VARIATIONAL ASSIMILATION TECHNIQUES — talk synopsis

(F. Bouttier, Météo-France, Aug 2003)

- Basic formulae
- From 3DVar to 4DVar
- Practical implementation
- Background term
- Observation term
- Adjoint operators
- Extra constraints
- More mathematics
- Code debugging

NOTATION GUIDE & BASIC HYPOTHESES

- x, x_b, x_a, x_t : model state vector, background (=prior estimate), analysis, true state
- y : vector of observed values
- H : 'observation' operator used to simulate observed value from the model state (interpolations, integrals and simulation of radiative transfer, mostly)
- B : background error covariance matrix = covariances of $(x_b - x_t)$
- A : analysis error covariance matrix = covariances of $(x_a - x_t)$
- R : observation error covariance matrix = covariances of $(y - H(x_t))$
- M : forecast model
- hypothesis 1 : no bias in $(x_b - x_t)$ and $(y - H(x_t))$
- hypothesis 2 : $(x_b - x_t)$ and $(y - H(x_t))$ are not mutually correlated
- hypothesis 3 : M and H are linearized with respect to a suitable state. Analysis will only be optimal if linearity is exact. Approximate linearization is usually acceptable if perturbations, notably $(xa - xb)$, are small.

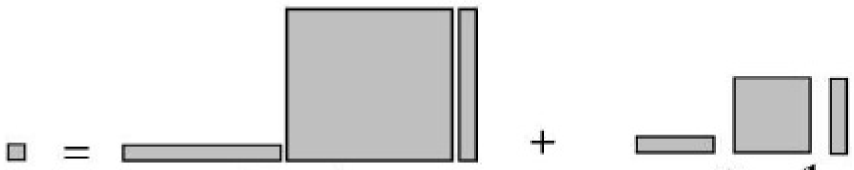
BASIC FORMULAE (1) : equivalence with BLUE

First approach : from the Best Linear Unbiased Estimator (itself derived from e.g. a statistical optimality principle) :

- if $K = BH^T(HBH^T + R)^{-1}$ or $K = (B^{-1} + H^T R^{-1} H) H^T R^{-1}$
- then the analysis $x_a = x_b + K(y - H(x_b))$ is also
- the x that minimises the *cost-function* $J(x) = (x - x_b)^T B^{-1} (x - x_b) + (y - H(x))^T R^{-1} (y - H(x))$
- (the minimum exists and is unique if B and R are positive definite)
- proof : write that the gradient of J is null for $x = x_a$.

i.e. the variational analysis is just an equivalent variational form to the problem of computing the BLUE (= solving a linear system).

Matrix shapes in 3D/4D-Var



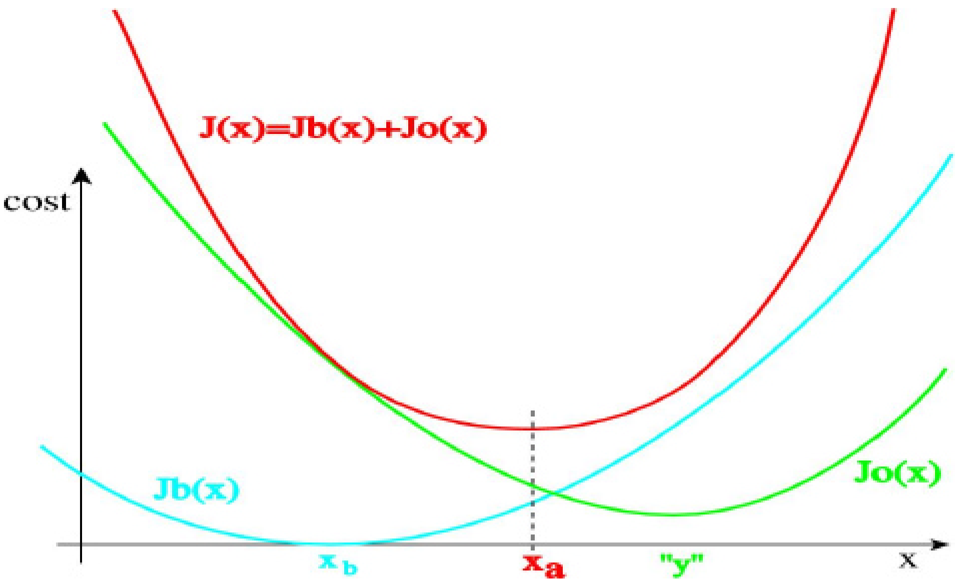
$$J(x) = (x-x_b)^T B^{-1} (x-x_b) + (y-Hx)^T R^{-1} (y-Hx)$$

