Data Assimilation
- A11. 4DVAR -

Shunji Kotsuki

Center for Environmental Remote Sensing / Institute of Advanced Academic Research

(shunji.kotsuki@chiba-u.jp)

CHIBA UNIVERSITY

Environmental Prediction Science Laboratory

DA Lectures A (Basic Course)

- (1) Introduction and NWP
- (2) Deterministic Chaos and Lorenz-96 model
- (3) A toy model and Bayesian estimation
- (4) Kalman Filter (KF)
- (5) 3D Variational Method (3DVAR)
- (6) Ensemble Kalman Filter (PO method)
- (7) Serial Ens. Square Root Filter (Serial EnSRF)
- (8) Local Ens. Transform Kalman Filter (LETKF)
- (9) Innovation Statistics
- (10) Adaptive Inflations
- (11) 4D Variational Method (4DVAR)

Today's Goal

- **Lecture**
  - what is the 4D-Var?
  - what is the cost function of 4DVAR?
  - what is adjoint and back propagation?

- **Training Course**
  - to implement 4DVAR

Review: 3DVAR

3DVAR Equations

Kalman Gain

$$\mathbf{K}_t = \mathbf{B}\mathbf{H}^T(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T)^{-1} = \mathbf{A}\mathbf{H}^T\mathbf{R}^{-1}$$

Analysis Error Covariance

$$\mathbf{A} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{B} \quad \Leftrightarrow \quad \mathbf{A}^{-1} = \mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$$

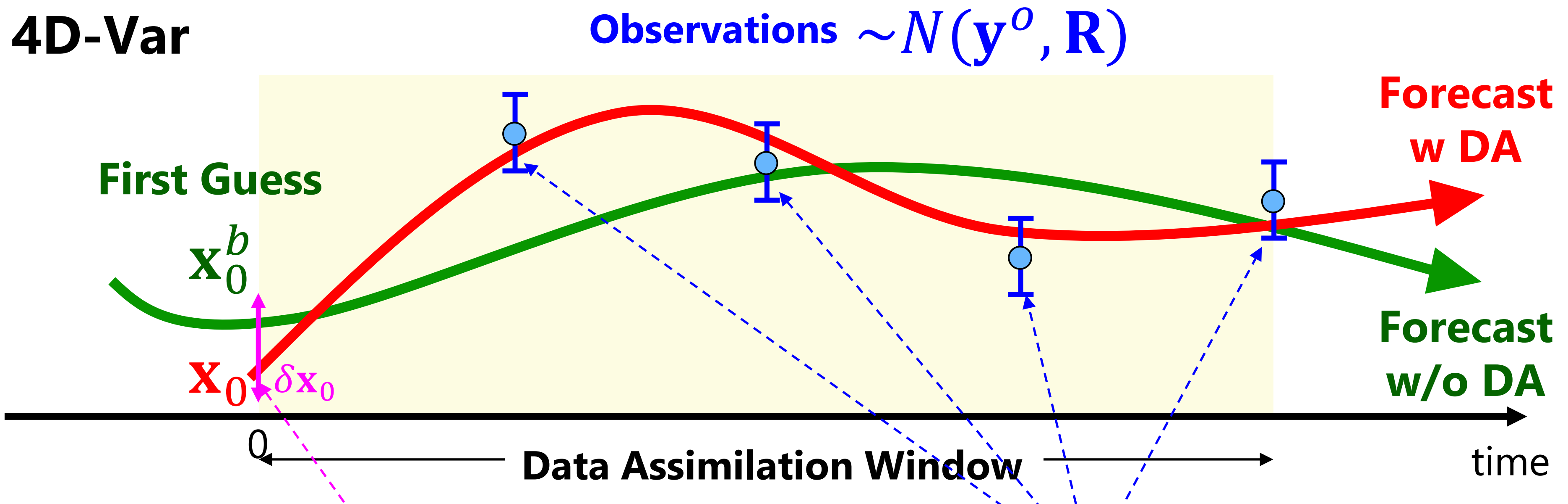
Analysis Update Equation

$$\mathbf{x}_t^a = \mathbf{x}_t^b + \mathbf{K}_t\mathbf{d}_t^{o-b} = \mathbf{A}\left[\mathbf{B}^{-1}\mathbf{x}_t^b + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{y}_t^o\right]$$

$$\Leftrightarrow \mathbf{A}^{-1}\mathbf{x}_t^a = \mathbf{B}^{-1}\mathbf{x}_t^b + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{y}_t^o$$

