École Doctorale des Sciences de l'Environnement d'Île-de-France
Année 2007-2008

Modélisation Numérique de l'Écoulement Atmosphérique et Assimilation d'Observations

Olivier Talagrand
Cours 7
10 Juin 2008

Best Linear Unbiased Estimate

Available data consist of

Background $x^b = x + \zeta^b$

'Observations' $y = Hx + \varepsilon$

Errors assumed to be unbiased, $E(\zeta^b \zeta^{bT}) = P^b$, $E(\varepsilon\varepsilon^T) = R$, $E(\zeta^b \varepsilon^T) = 0$

BLUE

$$x^a = x^b + P^b H^T [HP^b H^T + R]^{-1} (y - Hx^b)$$
$$P^a = P^b - P^b H^T [HP^b H^T + R]^{-1} HP^b$$

Minimization of scalar *objective function*, defined on state space

$$\xi \in \mathcal{S} \rightarrow$$

$$\mathcal{J}(\xi) = (1/2)\,(x^b - \xi)^{\mathrm{T}}\,[P^b]^{-1}\,(x^b - \xi) + (1/2)\,(y - H\xi)^{\mathrm{T}}\,R^{-1}\,(y - H\xi)$$

$$= \mathcal{J}_b + \mathcal{J}_o$$

leads to

$$\boldsymbol{x}^a = \boldsymbol{x}^b + P^a\,H^{\mathrm{T}}\,R^{-1}\,(\boldsymbol{y} - H\boldsymbol{x}^b)$$
$$[P^a]^{-1} = [P^b]^{-1} + H^{\mathrm{T}}\,R^{-1}H$$

which has been said to be another, equivalent, set of equations for the *BLUE*.

Approach can easily be extended to time dimension.

Suppose for instance available data consist of

- Background estimate at time 0

  $x_0^b = x_0 + \xi_0^b \qquad E(\xi_0^b \xi_0^{bT}) = P_0^b$

- Observations at times $k = 0, \ldots, K$

  $y_k = H_k x_k + \varepsilon_k \qquad E(\varepsilon_k \varepsilon_j^T) = R_k$

- Model (supposed for the time being to be exact)

  $x_{k+1} = M_k x_k \qquad k = 0, \ldots, K\text{-}1$

Errors assumed to be unbiased and uncorrelated in time, H_k and M_k linear

Then objective function

$\xi_0 \in S \rightarrow$

$$\mathcal{J}(\xi_0) = (1/2)\,(x_0^b - \xi_0)^T [P_0^b]^{-1} (x_0^b - \xi_0) + (1/2)\,\Sigma_k [y_k - H_k \xi_k]^T R_k^{-1} [y_k - H_k \xi_k]$$

subject to $\xi_{k+1} = M_k \xi_k, \qquad k = 0, \ldots, K\text{-}1$

$$\mathcal{J}(\xi_0) \;=\; (1/2)\,(x_0{}^b - \xi_0)^{\mathrm{T}}\,[P_0{}^b]^{-1}\,(x_0{}^b - \xi_0) + (1/2)\,\Sigma_k[y_k - H_k\xi_k]^{\mathrm{T}}\,R_k^{-1}\,[y_k - H_k\xi_k]$$

Background is not necessary, if observations are in sufficient number to overdetermine the problem. Nor is strict linearity.

How to minimize objective function with respect to initial state $u = \xi_0$ (u is called the *control variable* of the problem) ?

Use iterative minimization algorithm, each step of which requires the explicit knowledge of the local gradient $\nabla_u \mathcal{J} \equiv (\partial\mathcal{J}/\partial u_i)$ of $\mathcal{J}$ with respect to u.

Gradient computed by *adjoint method*.

How to numerically compute the gradient $\nabla_u \mathcal{J}$?

Direct perturbation, in order to obtain partial derivatives $\partial \mathcal{J} / \partial u_i$ by finite differences ? That would require as many explicit computations of the objective function $\mathcal{J}$ as there are components in u. Practically impossible.

Adjoint Method

Input vector $\boldsymbol{u} = (u_i)$, $\dim \boldsymbol{u} = n$

Numerical process, implemented on computer (*e. g.* integration of numerical model)

$$\boldsymbol{u} \rightarrow \boldsymbol{v} = \boldsymbol{G}(\boldsymbol{u})$$

$\boldsymbol{v} = (v_j)$ is *output vector* , $\dim \boldsymbol{v} = m$

Perturbation $\delta \boldsymbol{u} = (\delta u_i)$ of input. Resulting first-order perturbation on $\boldsymbol{v}$

$$\delta v_j = \Sigma_i \, (\partial v_j / \partial u_i) \; \delta u_i$$

or, in matrix form

$$\delta \boldsymbol{v} \; = \; \boldsymbol{G'} \, \delta \boldsymbol{u}$$

where $\boldsymbol{G'} \equiv (\partial v_j / \partial u_i)$ is local matrix of partial derivatives, or jacobian matrix, of $\boldsymbol{G}$.

Adjoint Method (continued 1)

$$\delta \boldsymbol{v} \;=\; \boldsymbol{G}'\,\delta \boldsymbol{u} \qquad \text{(D)}$$

Scalar function of output

$$\mathcal{J}(\boldsymbol{v}) \;=\; \mathcal{J}[\boldsymbol{G}(\boldsymbol{u})]$$

Gradient $\nabla_{\boldsymbol{u}} \mathcal{J}$ of $\mathcal{J}$ with respect to input $\boldsymbol{u}$?

'Chain rule'

$$\partial \mathcal{J}/\partial u_i = \Sigma_j \, \partial \mathcal{J}/\partial v_j \, (\partial v_j/\partial u_i)$$

or

$$\nabla_{\boldsymbol{u}} \mathcal{J} \;=\; \boldsymbol{G}'^{\mathrm{T}} \nabla_{\boldsymbol{v}} \mathcal{J} \qquad \text{(A)}$$

Adjoint Method (continued 2)

$\boldsymbol{G}$ is the composition of a number of successive steps

$$\boldsymbol{G} = \boldsymbol{G}_N \circ \ldots \circ \boldsymbol{G}_2 \circ \boldsymbol{G}_1$$

'Chain rule'

$$\boldsymbol{G}' = \boldsymbol{G}_N' \ldots \boldsymbol{G}_2' \, \boldsymbol{G}_1'$$

Transpose

$$\boldsymbol{G}'^{\mathrm{T}} = \boldsymbol{G}_1'^{\mathrm{T}} \, \boldsymbol{G}_2'^{\mathrm{T}} \ldots \boldsymbol{G}_N'^{\mathrm{T}}$$

Transpose, or *adjoint*, computations are performed in reversed order of direct computations.

If $\boldsymbol{G}$ is nonlinear, local jacobian $\boldsymbol{G}'$ depends on local value of input $\boldsymbol{u}$. Any quantity which is an argument of a nonlinear operation in the direct computation will be used gain in the adjoint computation. It must be kept in memory from the direct computation (or else be recomputed again in the course of the adjoint computation).

If everything is kept in memory, total operation count of adjoint computation is at most 4 times operation count of direct computation (in practice about 2).