BASIC FORMULAE (2) : MEANING OF THE COST-FUNCTION J

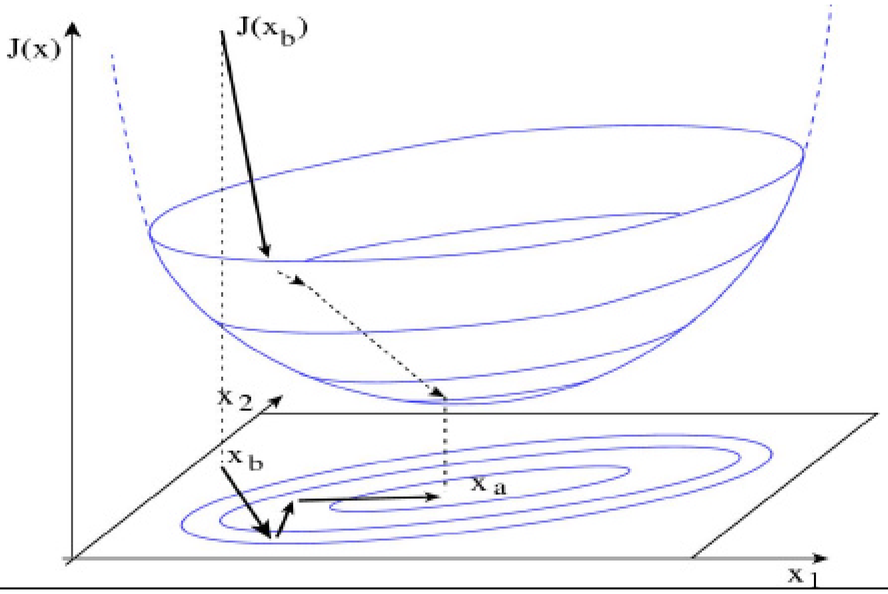
Second approach : directly in variational form

- write a quadratic J function and call its minimum the 'analysis'.
- the analysis verifies the criteria built into the definition of J i.e. to find a compromise between the background and observations.
- formally, $J(x) = J_b(x) + J_o(x)$. Additional *penalty constraints* can be added to enforce other criteria (e.g. on dynamical trends, on boundaries...)
- $J_b = (x - x_b)^T B^{-1}(x - x_b)$ is a Euclidian norm with metrics implied by B. The larger the background error variance, the weaker the background constraint.
- $J_o = (y - H(x))^T R^{-1}(y - H(x)^{-1}$ is a quadratic form (a norm if H is invertible) with metrics implied by H and R. The smaller the observation error variance, the stronger it pulls the analysis towards it.
- J is a quadratic form i.e. its isovalues are ellipsoids, it is strictly convex.
- the norm of the gradient of J is a measure of the distance to the minimum in non-dimensional units.