4DVAR

Conceptual Image of 4DVAR



Cost Function

$$J(\mathbf{x}_0) = \underbrace{\frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^b\right)^T \mathbf{B}_0^{-1}\left(\mathbf{x}_0 - \mathbf{x}_0^b\right)}_{\text{Regularization}} + \underbrace{\sum_{i=1}^{k} \frac{1}{2}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)^T \mathbf{R}_i^{-1}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)}_{\text{Misfit to 4d Obs}}$$

incremental form $\quad \mathbf{x}_0 = \mathbf{x}_0^b + \delta\mathbf{x}_0$

$$J(\delta\mathbf{x}_0) \approx \frac{1}{2}\delta\mathbf{x}_0^T \mathbf{B}_0^{-1}\delta\mathbf{x}_0 + \sum_{i=1}^{k} \frac{1}{2}\left(\mathbf{H}_i \mathbf{M}_{i|0}\delta\mathbf{x}_0 - \mathbf{d}_i^{o-b}\right)^T \mathbf{R}_i^{-1}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)$$

$$\mathbf{d}_i^{o-b} = \mathbf{y}_i^o - H_i(M_{i|0}(\mathbf{x}_0^b))$$

$M_{i|0}()$: nonlinear model from t=0 to t=i, $\quad H_i()$: nonlinear obs. operator at t=i

4DVAR Equation

Environmental Prediction Science Laboratory

Cost Function (scalar)

Tangent Linear Model $\mathbf{M}_{i|0} = \mathbf{M}_{i-1} \cdots \mathbf{M}_1 \mathbf{M}_0$

$$J(\delta\mathbf{x}_0) \approx \frac{1}{2}\delta\mathbf{x}_0^T \mathbf{B}_0^{-1}\delta\mathbf{x}_0 + \sum_{i=1}^{k}\frac{1}{2}\left(\mathbf{H}_i\mathbf{M}_{i|0}\delta\mathbf{x}_0 - \mathbf{d}_i^{o-b}\right)^T \mathbf{R}_i^{-1}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)$$

Jacobian ($\in \mathbb{R}^n$)

$$\frac{\partial J}{\partial(\delta\mathbf{x}_0)} \approx \mathbf{B}_0^{-1}(\delta\mathbf{x}_0) + \sum_{i=1}^{k}\underbrace{\overbrace{\boxed{\mathbf{M}_0^T \cdots \mathbf{M}_{i-2}^T\mathbf{M}_{i-1}^T}}^{\text{numerical model}}\overbrace{\boxed{\mathbf{H}_i^T}}^{\text{obs operator}}}_{\textbf{backpropagations}}\mathbf{R}_i^{-1}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)$$

Hessian ($\in \mathbb{R}^{n\times n}$)

$$\frac{\partial^2 J}{\partial(\delta\mathbf{x}_0)^2} \approx \mathbf{B}_0^{-1} + \sum_{i=1}^{k}\mathbf{M}_0^T \cdots \mathbf{M}_{i-2}^T\mathbf{M}_{i-1}^T\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i\,\mathbf{M}_{i-1}\cdots\mathbf{M}_1\mathbf{M}_0 = \mathbf{A}_0^{-1}$$

Problem to be solved

$$\delta\mathbf{x}_0^a = \operatorname{argmin} J(\delta\mathbf{x}_0) \quad \text{subject to} \quad \frac{\partial J}{\partial(\delta\mathbf{x}_0)} = 0$$

$$\mathbf{x}_0^a = \mathbf{x}_0^b + \delta\mathbf{x}_0^a$$

$$\mathbf{x}_i^a = M_{i|0}(\mathbf{x}_0^a)$$

Why Hessian is A^{-1}?

