The Abdus Salam
International Centre
for Theoretical Physics

The International Union of Geodesy and Geophysics

2373-6

Workshop on Geophysical Data Analysis and Assimilation

29 October - 3 November, 2012

Data Assimilation (2)

Rodolfo Guzzi
Italian Space Agency
Italy

Data Assimilation

Rodolfo Guzzi
Italian Space Agency

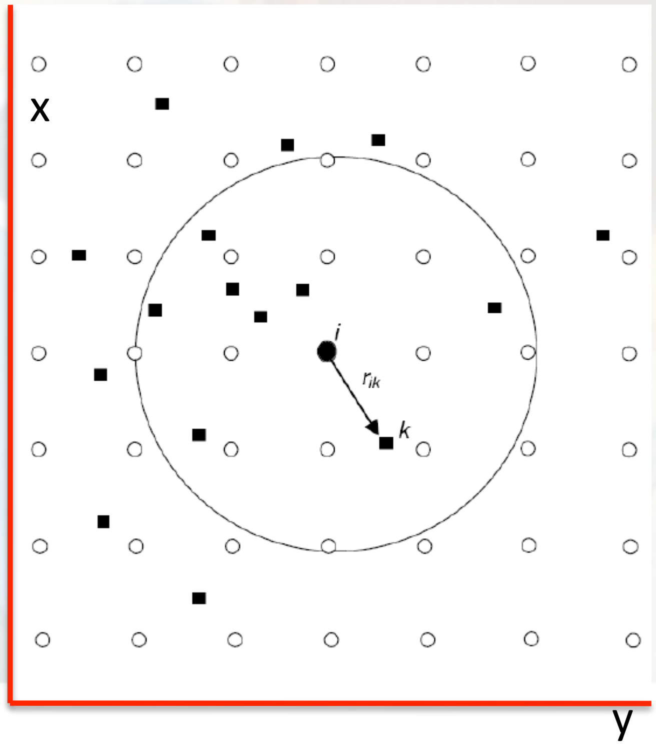
DA Definition

- Data Assimilation is a anlysis technique that helps to integrate observed information to our model state by taking advantage of the consistency of constraints with respect to laws of time evolution and physical properties
- There are two approach:

1. Continuous assimilation where information are handled in long batches (more realistic)
2. Intermittent assimilation where information are processes in small batches (more convenient)

Primitive scheme

- One of the first is the local polynomial, assuming that in our investigation domain data are irregularly distributed (squares).
- In order to have an input for numerical models we interpolate these observations to grid points (circles) with coordinate (x,y)



Primitive scheme

- A quadratic polynomial in x and y is defined as:

$$z(x,y) = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2$$

- We need to find the coefficients a_{ij} which matches observation to each grid points
- Should be perfect if we could integrate data from all station but:

1. Calculation explode
2. Data excessely distant from grid points are profitless

Primitive scheme

- The solution is to define a *radius of influence* where z(x,y) mantain a proper precision.
- Coefficients a_{ij} *are* determined by

$$J = \sum_{k=1}^{K_z} p_k [z(x_k, y_k) - z_k^{ob}]^2 \xrightarrow{a_{ij}} min$$

Local observation

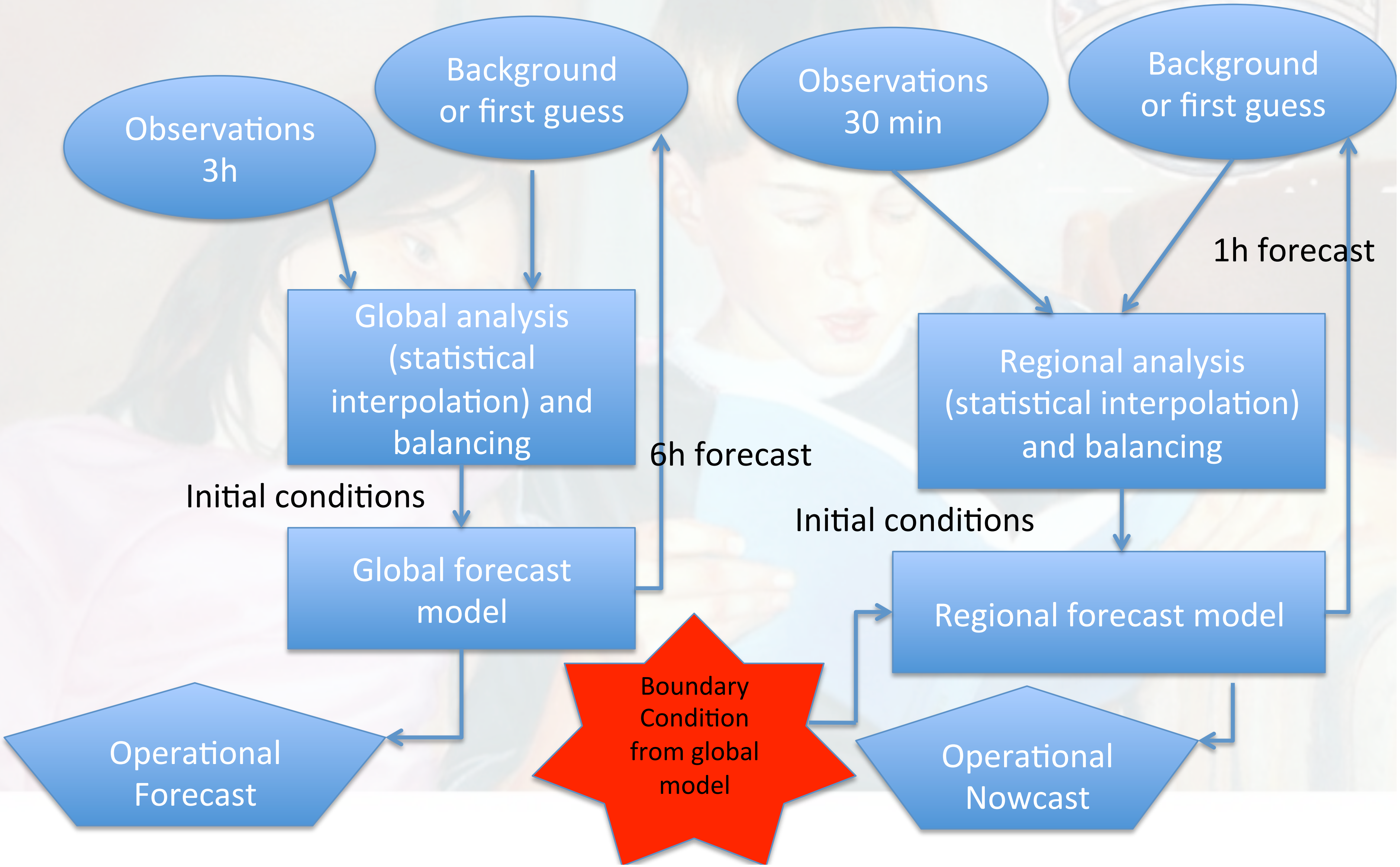
Empirical weighting coefficients

Issues of DA models

1. When we do not have any observation within the radius of influence no inizialization can be done.
2. Not all data can be directly assimilated (remote sensing data for instance)
3. Observations are not uniform in space and time