Jo and Jb cost functions



Multidimensional cost-function

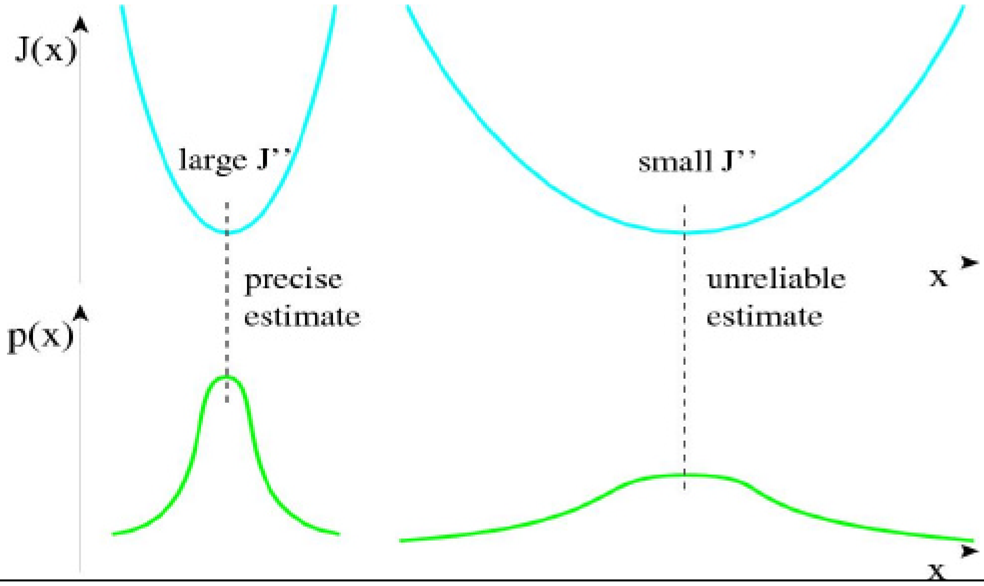


BASIC FORMULAE (3) : HESSIAN INFORMATION

The Hessian J'' is the square matrix (size $\dim x \times \dim x$) of second derivatives of the cost-function J

- $J'' = 2B^{-1} + 2H^T R^{-1} H$ is a constant.
- if B and R are the exact covariance matrices of the problem, then $J'' = A^{-1}$ where A is the analysis error covariance matrix = covariances of $(x_a - x_t)$.
- in practice, B and R are not exact so this is not true. J'' mainly reflects how B and R were built, and the distribution of the observing network.
- $J'' = A^{-1}$ implies that very dense observations will produce a nearly perfect analysis. This is not true at all in real assimilations.

Hessian information



BASIC FORMULAE (4) : THE DUAL 'PSAS' FORM

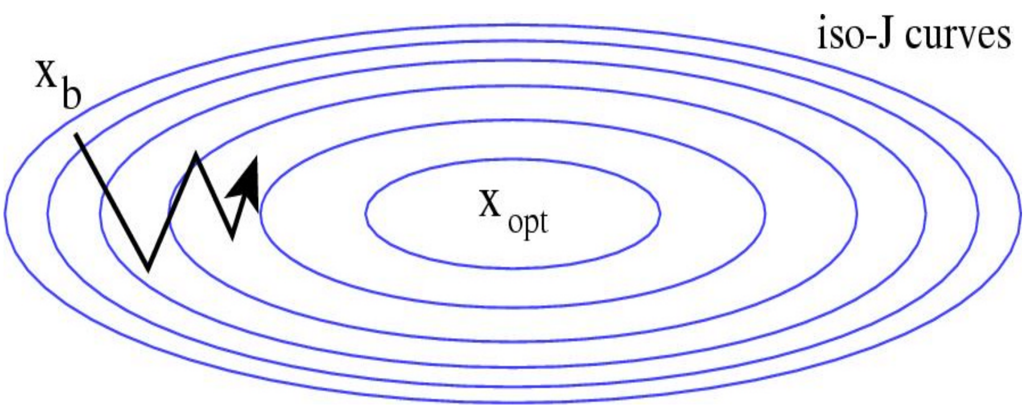
An equivalent variational form of J that gives the same result ('Physical Space Analysis System') : let w be a vector of size dim$y$ ('observation space') :

- (1) find w_a, the minimum of $P(w) = w^T(HBH^T + R)w - 2w^T(y - Hx_b)$
- (2) compute $x_a = x_b + BH^T w_a$
- the difference is that the minimization problem size is the number of observations, instead of the model state size
- the other difference is that B is used instead of its inverse (does not matter with preconditioning)
- no way to introduce an extra constraint in the space of the model state
- attractive if the observation set is much smaller than the model state (no longer true with remote-sensed data)

BASIC FORMULAE (5) : PRECONDITIONING

- The argument of the cost function, x, can be replaced by any invertible variable transform L into z by $x = Lz$.
- Minimizing $J(x)$ is equivalent to minimizing $J(L(z))$.
- Useful to improve the numerical conditioning of the minimization. The best is to use matrix J'' if known.
- the preconditioning should make the gradient of $J(L(z))$ point as much as possible toward the minimum.
- a good preconditioning is the B-preconditioning provided by $B = LL^T$ and $x - xb = Lz$. Even better ones can be computed using an iterative Lanczos method on a cheap approximation of J.
- unpreconditioned variational analysis will require thousands iterations, B-preconditioning require about 100 iterations.
- B-preconditioning means that B has to be easy to factorize.

