

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial z} &= 0, \quad 0 \le z \le L, \quad t > 0, \\ u(0,t) &= u(L,t), \quad t > 0, \\ u(z,0) &= u_0(z,0), \quad 0 \le z \le L. \end{aligned} \tag{1}$$

Discretizing in space, by setting $z_i = ih, h = L/N$, $u_i(t) = u(z_i,t)$ for $i = 0, \cdots, N$ and applying centered differences, gives the initial value system of N ordinary differential equations:

$$\begin{aligned} \frac{du_i(t)}{dt} &= -u_i(t)\left(\frac{u_{i+1}(t) - u_{i-1}(t)}{2h}\right), \quad i = 0, \cdots, N-1 \\ u_i(0) &= u_0(z_i,0), \quad i = 0, \cdots, N-1 \end{aligned} \tag{2}$$

where $u_N(t) = u_0(t)$ and $u_{-1}(t) = u_{N-1}(t)$ for all $t \ge 0$.

Writing

$$\mathbf{x}(t) = \big(u_0(t), \cdots, u_{N-1}(t)\big)^T \text{ and } \mathbf{F}(\mathbf{x}) = (f_0(\mathbf{x}), \cdots, f_{N-1}(\mathbf{x}))^T, \tag{3}$$

where

$$f_j(\mathbf{x}) = -u_j\left(\frac{u_{j+1} - u_{j-1}}{2h}\right), \quad j = 0, \cdots, N-1,$$

this system can be rewritten in vector form as

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \mathbf{F}(\mathbf{x}(t)), \\ \mathbf{x}(0) &= \mathbf{x}_0 = \big(u_0(0), \cdots, u_{N-1}(0)\big)^T. \end{aligned} \tag{4}$$

At this point a time discretization (advection scheme) must be selected. We shall consider the Adams-Bashforth method of order 2. Thus, the advection formulas are given by

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_0 + h\mathbf{F}(\mathbf{x}_0) \\ \mathbf{x}_j &= \mathbf{x}_{j-1} + \frac{h}{2}[3\mathbf{F}(\mathbf{x}_{j-1}) - \mathbf{F}(\mathbf{x}_{j-2})] \quad j \ge 2, \end{aligned} \tag{5}$$

where $\mathbf{F}$ is given by (3) and here the solution $\mathbf{x}_j$ is an approximation for $\big(u(z_0, jh), \cdots, u(z_{N-1}, jh)\big)^T$.

Now, suppose that the exact initialization for (1) is not known, but that observations exist for certain values of $u(z_i, t)$ at some distinct times t_j. Then, one seeks to determine an initialization, $\mathbf{x}_0 = (x_0^0, \cdots, x_{N-1}^0)^T$, for which the model solution of (1) corresponding to this initialization is the "closest" to these observations from the class of all possible initializations.

* For a given initialization $\mathbf{x}_0$ define target values $\mathbf{x}_j^{obs}$ corresponding to the set of advected values $\mathbf{x}_j$ predicted by $\mathbf{x}_0$ by requiring that $\mathbf{x}_j^{obs}$ have the components determined by the observational data where ever possible and have all remaining components identical with those of the of model solution.

* Next, assuming all observations occur by the time step $t_r = rh$, define a cost function $J(\mathbf{x}_0)$ by

$$J(\mathbf{x}_0) = \sum_{j=0}^{r} (\mathbf{x}_j - \mathbf{x}_j^{obs})^T (\mathbf{x}_j - \mathbf{x}_j^{obs}). \tag{6}$$

* Because of the manner in which $\mathbf{x}_j^{obs}$ has been defined, $J(\mathbf{x}_0)$ is precisely the sum of the squares of the differences of the model values and observational values at all components where the observational data can be represented.

* In this setting the problem of data assimilation *is taken* to mean

$$solve \;\; min\{J(\mathbf{x}_0) : \mathbf{x}_0 \in \mathbb{R}^N\}. \tag{7}$$

* This can be done using a black box gradient based (iterative) minimization routine (e.g. Buckley (1985)) when one is using the adjoint method since this method calculates $\nabla_{\mathbf{x}_0} J(\mathbf{x}_0)$, the gradient of $J(\mathbf{x}_0)$ at $\mathbf{x}_0$, which is required at each iteration step of such a minimization scheme.

Since, in this setting, one must simply apply the chain rule to find the gradient of $J(\mathbf{x_0})$ with respect to $\mathbf{x_0}$, we shall now summarize some derivatives rules for functions of several variables.