- As a result we need to merge into the model a first guess estimate at grid point that are the initial condition for *nowcast* and *forecast*.
- Such information is called *prior information* or *background information*

Forecast vs Nowcast



DA State vector, observations and errors

- The set of numbers to represent the state of of model is collected as a column matrix named *state vector* x.
- The *true state vector* describing the real state of the physical environment is denoted as x_t
- The *background state vector* is x_b which represents the true state before the analysis
- The *analysis state vector* x_a is the target of our study

continue

- In seeking the solution for model state analysis it is impossible to solve all the components of the model due to:

1. Our limited capability to model all components
2. Insufficient computational power to perform the analysis

- Instead of striving the true analysis we may restrict our interest to find the correction of background vector so that the analysis vector come closest to the true vector
- In case where our objective space is not the analyisis but the correction, they are the so called *control space*, we have:

$$x_a = x_b + \delta x \longrightarrow x_t$$

correction

continue

- The observed value or observation vector is y
- The observation operator is H
- The key of data assimilation is to recognize the difference between observation and state vector

$$y=H(x)$$

- The observation operator can be linear or not linear
- In this last case the general approch is to linearize it by a first order Taylor expansion. Other approaches will be seen in successive slides.

Errors and their covariances

- Error is in form of differences bewteen true state vector and other state vectors.
- In modeling tese errors the most powerfull means to represent the uncertaintiy is to use the *probability density function* (pdf).
- This is because we know exactly what error occur in each individual case but we can own its statistics and its average

continue

- We can define:

	errors	matrix	trace	
background	$\epsilon_b = x_b - x_t$	$B = \overline{(\epsilon_b - \bar{\epsilon}_b)(\epsilon_b - \bar{\epsilon}_b)^T}$		
observation	$\epsilon_{ob} = y - H(x_t)$	$R = \overline{(\epsilon_{ob} - \bar{\epsilon}_{ob})(\epsilon_{ob} - \bar{\epsilon}_{ob})^T}$		They consist of instrument errors
analysis	$\epsilon_a = x_a - x_t$	$A = \overline{(\epsilon_a - \bar{\epsilon}_a)(\epsilon_a - \bar{\epsilon}_a)^T}$	$Tr(A) = \overline{\|(\epsilon_a - \bar{\epsilon}_a)\|^2}$	

The average of errors are called biases

Optimal Least Square Method

- In the 1° toy model we have learned the least square method of a scalar at a fixed point
- Now we discus how to find optimal analysis for several variables. Such analysis is called Optimal Least Square Estimator or BLUE (Best Linear Unbiased Estimation) analysis.
- At beginning we use the linear assumption of observation operator with $h=x-x_b$

$$y - H(x) \approx y - H(x_b) - H(x - x_b)$$

continue

- The departure of true state vector x_t after swapping positions of $H(x_t)$ and $H(x_b)$ is:

$$y - H(x_b) = y - H(x_t) - H(x_t - x_b)$$

- Apply the definition of errors we have:

$$y - H(x_b) = \epsilon_{ob} - H\epsilon_b$$

- Assume now the analysis x_a is calculated by the background x_b and the observation departure through the linear equation

$$x_a = x_b + K[y - H(x_b)]$$

Unknown

continue

- Instead seeking their state vectors the equivalence is to find their errors:
- Adding x_t into the previous equation:

$$x_a - x_t = x_b - x_t + K[y - H(x_b)]$$

to have:

$$\epsilon_a = \epsilon_b - K[\epsilon_{ob} - H\epsilon_b] = (I - KH)\epsilon_b + K\epsilon_{ob}$$

where I is the Identity Matrix

continue

- Remembering the Analysis Errors Covariance A

$$A = \overline{(\epsilon_a - \bar{\epsilon}_a)(\epsilon_a - \bar{\epsilon}_a)^T}$$

we obtain:

$$A = (I - KH)B(I - KH)^T + KRK^T$$

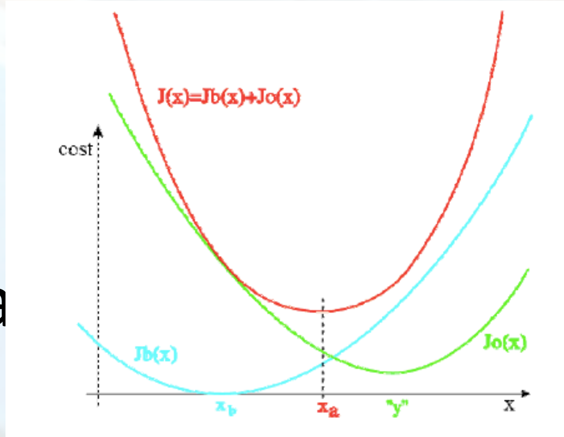
- To obtain K $\quad \frac{dA}{dK} = 0$

$$K = BH^T(HBH^T + R)^{-1}$$

That is called Weight Matrix or Gain.

Variational Method and 3D-Var

- An alternative approach to obtain the optimal solution is to use the variational method also named Cost Function approach.
- The idea is to obtain the minimum of the misfit of the estimate with respect to the observations.
- We could apply the VM in general form by introducing a cost function J



continue

- A specific variational assimilation problem is that finds the optimal analysis x_a field that minimizes a (scalar) cost function.
- The cost function is defined as the (weighted) distance between x and the background x_b, plus the (weighted) distance to the the observations y,
- **The Cost Function, J, is the link between the observational data and the model variables**

$$J = \frac{1}{2}\left\{(x - x_b)^T B^{-1}(x - x_b) + [y - H(x)]^T R^{-1}[y - H(x)]\right\}$$

Observations are either assumed unbiased, or are "debiased" by some adjustment method

Bayes Theorem

Maximum Conditional Probability is given by:

