Advanced assimilation methods.
Variational assimilation. Adjoint equations
Part 3

Olivier Talagrand
School *Data Assimilation*
Nordic Institute for Theoretical Physics (NORDITA)
Stockholm, Sweden
27 April 2011

Best Linear Unbiased Estimate

State vector x, belonging to *state space* $\mathcal{S}$ ($\dim\mathcal{S} = n$), to be estimated.
Available data in the form of

- A '*background*' estimate (*e. g.* forecast from the past), belonging to *state space*, with dimension n

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

- An additional set of data (*e. g.* observations), belonging to *observation space*, with dimension p

  $$y = Hx + \varepsilon$$

  H is known linear *observation operator*.

Assume probability distribution is known for the couple (ζ^b, ε).
Assume $E(\zeta^b) = 0$, $E(\varepsilon) = 0$, $E(\zeta^b \varepsilon^{\mathrm{T}}) = 0$ (not restrictive)
Set $E(\zeta^b \zeta^{b\mathrm{T}}) \equiv P^b$ (also often denoted B), $E(\varepsilon\varepsilon^{\mathrm{T}}) \equiv R$

Then

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

x^a is the *Best Linear Unbiased Estimate* (*BLUE*) of x from x^b and y.

Blue x^a minimizes *objective function* (also called *cost function*), defined on state space

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

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

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

'*3D-Var*'

used operationally in USA, Australia, China, …

Approach can easily be extended to time dimension.

Suppose for instance available data consist of

- Background estimate at time 0

  $x_0^b = x_0 + \zeta_0^b$ $\qquad E(\zeta_0^b \zeta_0^{bT}) = P_0^b$

- 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

$\xi_0 \in \mathcal{S} \rightarrow$

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

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

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

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) ?

Only practical method seems to be iterative minimization, each step of which requires the explicit knowledge of the gradient

$$\nabla_u \mathcal{J} \equiv (\partial \mathcal{J}/\partial u_i)$$

of $\mathcal{J}$ with respect to u.

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 Approach

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

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

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

$\boldsymbol{v} = (v_j)$ 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

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

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

Adjoint Approach (continued 1)

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

Scalar function of output

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

Gradient $\nabla_u \mathcal{J}$ 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

$$\nabla_u \mathcal{J} = \mathbf{G}'^{\mathrm{T}} \, \nabla_v \mathcal{J} \qquad \text{(A)}$$

Adjoint Approach (continued 2)

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

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

'Chain rule'

$$\mathbf{G}' = \mathbf{G}_K' \ldots \mathbf{G}_2' \mathbf{G}_1'$$

Transpose

$$\mathbf{G}'^{\mathrm{T}} = \mathbf{G}_1'^{\mathrm{T}} \mathbf{G}_2'^{\mathrm{T}} \ldots \mathbf{G}_K^{\mathrm{T}}$$

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

If $\boldsymbol{G}$ is nonlinear, local jacobian $\mathbf{G}'$ depends on local value of input $\boldsymbol{u}$. Any quantity that 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).

Adjoint Approach (continued 3)

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

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

Control variable $\quad \xi_0 = \boldsymbol{u}$

Adjoint equation

$$\lambda_K = \qquad H_K^T R_K^{-1}\,[H_K\,\xi_K - y_K]$$

…

…

$$\lambda_k = M_k^T\lambda_{k+1} + H_k^T R_k^{-1}\,[H_k\,\xi_k - y_k] \qquad k = K\text{-}1, \ldots, 1$$

…

…

$$\lambda_0 = M_0^T\lambda_1 \quad + H_0^T R_0^{-1}\,[H_0\,\xi_0 - y_0] \;+\; [P_0^b]^{-1}\,(\xi_0 - x_0^b)$$

$$\nabla_u \mathcal{J} \;=\; \lambda_0$$

Result of direct integration (ξ_k), which appears in quadratic terms in expression of objective function, must be kept in memory from direct integration.

Adjoint Approach (continued 4)

Nonlinearities ?

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

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

Control variable $\xi_0 = \boldsymbol{u}$

Adjoint equation

$\lambda_K = H_K'^T R_K^{-1}\,[H_K(\xi_K) - y_K]$

$\lambda_k = M_k'^T \lambda_{k+1} + H_k'^T R_k^{-1}\,[H_k(\xi_k) - y_k]$ $k = K\text{-}1, \ldots, 1$

$\lambda_0 = M_0'^T \lambda_1 + H_0'^T R_0^{-1}\,[H_0(\xi_0) - y_0] + [P_0^b]^{-1}\,(\xi_0 - x_0^b)$

$$\nabla_u \mathcal{J} = \lambda_0$$

Not heuristic (it gives the exact gradient $\nabla_u \mathcal{J}$), and really used as described here.

Adjoint Approach (continued 5)

It works (Le Dimet, Courtier *et al.*) !

‘4*D-Var*’

Used operationally at European Centre for Medium-range Weather Forecasts (ECMWF), Météo-France, Meteorological Office (UK), Canadian Meteorological Centre (together with an ensemble Kalman filter), Japan Meteorological Agency