* If $f : \mathbb{R}^N \rightarrow \mathbb{R}$ is a differentiable function then the derivative of f with respect to $\mathbf{x} \in \mathbb{R}^n$ is the gradient of f and is given by

$$\mathbf{D_x} f = \nabla_{\mathbf{x}} f = (\frac{\partial f}{\partial x_0}, \cdots, \frac{\partial f}{\partial x_{N-1}}) \text{ where } \mathbf{x} = (x_0, \cdots, x_{N-1}).$$

* If $\mathbf{y} \in \mathbb{R}^N$ then one writes

$$\mathbf{D_x} f(\mathbf{y}) = \nabla_{\mathbf{x}} f(\mathbf{y}) = (\frac{\partial f}{\partial x_0}(\mathbf{y}), \cdots, \frac{\partial f}{\partial x_{N-1}}(\mathbf{y}))$$

to represent the gradient of f evaluated at $\mathbf{y}$.

* If $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a differentiable function then $\mathbf{F} = (f_0, \cdots, f_{N-1})^T$ where each $f_j : \mathbb{R}^N \rightarrow \mathbb{R}$ is a differentiable function, for $j = 0, \cdots, N-1$.

* The derivative of $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ with respect to $\mathbf{x} \in \mathbb{R}^N$ is the jacobian of $\mathbf{F}$ and is given by

$$\mathbf{D_x F} = \begin{pmatrix} \frac{\partial f_0}{\partial x_0} & \frac{\partial f_0}{\partial x_1} & \cdots & \frac{\partial f_0}{\partial x_{N-1}} \\ \frac{\partial f_1}{\partial x_0} & \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{N-1}}{\partial x_0} & \frac{\partial f_{N-1}}{\partial x_1} & \cdots & \frac{\partial f_{N-1}}{\partial x_{N-1}} \end{pmatrix}$$

* If $\mathbf{y} \in \mathbb{R}^N$, then one writes

$$\mathbf{D_x F}(\mathbf{y}) = \begin{pmatrix} \frac{\partial f_0}{\partial x_0}(\mathbf{y}) & \frac{\partial f_0}{\partial x_1}(\mathbf{y}) & \cdots & \frac{\partial f_0}{\partial x_{N-1}}(\mathbf{y}) \\ \frac{\partial f_1}{\partial x_0}(\mathbf{y}) & \frac{\partial f_1}{\partial x_1}(\mathbf{y}) & \cdots & \frac{\partial f_1}{\partial x_{N-1}}(\mathbf{y}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{N-1}}{\partial x_0}(\mathbf{y}) & \frac{\partial f_{N-1}}{\partial x_1}(\mathbf{y}) & \cdots & \frac{\partial f_{N-1}}{\partial x_{N-1}}(\mathbf{y}) \end{pmatrix}$$

to denote the jacobian of $\mathbf{F}$ evaluated at $\mathbf{y}$.

At this point in our survey of derivative rules in $\mathbb{R}^N$ let $\mathbf{x}_0 = (x_0^0, \cdots, x_{N-1}^0)$ represent the independent variable of $\mathbb{R}^N$.

* First, by the product rule, the first term of the sum defining J gives

$$\begin{aligned}\mathbf{D}_{\mathbf{x}_0}\Big((\mathbf{x}_0 - \mathbf{x}_0^{obs})^T(\mathbf{x}_0 - \mathbf{x}_0^{obs})\Big) &= 2(\mathbf{x}_0 - \mathbf{x}_0^{obs})^T\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_0 \\ &= 2(\mathbf{x}_0 - \mathbf{x}_0^{obs})^T\end{aligned}$$

since $\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_0 = I$, the $N \times N$ identity matrix on $\mathbb{R}^N$.

* Since the advected vectors $\mathbf{x}_j$ of the time differencing scheme (5) are all functions of $\mathbf{x}_0$, their derivatives with respect to $\mathbf{x}_0$ can be calculated using the chain rule of advanced calculus.

* For the first step giving $\mathbf{x}_1$ we have that

$$\begin{aligned}\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_1 &= \mathbf{D}_{\mathbf{x}_0}\Big(\mathbf{x}_0 + h\mathbf{F}(\mathbf{x}_0)\Big) \\ &= I + h\mathbf{D}_{\mathbf{x}_0}\mathbf{F}(\mathbf{x}_0)\end{aligned} \tag{8}$$

where $\mathbf{F}$ defined in (3) is a differentiable mapping of $\mathbb{R}^N$ into $\mathbb{R}^N$ and $\mathbf{D}_{\mathbf{x}_0}\mathbf{F}(\mathbf{x}_0)$ is the jacobian (matrix) of $\mathbf{F}$ evaluated at $\mathbf{x}_0$.