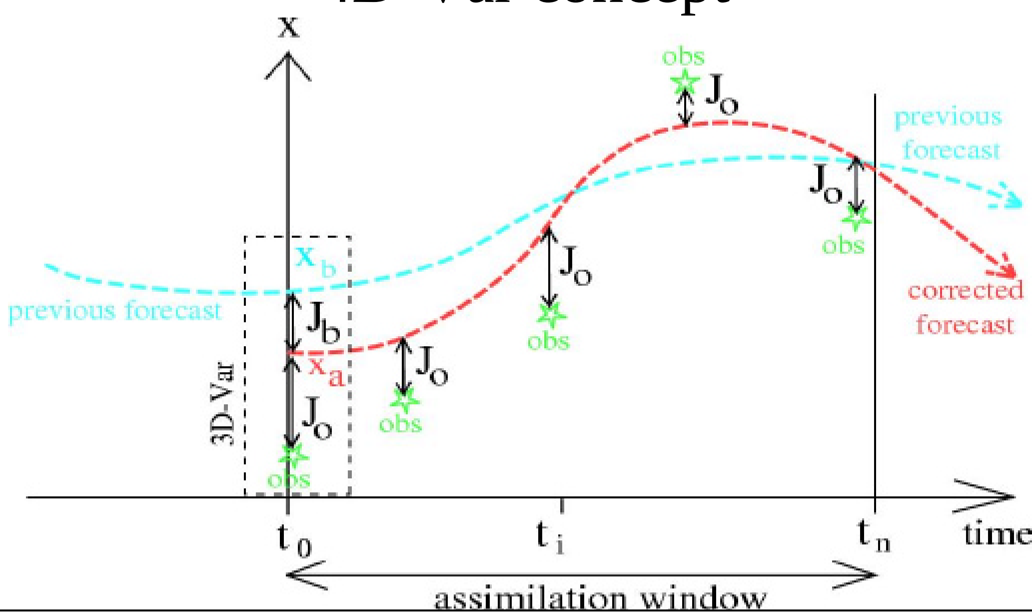
The importance of preconditioning



FROM 3DVAR TO 4DVAR

- previous equations define 3D-Var i.e. analysis with observations at a given time.
- with the hypothesis that the model M is perfect, a statistically optimal analysis with observations over a time interval is the minimum of :
- $J(x) = (x - x_b)^T B^{-1}(x - x_b) + (y - HM(x))^T R^{-1}(y - HM(x))$
- this is the same as 3DVar except H is replaced by HM = forecast from the initial time to the observation time, then observation operator.
- the control variable x and the background x_b are at the beginning of the time interval.
- the minimum x provides a statistically optimal forecast, equal to a Kalman filter analysis.

4D-Var concept



PRACTICAL IMPLEMENTATION (1) : COST-FUNCTION COMPUTATION

Modern minimization techniques use iterative computations of $J(x)$ and $J'(x)$ to 'walk down the slope' toward the minimum, x_a.

With B-preconditioning the minimization uses $J(L(z)))$ and its derivative, at the end $x_a = x_b + Lz$.

Computation of $J(x)$ (actually, $J(L(z)))$:

- $Jb(x)$ is the canonical norm of z.
- $Jo(x)$ requires computing $x = Lz + x_b$, then $y - H(x)$, then $R^{-1}(y - H(x))$, then $(y - H(x))R^{-1}(y = H(x))$.
- if the minimization starts from x_b, the first z is zero, so we never need L^{-1}.

Computation of $J'(x)$ (actually , the derivative of $J(L(z)))$:

- $J_b'(z)$ is z.
- $J_o'(z)$ is $R^{-1}(y - H(x))$ which was already computed above.