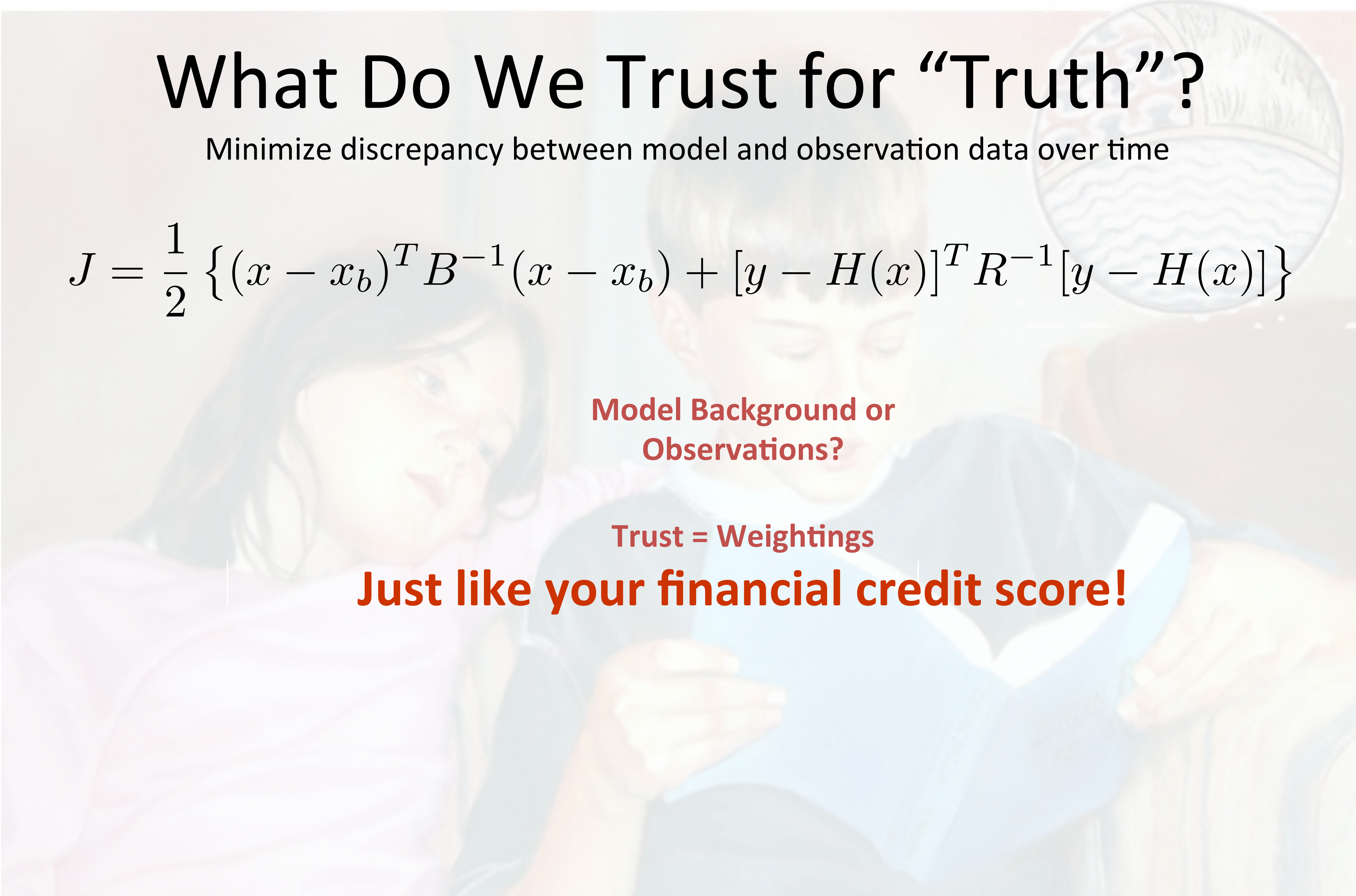
$P(x \mid y) \sim P(y \mid x)\, P(x)$

Assuming Gaussian distributions…

$P(y \mid x) \sim \exp\{-1/2\, [y - H(x)]^T R^{-1} [y - H(x)]\}$
$P(x) \sim \exp\{-1/2\, [x - x_b]^T B^{-1} [x - x_b]\}$

} e.g., 3DVAR

Lorenc (1986)



What Do We Trust for "Truth"?

Minimize discrepancy between model and observation data over time

$$J = \frac{1}{2}\left\{(x - x_b)^T B^{-1}(x - x_b) + [y - H(x)]^T R^{-1}[y - H(x)]\right\}$$

Model Background or Observations?

Trust = Weightings

Just like your financial credit score!

Who are the Candidates for "Truth"?

Minimize discrepancy between model and observation data over time

$$J = \frac{1}{2}\left\{\underline{(x - x_b)^T B^{-1}(x - x_b)} + [y - H(x)]^T R^{-1}[y - H(x)]\right\}$$

Candidate 1: Background Term

"x" **is the model state vector at the initial time t_0**

this is also the "control variable",

the object of the minimization process

"x_b" **is the model background state vector**

"B" **is the background error covariance**

of the forecast and model errors

Who are the Candidates for "Truth"?

Minimize discrepancy between model and observation data over time

$$J = \frac{1}{2}\left\{(x - x_b)^T B^{-1}(x - x_b) + [y - H(x)]^T R^{-1}[y - H(x)]\right\}$$

Candidate 2: Observational Term

"y" **is the observational vector, e.g., the satellite input data (typically radiances), salinity, sounding profiles**

" (x)" is the model state at the observation time "i"

"h" **is the observational operator, for example the "forward radiative transfer model"**

"R" **is the observational error covariance matrix that specifies the instrumental noise and data representation errors (currently assumed to be diagonal…)**

What Do We Trust for "Truth"?

Minimize discrepancy between model and observation data over time

$$J = \frac{1}{2}\left\{\underline{(x - x_b)^T B^{-1}(x - x_b)} + [y - H(x)]^T R^{-1}[y - H(x)]\right\}$$

Candidate 1: Background Term

The default condition for the assimilation when

1. **data are not available or**
2. **the available data have no significant sensitivity to the model state or**
3. **the available data are inaccurate**

Model Error Impacts our "Trust"

Minimize discrepancy between model and observation data over time

$$J = \frac{1}{2}\left\{\underline{(x - x_b)^T B^{-1} (x - x_b)} + [y - H(x)]^T R^{-1} [y - H(x)]\right\}$$

Candidate 1: Background Term

Model error issues are important

Model error varies as a function of the model time

Model error "grows" with time

Therefore the background term should be trusted more at the initial stages of the model run and trusted less at the end of the model run

How to Adjust for Model Error?