* For the example being consider here (2) - (5), $\mathbf{D}_{\mathbf{x}_0}\mathbf{F}(\mathbf{x}_0)$ is a structured sparse $N \times N$ matrix. Indeed, we have for this particular $\mathbf{F}$ that

$$\mathbf{D}_{\mathbf{x}_0}\mathbf{F}(\mathbf{x}_0) = \frac{1}{2h}\begin{pmatrix} (x_1^0 - x_{N-1}^0) & x_0^0 & 0 & \cdots & 0 & -x_0^0 \\ -x_1^0 & (x_2^0 - x_0^0) & x_1^0 & \cdots & 0 & 0 \\ 0 & -x_2^0 & (x_3^0 - x_1^0) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{N-1}^0 & 0 & 0 & \cdots & -x_{N-1}^0 & (x_0 - x_{N-2}^0) \end{pmatrix} \tag{9}$$

since each $f_j(\mathbf{x}_0) = \frac{1}{2h}x_j^0(x_{j+1}^0 - x_{j-1}^0)$ with the periodicity that $x_{-1}^0 = x_{N-1}^0$ and $x_N^0 = x_0^0$ holding for $j = 0, \cdots, N-1$ where $\mathbf{x}_0 = (x_0^0, \cdots, x_{N-1}^0)^T \epsilon\, \mathbb{R}^N$.

* Thus, the derivative with respect to $\mathbf{x}_0$ of the second term of the sum defining $J(\mathbf{x}_0)$ is given by

$$\mathbf{D}_{\mathbf{x}_0}\Big((\mathbf{x}_1 - \mathbf{x}_1^{obs})^T(\mathbf{x}_1 - \mathbf{x}_1^{obs})\Big) = 2(\mathbf{x}_1 - \mathbf{x}_1^{obs})^T\mathbf{D}_{\mathbf{x}_1}\mathbf{x}_0$$

where the matrix $\mathbf{D}_{\mathbf{x}_1}\mathbf{x}_0$ is given in (8) and (9).

* Continuing, one sees that for the second advection step giving $\mathbf{x}_2$ we have by the chain rule that

$$\begin{aligned}\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_2 &= \mathbf{D}_{\mathbf{x}_0}\Big(\mathbf{x}_1 + \frac{h}{2}[3\mathbf{F}(\mathbf{x}_1) - \mathbf{F}(\mathbf{x}_0)]\Big) \\ &= \mathbf{D}_{\mathbf{x}_0}\mathbf{x}_1 + \frac{h}{2}[3\mathbf{D}_{\mathbf{x}_1}\mathbf{F}(\mathbf{x}_1)\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_1 - \mathbf{D}_{\mathbf{x}_0}\mathbf{F}(\mathbf{x}_0)]\end{aligned} \tag{10}$$

where $\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_1$ is defined in (8) and (9), $\mathbf{D}_{\mathbf{x}_0}\mathbf{F}(\mathbf{x}_0)$ is defined in (9) and $\mathbf{D}_{\mathbf{x}_1}\mathbf{F}(\mathbf{x}_1)$ is given by (9) with x_j^0 replaced by x_j^1 throughout where $\mathbf{x}_1 = \left(x_0^1, \cdots, x_{N-1}^1\right)^T$.

* From this, it follows that the derivative with respect to $\mathbf{x}_0$ of the third term of the sum defining $J(\mathbf{x}_0)$ satisfies

$$\mathbf{D}_{\mathbf{x}_0}\Big((\mathbf{x}_2 - \mathbf{x}_2^{obs})^T(\mathbf{x}_2 - \mathbf{x}_2^{obs})\Big) = 2(\mathbf{x}_2 - \mathbf{x}_2^{obs})^T\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_2$$

where $\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_2$ is given in (10).

* This procedure can be continued, however, an iterative scheme for calculating the gradient of $J(\mathbf{x}_0)$ can be constructed if one differentiates from the last term of the sum back to the first.

* A second procedure for developing an iterative scheme for calculating the derivative is a Lagrange multiplier type of procedure that can be used in a general setting. We shall finish this example by illsutrating this technique.

* In order to describe the Lagrange multiplier approach for generating the adjoint scheme corresponding to this example, let us recall that the cost function is given by the sum (6)

$$J(\mathbf{x}_0) = \sum_{j=0}^{r} (\mathbf{x}_j - \mathbf{x}_j^{obs})^T (\mathbf{x}_j - \mathbf{x}_j^{obs}).$$

where the advected values and target values have been defined in the above discussion.