PRACTICAL IMPLEMENTATION (2) : INCREMENTAL TECHNIQUE

incremental technique : replace the original minimization by an approximation that is valid for small increments of x.

Applications :

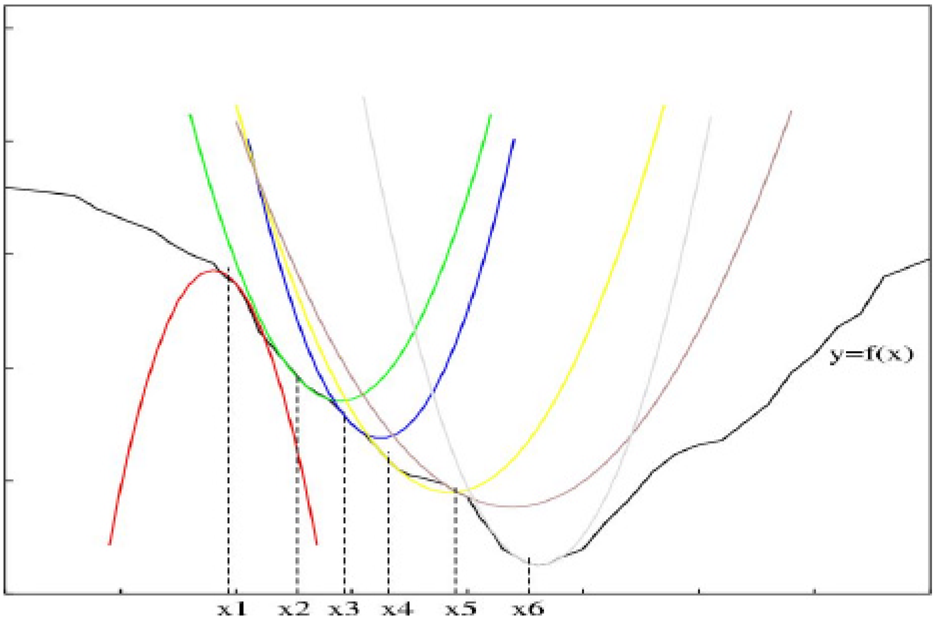
- non-linear H in $J_o = (y - H(x))R^{-1}(y - H(x)^{-1}$ implies a non-quadratic function, perhaps with a non-unique minimum. Numerically easier and more efficient by linearizing H (perhaps several times as x gets closer to x_a)
- 3D/4D-Var can be numerically costly. Reduce the size of the workspace by assuming $x - x_b$ is nonzero only for certain parameters and scales (ok if non-analysed features are forced by the analysed ones)

Mathematically :

- assume $H(x) = H(x_{ref}) + \hat{H}(x - x_{ref})$ e.g. Taylor formula
- minimize in terms of $x - x_{ref}$
- at the beginning, $x_{ref} = x_b$
- at the end, define injection C such that $xa \simeq xref + C(x - xref)$

Problem : may not converge towards the real x_a.

Quadratic approximations to cost-function



OBSERVATION TERM

Observation errors from independent sensors are usually uncorrelated.

Group observations into uncorrelated batches $y = yi$ so that R is made of blocks R_i : then, R^{-1} is made of blocks and $J_o = \sum_i (y_i - H_i(x))R_i^{-1}(y_i - H_i(x)) = \sum_i J_{oi}$

Monitoring the J_{oi} terms and their gradients is useful to diagnose the influence of each class of observations : per type, per area, per time of day, etc.