Minimize discrepancy between model and observation data over time

$$J = \frac{1}{2}\left\{\underline{(x - x_b)^T B^{-1} (x - x_b)} + [y - H(x)]^T R^{-1} [y - H(x)]\right\}$$

Candidate 1: Background Term

1. **Add a model error term to the cost function so that the weight at that specific model step is appropriately weighted or**
2. **Use other possible adjustments in the methodology, i.e., "make an assumption" about the model error impacts**

If model error adjustments or controls are used the DA system is said to be "weakly constrained"

What About Model Error Errors?

Minimize discrepancy between model and observation data over time

$$J = \frac{1}{2}\left\{\underline{(x - x_b)^T B^{-1}(x - x_b)} + [y - H(x)]^T R^{-1}[y - H(x)]\right\}$$

Candidate 1: Background Term

Model error adjustments to the weighting can be "wrong"

- **In particular, most assume some type of linearity**
- **Non-linear physical processes may break these assumptions and be more complexly interrelated**

A data assimilation system with no model error control is said to be "strongly constrained" (perfect model assumption)

What About other DA Errors?

Overlooked Issues?

1. **Data debiasing relative to the DA system "reference".**
   It is not <u>the</u> "Truth",
   however it is self-consistent.
2. **DA Methodology Errors?** **Synoptic vs. Mesoscale?**
   1. **Assumptions: Linearization, Gaussianity, Model errors**
   2. **Representation errors (space and time)**
   3. **Poorly known background error covariances**
   4. **Imperfect observational operators**
   5. **Overly aggressive data "quality control"**
   6. **Historical emphasis on dynamical impact vs. physical**

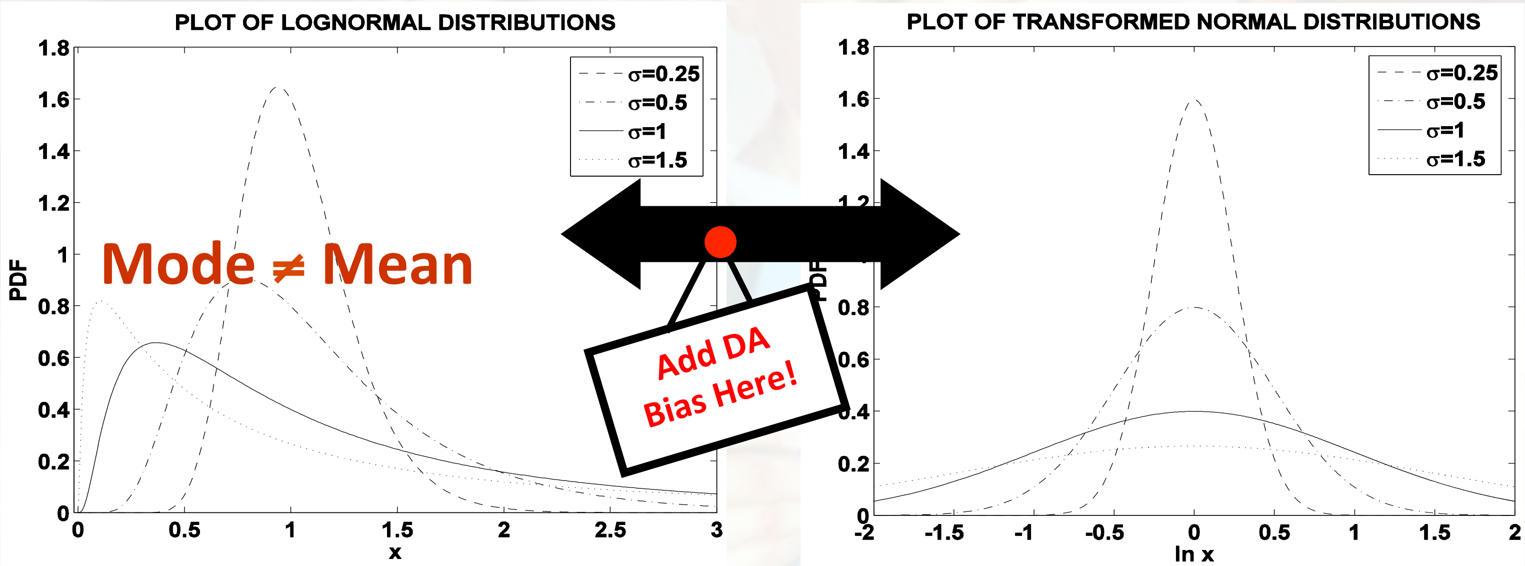
DA Theory is Still Maturing

The Future: Lognormal DA (Fletcher and Zupanski, 2006, 2007)

Gaussian systems typically force lognormal variables to become Gaussian introducing an ***avoidable*** data assimilation system bias

Lognormal Variables
Clouds
Precipitation
Water vapor
Emissivities

Many other hydrologic fields



Many important variables are lognormally distributed

Gaussian data assimilation system variables are “Gaussian”

What Do We Trust for "Truth"?

Minimize discrepancy between model and observation data over time

$$J = \frac{1}{2}\left\{(x - x_b)^T B^{-1}(x - x_b) + [y - H(x)]^T R^{-1}[y - H(x)]\right\}$$

Candidate 2: Observational Term

The non-default condition for the assimilation when

1. **data are available and**
2. **data are sensitive to the model state and**
3. **data are precise (not necessarily "accurate") and**
4. **data are not thrown away by DA "quality control" methods**

What "Truth" Do We Have?

Minimize discrepancy between model and observation data over time

$$J = \frac{1}{2}\left\{(x - x_b)^T B^{-1}(x - x_b) + [y - H(x)]^T R^{-1}[y - H(x)]\right\}$$

MODEL CENTRIC

DATA CENTRIC

TRUTH

3D-Var

- The minimum of J(x) is attained for $x=x_a$ such that:

$$\frac{\partial J}{\partial x} = \nabla_x J(x_a) = 0 \qquad (n \times 1)$$

- Assuming the analysis is closed to the truth we write:

$$x = [x_b + (x - x_b)]$$

- Assuming $x-x_b$ is small we can linearize the observation operator

$$[y - H(x)] = y - H[x_b + (x - x_b)] = \{y - H(x_b)\} - H(x - x_b)$$

continue

- Substitute it into the Cost Function we obtain:

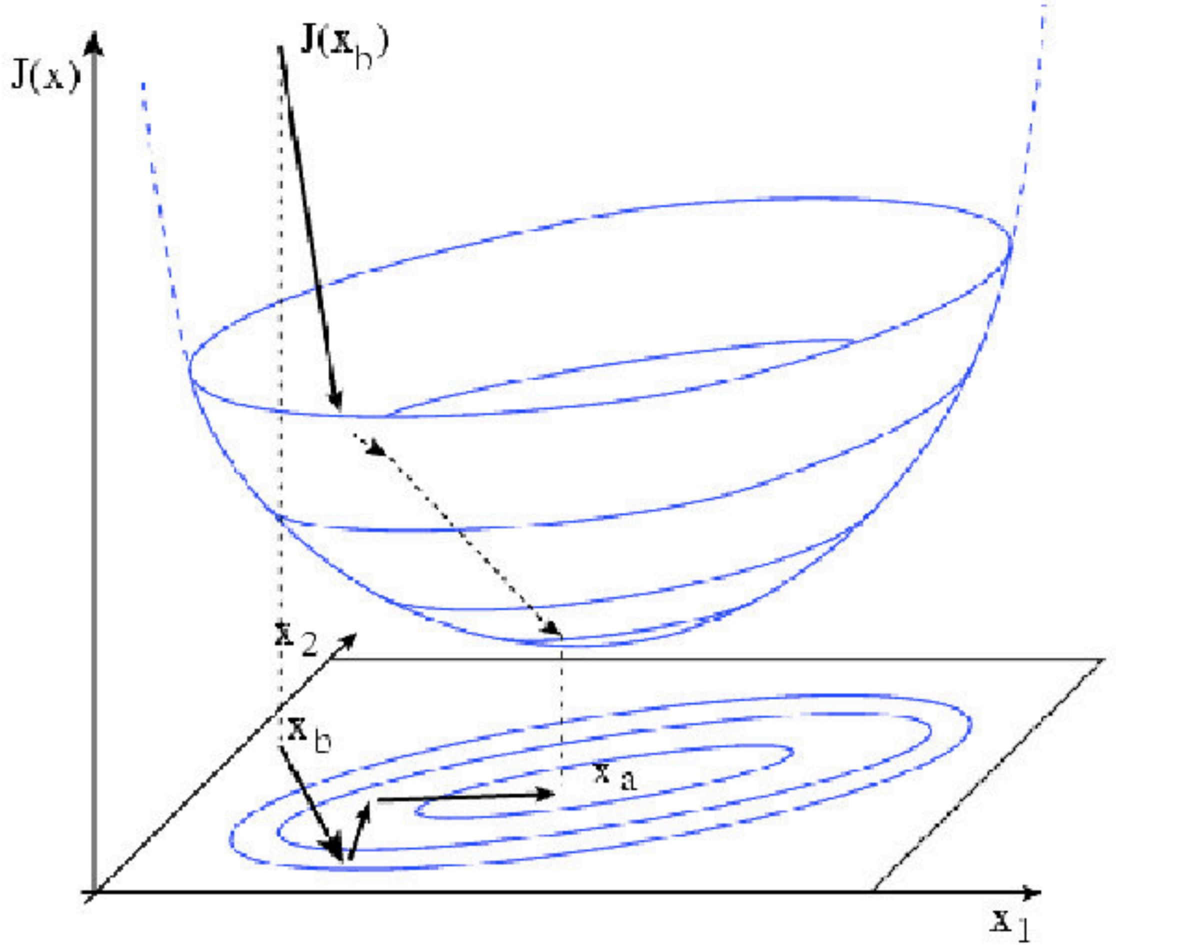
$$2J(x) = (x - x_b)^T B^{-1} (x - x_b)$$
$$+ [\{y - H(x_b)\} - H(x - x_b)]^T R^{-1} [\{y - H(x_b)\} - H(x - x_b)]$$

- Expanding the products we get:

$$2J(x) = (x - x_b)^T B^{-1} (x - x_b) + (x - x_b)^T H^T R^{-1} H (x - x_b)$$
$$- \{y - H(x_b)\}^T R^{-1} H (x - x_b)$$
$$- (x - x_b)^T H^T R^{-1} \{y - H(x_b)\}$$
$$+ \{y - H(x_b)\}^T R^{-1} \{y - H(x_b)\}$$

- The cost function is a quadratic function of the analysis increment $x\text{-}x_b$

continue



The two dimensions Cost Function.

The minimum is found by moving down-gradient in discrete steps.

continue

- Recall the cost function and combine the first two terms we get:

$$2J(x) = (x - x_b)^T [B^{-1} + H^T R^{-1} H]$$
$$- \{y - H(x_b)\}^T R^{-1} H (x - x_b)$$
$$- (x - x_b)^T H^T R^{-1} \{y - H(x_b)\}$$
$$+ \{y - H(x_b)\}^T R^{-1} \{y - H(x_b)\}$$

- The gradient of J respect x is:

$$\nabla J(x) = [B^{-1} + H^T R^{-1} H](x - x_b) - H^T R^{-1} \{y - H(x_b)\}$$

continue

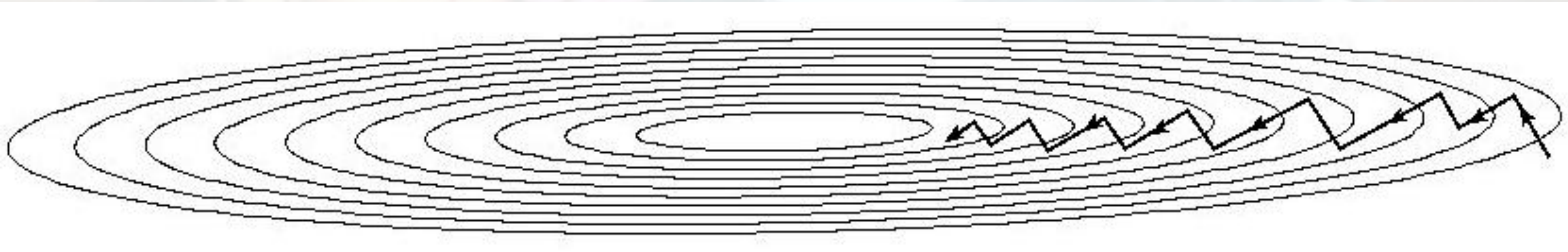
- Setting the gradient $\nabla J(x)\,|_{x_a} = 0$ we obtain:

$$x_a = x_b + [B^{-1} + H^T R^{-1} H]^{-1} H^T R^{-1} \{y - H(x_b)\}$$

- This is the solution of the 3 Dimensional Variational (3D-Var) analysis problem
- This is the formal solution
- In pratical 3D-Var we do not invert a huge matrix