Environmental Prediction Science Laboratory

Cost Function w.r.t. background and analysis (scalar)

$$J(\mathbf{x}_0) = \frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^b\right)^T \mathbf{B}_0^{-1}\left(\mathbf{x}_0 - \mathbf{x}_0^b\right) + \sum_{i=1}^{k}\frac{1}{2}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)^T \mathbf{R}_i^{-1}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)$$

$$= \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^a)^T \mathbf{A}_0^{-1}(\mathbf{x}_0 - \mathbf{x}_0^a)$$

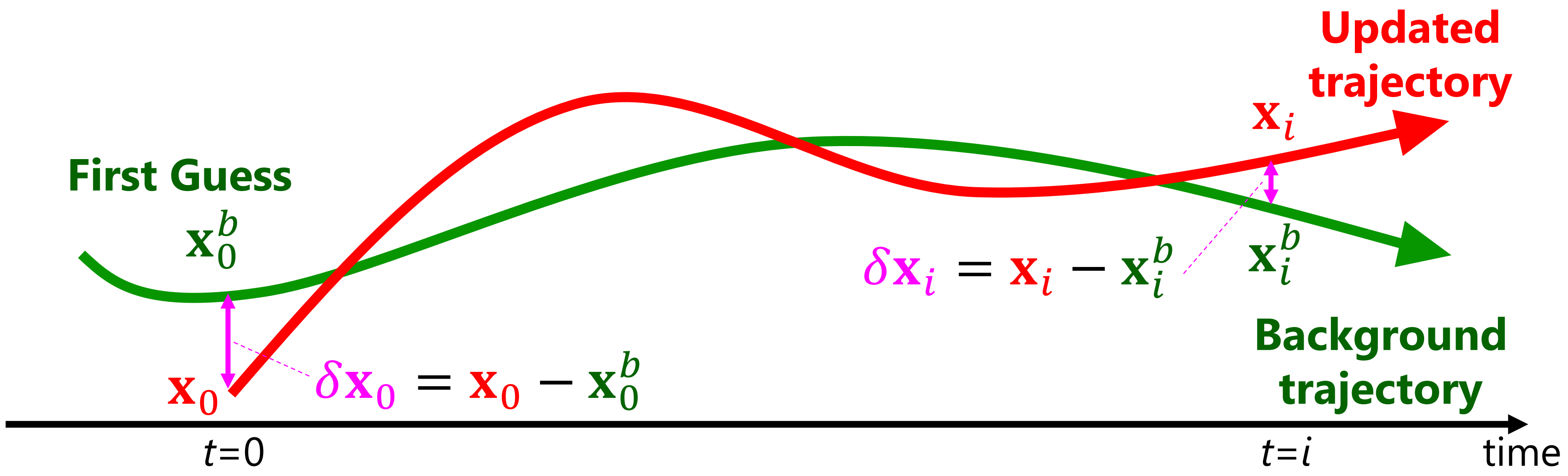
Jacobian ($\in \mathbb{R}^n$)

$$\frac{\partial J}{\partial(\mathbf{x}_0)} \approx \mathbf{A}_0^{-1}(\mathbf{x}_0 - \mathbf{x}_0^a)$$

Hessian ($\in \mathbb{R}^{n\times n}$)

$$\frac{\partial^2 J}{\partial(\mathbf{x}_0)^2} \approx \mathbf{A}_0^{-1}$$

Nonlinear, TLM & ADJ models



Nonlinear Models: $M_{i|0}()$, $H_i()$

$$\mathbf{x}_i^b = M_{i|0}(\mathbf{x}_0^b) \qquad \mathbf{x}_i = M_{i|0}(\mathbf{x}_0)$$

Tangent Linear Models (TLM): $\mathbf{M}_{i|0}$, $\mathbf{H}_i$

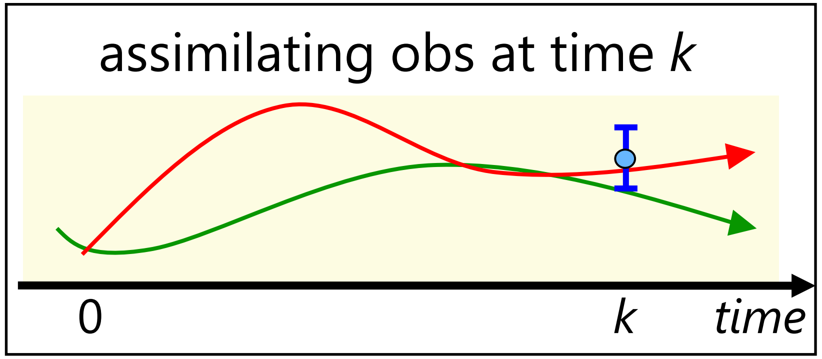
$$\delta\mathbf{x}_i \approx \mathbf{M}_{i|0}\delta\mathbf{x}_0 = \mathbf{M}_{i-1}\cdots\mathbf{M}_1\mathbf{M}_0\delta\mathbf{x}_0 \qquad H_i(\mathbf{x}_i) - H_i\left(\mathbf{x}_i^b\right) \approx \mathbf{H}_i\delta\mathbf{x}_0$$