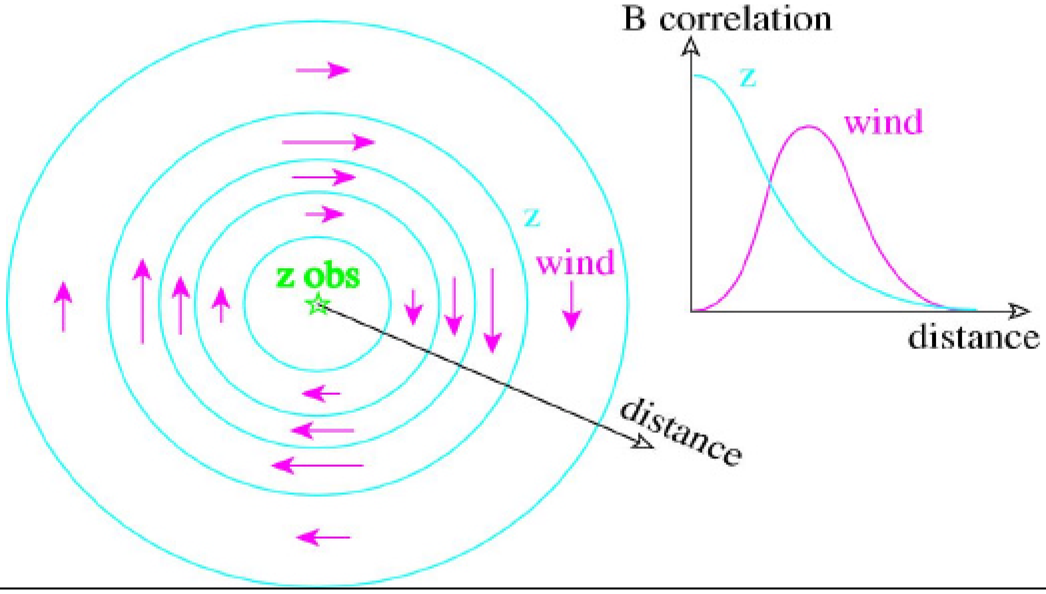
BACKGROUND TERM

B determines the shape of the influence of observations in space, and the multivariate coupling (modified by the model in 4D-var).

Essential B ingredients :

- positive definite (otherwise the cost-function may not be minimized)
- diagonal = scaling of the background errors in model space. Defines the areas where observations will have an impact.
- smoothing effect in all directions in space, usually product of (more or less homogeneous isotropic) horizontal smoother (spectral or gridpoint filter) and vertical correlation operator.
- multivariate cross-correlations, usually defined as a 'balance' e.g. $b = B(a) + b_u$ with b_u assumed much smaller than $b_b = B(a)$.
- most likely error structures that have grown in the previous forecast e.g. $B = B_s + \sum_i z_i z_i^T$ with B_s 'static' matrix and z_i situation-dependent error structures (e.g. Ensemble Kalman Filter).
- these operator are mixed cheaply using decorrelation assumptions = tensor products of small operators.

Structure functions



ADJOINT OPERATORS

Transpose matrices are rarely written explicitly, they are coded as operators that verify $< Ax, z > = < x, A^* z >$

- If the scalar product is the inner dot product, then $A^* = A^T$
- otherwise, the scalar product metric must be used (e.g. L2 norm to compute averages, or Parseval identity on FFTs).
- Coding rule 1 : reverse order of computations, $(ABCD \cdots)^T = (\cdots D^T C^T B^T A^T)$
- Coding rule 2 : the adjoint of an operation on a few variables only acts on those few variables.
- Coding rule 3 : the adjoint of the disappearance of a variable (when it is last used) is the setting of this variable to zero (adjoint of projector)
- Coding rule 4 : the adjoint always acts in an additive way, $x1 = a.x2$ become $x2 = x2 + a.x1$, except $x1 = a.x1$ becomes itself.
- Coding rule 5 : code and validate the adjoint locally, do not try to adjoint big chunks of code.

Automated adjoint code compilers are commercially available, but may not be optimized enough for large models.

EXTRA CONSTRAINTS

Penalty : constraint added to a cost-function in order to give certain properties to the analysis. Examples :

- reduce tendencies of gravity waves, $J_c = \alpha \parallel \partial G/\partial \parallel^2$
- prevent fast transients, $J_c = \alpha \parallel DFI(x_{t1} \cdots x_{tn}) \parallel^2$
- discourage the increment from crossing certain values
- etc...

These properties should ideally be built into B, but we often do not know how to formulate them as a covariance matrix. J_c will sually degrade the B-conditioning an slow down minimization.