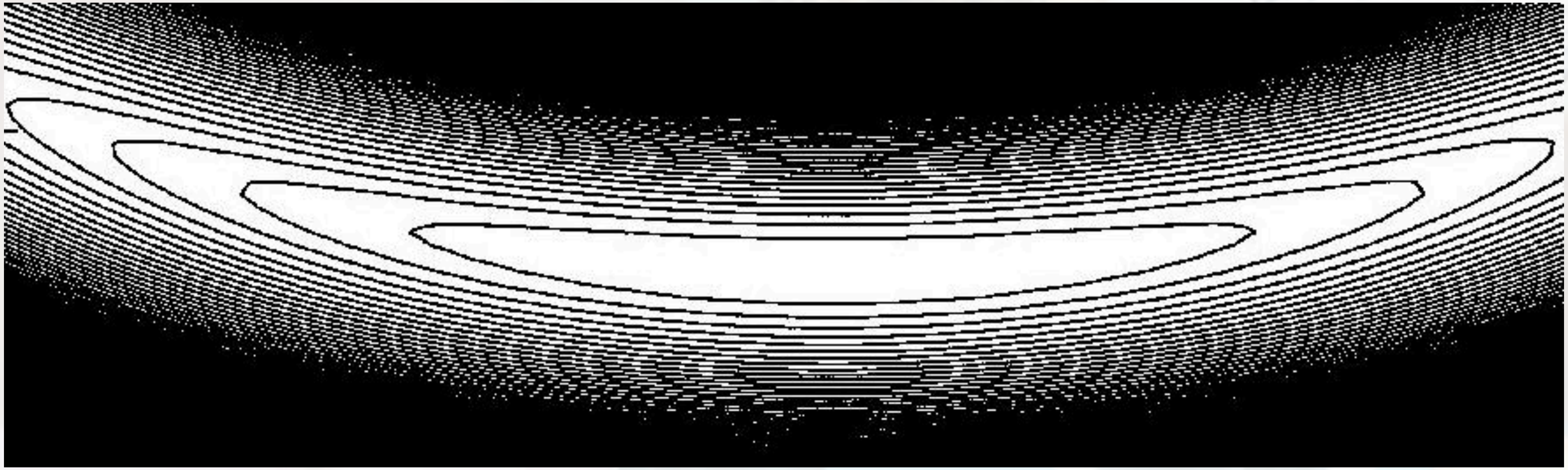
continue

- The idea is to proceed downhill as quickly possible: example are Steepest Descend algorithms, Newton' method, Levenberg Marquardt method, etc
- The location of the minimum depends on the nature of J function
- As an example we consider the shape of the surface J=J(x,y)



- For a purely elliptic surface the minimum is easily located

continue



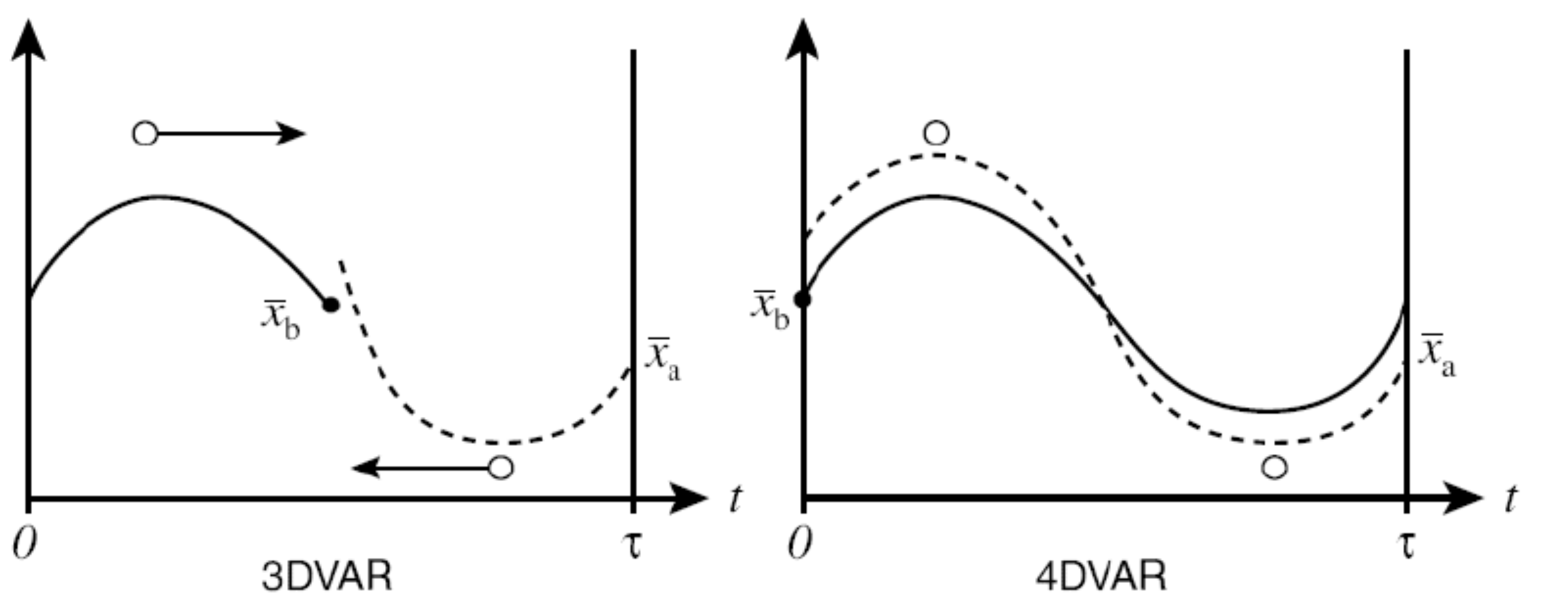
- For a banana shaped surface the minimum is much harder to find.

4D-Var

- 4D-var is generalizion of 3D-Var including observation at different times

$$J(x(t_0)) = (x_0(t_0) - x_b(t_0))^T B_0^{-1}(x(t_0) - x_b(t_0))$$

$$+ \sum_{i=0}^{n} [y_i^0 - H_i(x_i)]^T R_i^{-1} [y_i^0 - H_i(x_i)]$$



continue

- Need to define $\nabla J(x(t_0))$ in oder to minimize $J(x(t_0))$

Separate $J(x(t_0))$ into "background" and "observation" terms

$$J = J_b + J_o, \quad \frac{\partial J}{\partial \mathrm{x}(t_0)} = \frac{\partial J_b}{\partial \mathrm{x}(t_0)} + \frac{\partial J_o}{\partial \mathrm{x}(t_0)}$$

First, let's consider $J_b(\mathbf{x}(t_0))$

Given a symmetric matrix $\mathbf{A}$, and

a function $J = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}$, the gradient is given by $\frac{\partial J}{\partial \mathbf{x}} = \mathbf{A}\mathbf{y}$

continue

$$J_b = \frac{1}{2}[x(t_0) - x_b(t_0)]^T B^{-1}[x(t_0) - x_b(t_0)]$$

$$\frac{\partial J_b}{\partial x(t_0)} = B^{-1}[x(t_0) - x_b(t_0)]$$

∇J_o is more complicated, because it involves the integration of the model:

$$J_o = \frac{1}{2}\sum_{i=0}^{N}[H(x_i) - y_i^o]R_i^{-1}[H(x_i) - y_i^o]$$

If $J = \mathbf{y}^T\mathbf{A}\mathbf{y}$ and $\mathbf{y} = \mathbf{y}(\mathbf{x})$, then $\frac{\partial J}{\partial \mathbf{x}} = \left[\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right]^T \mathbf{A}\mathbf{x}$,

where $\left[\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right]_{k,l} = \frac{\partial y_k}{\partial x_l}$ is a matrix.