* We begin by introducing a function H of the the complete set of vectors $\mathbf{x}_0, \cdots, \mathbf{x}_n$ and an additional set of r vectors in $\mathbb{R}^N$ denoted by $\lambda_1, \cdots, \lambda_r$ as follows

$$\begin{aligned} H(\mathbf{x}_0, \cdots, \mathbf{x}_r, \lambda_1, \cdots, \lambda_r) = & \sum_{j=0}^{r} (\mathbf{x}_j - \mathbf{x}_j^{obs})^T (\mathbf{x}_j - \mathbf{x}_j^{obs}) \\ & + \lambda_1^T \big(\mathbf{x}_0 + h\mathbf{F}(\mathbf{x}_0)\big) \\ & + \sum_{j=2}^{r} \lambda_j^T \Big[\mathbf{x}_{j-1} + \frac{h}{2}\big(3\mathbf{F}(\mathbf{x}_{j-1}) - \mathbf{F}(\mathbf{x}_{j-2})\big) - \mathbf{x}_j\Big] \end{aligned} \tag{11}$$

* Note that when we require the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_r$ to be defined by the Adams-Bashforth method of order 2 then H reduces to J.

* Now, considering all the arguments of H to be functions of $\mathbf{x}_0$ we have by the chain rule that

$$\begin{aligned} & \nabla_{\mathbf{x}_0} H(\mathbf{x}_0, \cdots, \mathbf{x}_r, \lambda_1, \cdots, \lambda_r) = \\ & \quad \Big[2(\mathbf{x}_0 - \mathbf{x}_0^{obs})^T + \lambda_1^T + h\big(\lambda_1^T - \frac{1}{2}\lambda_2^T\big)\mathbf{D}_{\mathbf{x}_0}\mathbf{F}(\mathbf{x}_0)\Big] \\ & \quad + \Big[2(\mathbf{x}_1 - \mathbf{x}_1^{obs})^T + \lambda_2^T + \frac{h}{2}\Big[\big(3\lambda_2^T - \lambda_3^T\big)\mathbf{D}_{\mathbf{x}_1}\mathbf{F}(\mathbf{x}_1) - \lambda_1^T\Big]\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_1 \\ & \quad + \sum_{j=2}^{r} \Big[2(\mathbf{x}_j - \mathbf{x}_j^{obs})^T + \lambda_{j+1}^T + \frac{h}{2}\big(3\lambda_{j+1}^T - \lambda_{j+2}^T\big)\mathbf{D}_{\mathbf{x}_j}\mathbf{F}(\mathbf{x}_j) - \lambda_j^T\Big]\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_j \\ & \quad + \Big[\mathbf{x}_0 + h\mathbf{F}(\mathbf{x}_0) - \mathbf{x}_1\Big]\mathbf{D}_{\mathbf{x}_0}\lambda_1 \\ & \quad + \sum_{j=2}^{r} \Big[\mathbf{x}_{j-1} + \frac{h}{2}\big(3\mathbf{F}(\mathbf{x}_{j-1}) - \mathbf{F}(\mathbf{x}_{j-2})\big) - \mathbf{x}_j\Big]^T \mathbf{D}_{\mathbf{x}_0}\lambda_j. \end{aligned} \tag{12}$$

* We shall now assume that we are using the Adams-Bashforth method as our time advection scheme, so that the last summation of (12) is zero. Thus, (12) reduces to

$$\begin{aligned}
&\nabla_{\mathbf{x}_0} H(\mathbf{x}_0, \cdots, \mathbf{x}_r, \lambda_1, \cdots, \lambda_r) = \\
&\quad \Big[2(\mathbf{x}_0 - \mathbf{x}_0^{obs})^T + \lambda_1^T + h(\lambda_1^T - \frac{1}{2}\lambda_2^T)\mathbf{D}_{\mathbf{x}_0}\mathbf{F}(\mathbf{x}_0)\Big] \\
&\quad + \Big[2(\mathbf{x}_1 - \mathbf{x}_1^{obs})^T + \lambda_2^T + \frac{h}{2}\Big[(3\lambda_2^T - \lambda_3^T)\mathbf{D}_{\mathbf{x}_1}\mathbf{F}(\mathbf{x}_1) - \lambda_1^T\Big]\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_1 \\
&\quad + \sum_{j=2}^{r}\Big[2(\mathbf{x}_j - \mathbf{x}_j^{obs})^T + \lambda_{j+1}^T \\
&\qquad + \frac{h}{2}(3\lambda_{j+1}^T - \lambda_{j+2}^T)\mathbf{D}_{\mathbf{x}_j}\mathbf{F}(\mathbf{x}_j) - \lambda_j^T\Big]\mathbf{D}_{\mathbf{x}_0}\mathbf{x}_j.
\end{aligned} \tag{13}$$