(A BIT) MORE MATHEMATICS

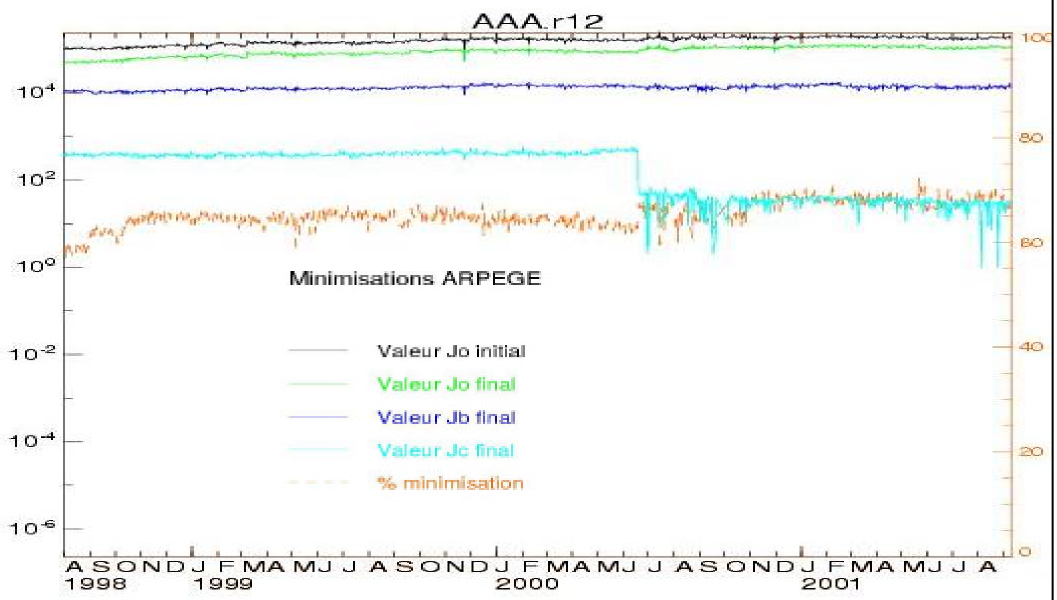
- A quadratic cost function will normally be minimized using a conjugate-gradient algorithm.
- The minimization involves **simulations** = evaluations of the cost-function and its gradient, and **iterations** = corrections of the control variable x
- The minimization involves estimating eigenvectors of the Hessian, J''.
- The Hessian estimate is a by-product of the minimization and may be used to precondition a subsequent 3D/4DVar if the shape of the cost function (obs distribution and model dynamics) does not change much.
- Fields of effective background B and A variances may be estimated cheaply using randomization in the model or observation space : shows how effectively each obs is used.
- In a statistically well-tuned system, the $Jo(xa)$ expectation is the number of obs i.e. 1 for each independent obs : shows if some obs are too much used or not enough.

CODE DEBUGGING

Variational sanity checks :

- test adjoint identity on all adjoint codes (for B, R and M)
- test Taylor identity for gradients of J_o and J_b
- check J_o and J_b are quadratic
- check the cost function is reduced by the minimization
- check the norm of the gradient of J decreases (approximately exponentially) during the minimization, of at least two orders of magnitude
- check output condition of minimizer (number of iterations, simulations, decrease of gradient)
- check Jo is decreased = we get closer to the observations (for each observation types, on large enough samples)
- check adding more obs does not degrade the fit to the other obs
- run one-obs experiments to check structure functions implied by J_b

monitoring the cost-function



REFERENCES

Andersson, E., J. Pailleux, J.-N. Thépaut, J. Eyre, A. McNally, G. Kelly and P. Courtier, 1994 : Use of cloud-cleared radiances in three/four-dimensional variational data assimilation. *Quart. Jour. Roy. Met. Soc.*, **120,** 627-653.

Bouttier, F. and Courtier, 1997 : Data assimilation concepts and methods. *ECMWF training course lecture notes.* Available online at www.ecmwf.int

Derber, J., 1987 : variational four-dimensional analysis using quasi-geostrophic constraints. **Mon. Wea. Rev., 123,** 1128–1145.

Talagrand, O. and P. Courtier, 1987 : Variational assimilation of meteorological observations with the adjoint vorticity equation. I : Theory. *Quart. J. Roy. Meteor. Soc.*, **113,** 1311-1328.