$\mathbf{x}_i = M_i[\mathbf{x}(t_0)]$

$$\frac{\partial(H(\mathbf{x}_i) - y_i^o)}{\partial \mathbf{x}_0} = \frac{\partial H}{\partial \mathbf{x}_i}\frac{\partial M_i}{\partial \mathbf{x}_0} = \mathbf{H}_i\mathbf{L}(t_0, t_i) = \mathbf{H}_i\mathbf{L}_{i\text{-}1}\mathbf{L}_{i\text{-}2}\cdots\mathbf{L}_0$$

$$[\mathbf{H_i L_{i\text{-}1} L_{i\text{-}2} \ldots L_0}]^T = \mathbf{L}_0^T \cdots \mathbf{L}_{i-2}^T\mathbf{L}_{i-1}^T\mathbf{H}_i^T = \mathbf{L}^T(t_i, t_0)\mathbf{H}_i^T$$

$$\left[\frac{\partial J_o}{\partial \mathbf{x}(t_0)}\right] = \sum_{i=0}^{N} L^T(t_0, t_i) \boxed{H_i^T R_i^{-1}[H(x_i) - y_i^o]}$$

Adjoint model integrates increment backwards to t_0

weighted increment at observation time, t_i, in model coordinates

4D-Var

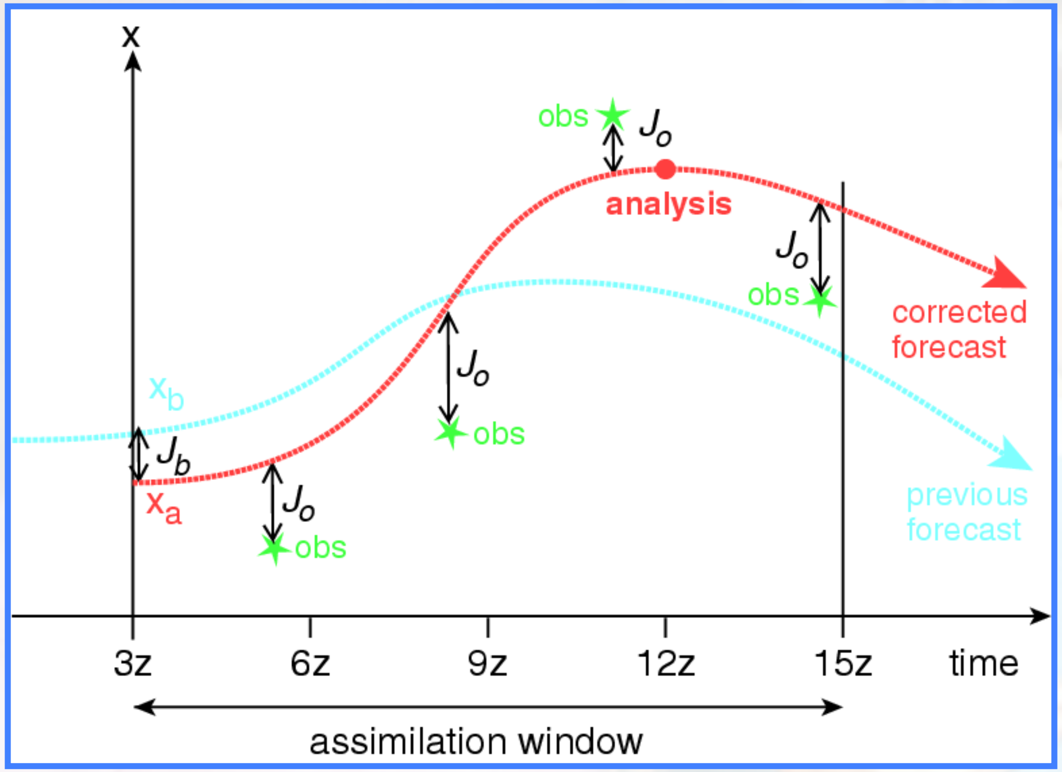
- The 4D-Var problem constraints sequence of model state to be a solution of model equations

$$x_i = M_i[x(t_0)]$$

where $M_i[x(t_0)]$ is the Predefined Model Forecast Operator .

- Then 4D-Var becomes a non linear optimization problem which is hard to solve

3D-Var vs. 4D-Var



- 4D-Var finds the 12-hour forecast evolution that best fits the available observations
- It does so by adjusting 1) surface pressure, and the upper-air fields of 2) temperature, 3) wind, 4) specific humidity and 5) ozone

1. 4D-Var assumes a perfect model. It will give the same credence to older bservations as to newer observations.
2. Background error covariance is time-independent in 3D-Var, but evolves implicitly in 4D-Var.
3. In 4D-Var, the adjoint model is required to compute ∇J

Practical implementation: use the incremental form

$$J(\delta\mathbf{x}_0) = \frac{1}{2}(\delta\mathbf{x}_0)^T \mathbf{B}_0^{-1}\delta\mathbf{x}_0 + \frac{1}{2}\sum_{i=0}^{N}[H_i\mathbf{L}(t_0,t_i)\delta\mathbf{x}_0 - \mathbf{d}_i^o]^T \mathbf{R}^{-1}[H_i\mathbf{L}(t_0,t_i)\delta\mathbf{x}_0 - \mathbf{d}_i^o]$$

where $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}_b$ and $\mathbf{d} = \mathbf{y}_\mathbf{o} - H(\mathbf{x})$