* At this point we note that (12) and (13) are true for any choice of the parameters $\lambda_1, \cdots, \lambda_r$. Hence, we recursively select them to force as many of the remaining terms of (13) to be zero as possible starting with the terms with the largest index and working back through these indices. This will give a recursive procedure for calculating $\nabla_{\mathbf{x}_0} J(\mathbf{x}_0)$.

* Thus, we define for notational convenience, $\lambda_{r+1} = \lambda_{r+2} = \mathbf{0}$. We then define the λ_j parameters of (13) recursively for $j = r, r-1, \cdots, 1$ by requiring that (where $\mathbf{D}_{\mathbf{x}}\mathbf{F}(\mathbf{y})$ denotes the jacobian of $\mathbf{F}$ evaluated at $\mathbf{y}$)

$$\lambda_j^T = 2(\mathbf{x}_j - \mathbf{x}_j^{obs})^T + \lambda_{j+1}^T + \frac{h}{2}(3\lambda_{j+1}^T - \lambda_{j+2}^T)\mathbf{D}_{\mathbf{x}_j}\mathbf{F}(\mathbf{x}_j). \tag{14}$$

* Note that this defines the complete set $\lambda_1, \cdots, \lambda_r$ of parameters and that the two final parameters λ_1, λ_2) calculated give the gradient of J according to

$$\nabla_{\mathbf{x}_0} J(\mathbf{x}_0) = 2(\mathbf{x}_0 - \mathbf{x}_0^{obs})^T + \lambda_1^T + h(\lambda_1^T - \frac{1}{2}\lambda_2^T)\mathbf{D}_{\mathbf{x}_0}\mathbf{F}(\mathbf{x}_0). \tag{14}$$

* As a final note, we wish to observe that the above development could be done completely componentwise using summations rather than matrices and vectors. Although this approach would probably be more tedious, it would actually exhibit the precise dependence of all the variables and probably lead to correct Jacobians in a somewhat more straight forward manner.

* It would seem that this approach could be effectively coupled with a symbolic algebra package to compute the needed partial derivatives.

* If one could couple with this a symbolic package that could also determine the structure of all matrices involved so that matrix vector multiplies could be replaced by vector products, then this would be real close to giving an effective automated procedure for the adjoint method. This would be useful since it would seem that in a production code the size of the vectors would dictate that matrix multiplies be avoided.

* A copy of these slides is available, as well, a more general description of these ideas. This latter description treats advection schemes corresponding to a Runge-Kutta second order method and a combining of the above Adams-Bashforth explicit scheme with a second order Adams-Moulton correction to get a combined explicit and implicit scheme. In addition, the development of the adjoint method for a FORTRAN code given in appendix F of the book "An Introduction to Three Dimensional climate Modeling" by Washington and Parkinson is began. (This code is a benchmark weather prediction program for comparing the performance of supercomputers that was coded by P. Swarztrauber in 1984 at NCAR. It is based on a paper by Sadourny, (1975) studying the dynamics of finite difference models of the shallow water equations.) Finally, included in these notes is an appendix giving some facts about derivatives of functions of several variables.

References

[1975] Sadourny, R., The dynamics of finite difference models of the shallow water equations, *J. Atm. Sci.*, **32**, 680 - 689.

[1986] Washington, W. M. and Parkinson, C. L., An Introduction to Three Dimensional Climate Modeling, *Oxford Press*, New York.

[1985] Buckley, A., ALGORITHM 630: BBVSCG - A variable-storage algorithm for function minimization, *ACM Trans. on Math. Soft.*, **11**, 103 - 119.

[1986] Le Dimet, F. X. and Talagrand, O., Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects, *Tellus*, **38A**, 97 - 110.

[1987a] Talagrand, O. and Courtier, P., Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory, *Q.J.R. Meteorol. Soc.*, **113**, 1311 - 1328.

[1987b] Talagrand, O. and Courtier, P., Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Numerical results, *Q.J.R. Meteorol. Soc.*, **113**, 1329-1347.

[1990] Courtier, P. and Talagrand, O., Variational assimilation of meteorological observations with the direct and adjoint shallow-equation, *Tellus*, **42A**, 531-549.