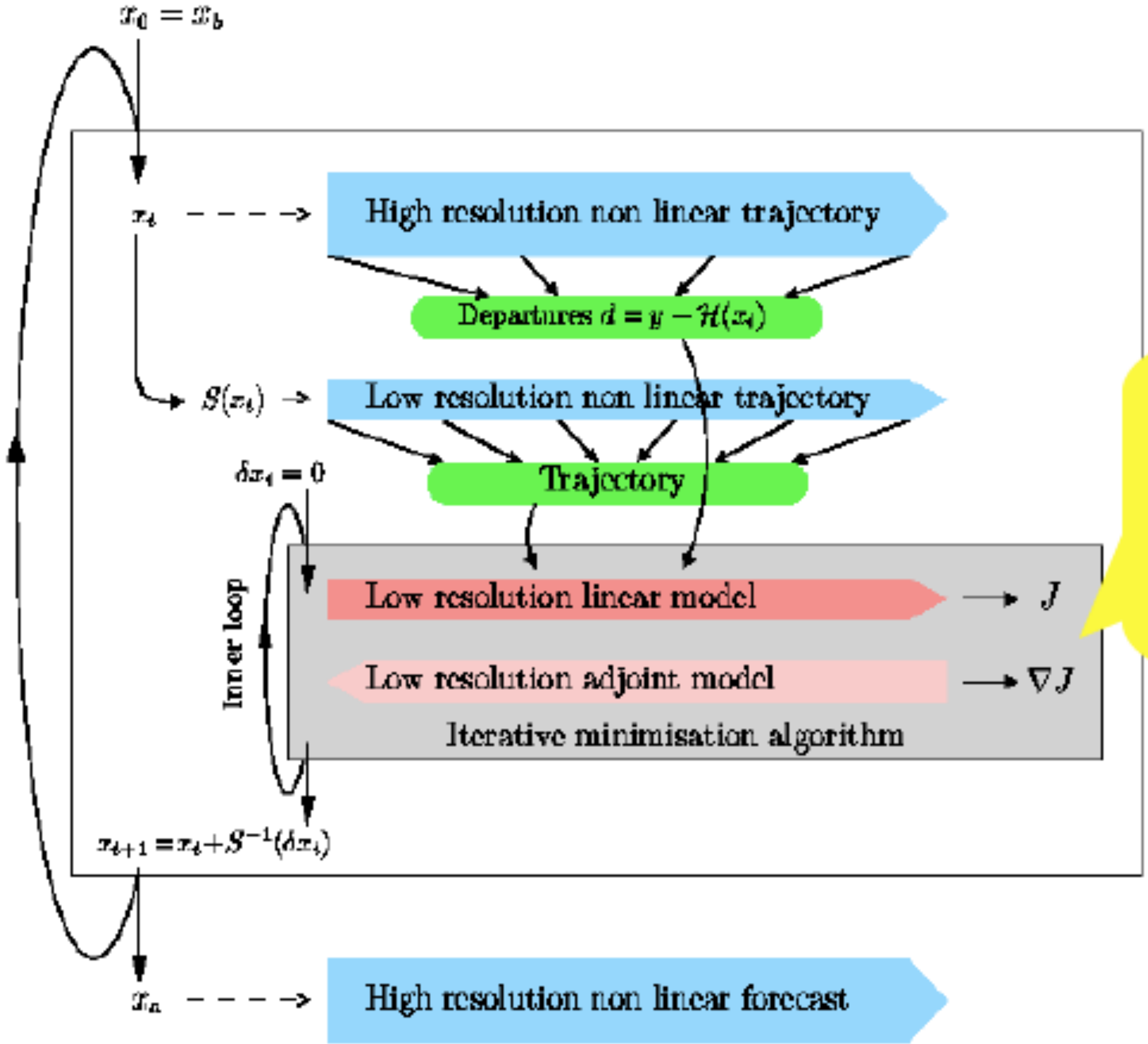
With this form, it is possible to choose a "simplification operator, $\mathbf{S}$" to solve the cost function in a low dimension space (change the control variable).

Now, $\delta\mathbf{w} = \mathbf{S}\delta\mathbf{x}$ and minimize $J(\delta\mathbf{w})$

The choice of the simplification operator

- Lower resolution
- Simplification of physical process

Example of using simplification operator



Both TLM and ADJ use a low resolution and also simplified physics due to the limitation of the computational cost.

TLM=Tangent Linear Model
ADJ= Adjoint model

Conclusions

- **Broad, Dynamic, Evolving, *Foundational* Science Field!**
- **Flexible unified frameworks, standards, and funding will improve training and education**
- **Continued need for advanced DA systems for research purposes (non-OPS)**
- **Can share OPS framework components, e.g., JCSDA http://www.jcsda.noaa.gov/**
- **JCSDA CRTM http://www.star.nesdis.noaa.gov/smcd/spb/CRTM/index.html**

REFERENCES

- **http://people.scs.fsu.edu/~navon/pubs/JCP1229.pdf**
- **http://www.ecmwf.int/newsevents/training/rcourse_notes/**
- **http://www.image.ucar.edu/DAReS/DART/index.php**
- **Geir Evensen: Data assimilation, The Ensemble Kalman Filter, 2nd ed., Springer, 2009** *see:* http://enkf.nersc.no/
- **Rodolfo Guzzi. Introduction to inverse methods with application to Remote Sensing and Geophysics,(in italian) Springer 2012**

Back up slides:The Role of the Adjoint, etc.

Adjoints are used in the cost function minimization procedure

But first…

Tangent Linear Models are used to approximate the non-linear model behaviors

$$\mathbf{L\ x'} = [\mathbf{M}(\mathbf{x}_1) - \mathbf{M}(\mathbf{x}_2)]\ /\ \alpha$$

L is the linear operator of the perturbation model

M is the non-linear forward model

α is the perturbation scaling-factor

$x_2 = x_1 + \alpha x'$

Useful Properties of the Adjoint

$\langle Lx', Lx' \rangle \cong \langle L^T Lx', x' \rangle$

L^T is the adjoint operator of the perturbation model

Typically the adjoint and the tangent linear operator can be automatically created using automated compilers

$$y = f(x_1, \ldots, x_n, y)$$

$$\delta^* x_i = \delta^* x_i + \delta^* y \, \partial f / \partial x_i$$

$$\delta^* y = \delta^* y \, \partial f / \partial y$$

where $\delta^* x_i$ and $\delta^* y$ are the "adjoint" variables

Useful Properties of the Adjoint

$$\langle Lx', Lx' \rangle \cong \langle L^T Lx', x' \rangle$$

L^T is the adjoint operator of the perturbation model

Typically the adjoint and the tangent linear operator can be automatically created using automated compilers

Of course, automated methods fail for complex variable types

(See Jones et al., 2004)

E.g., how can the compiler know when the variable is complex, when codes are decomposed into real and imaginary parts as common practice? (It can't.)