Adjoint Model (ADJ): $\mathbf{M}_{i|0}^T$, $\mathbf{H}_i^T$

$$\mathbf{M}_0^T\cdots\mathbf{M}_{i-2}^T\mathbf{M}_{i-1}^T\mathbf{H}_i^T$$

Update needed

Flow-dependent B in 4DVAR



Analysis of 3DVAR

$$\mathbf{x}_k^a = \mathbf{x}_k^b + \mathbf{B}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{B}\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}\mathbf{d}_k^{o-b}$$

static background ($\mathbf{B}$) is used

Analysis of 4DVAR

$$J(\delta\mathbf{x}_0) \approx \frac{1}{2}\delta\mathbf{x}_0^T\mathbf{B}_0^{-1}\delta\mathbf{x}_0 + \frac{1}{2}\left(\mathbf{H}_k\mathbf{M}_{k|0}\delta\mathbf{x}_0 - \mathbf{d}_k^{o-b}\right)^T\mathbf{R}_k^{-1}\left(\mathbf{H}_k\mathbf{M}_{k|0}\delta\mathbf{x}_0 - \mathbf{d}_k^{o-b}\right)$$

$$\frac{\partial J}{\partial(\delta\mathbf{x}_0)} \approx \mathbf{B}_0^{-1}(\delta\mathbf{x}_0) + \mathbf{M}_{k|0}^T\mathbf{H}_k^T\mathbf{R}_k^{-1}\left(\mathbf{H}_k\mathbf{M}_{k|0}\delta\mathbf{x}_0 - \mathbf{d}_k^{o-b}\right) = 0$$

$$\Leftrightarrow \left(\mathbf{B}_0^{-1} + \mathbf{M}_{k|0}^T\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{M}_{k|0}\right)\delta\mathbf{x}_0 = \mathbf{M}_{k|0}^T\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{d}_k^{o-b}$$

$$\Leftrightarrow \delta\mathbf{x}_0 = \left(\mathbf{B}_0^{-1} + \mathbf{M}_{k|0}^T\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{M}_{k|0}\right)^{-1}\mathbf{M}_{k|0}^T\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{d}_k^{o-b}$$

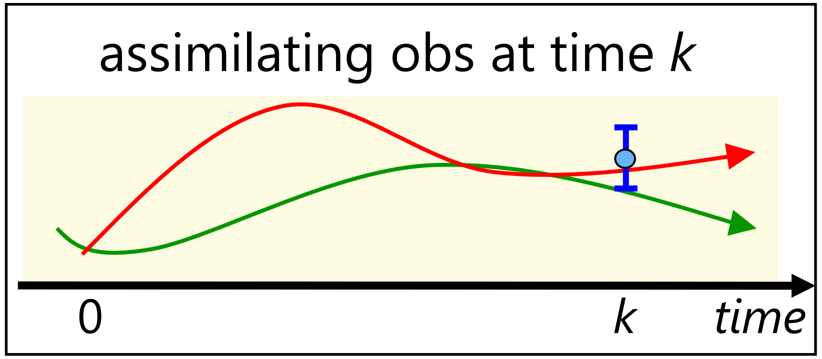
$$\Leftrightarrow \delta\mathbf{x}_0 = \mathbf{B}_0\mathbf{M}_{k|0}^T\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{M}_{k|0}\mathbf{B}_0\mathbf{M}_{k|0}^T\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}\mathbf{d}_k^{o-b}$$

$$\mathbf{x}_k^a = \mathbf{x}_k^b + M(\delta\mathbf{x}_0)$$

$$\approx \mathbf{x}_k^b + \mathbf{M}_{k|0}\mathbf{B}_0\mathbf{M}_{k|0}^T\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{M}_{k|0}\mathbf{B}_0\mathbf{M}_{k|0}^T\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}\mathbf{d}_k^{o-b}$$

flow-dependent background ($\mathbf{M}_{k|0}\mathbf{B}_0\mathbf{M}_{k|0}^T$) is used

The cost function at time *k*



Cost function of analysis time 0

$$J(\delta\mathbf{x}_0) \approx \frac{1}{2}\delta\mathbf{x}_0^T\mathbf{B}_0^{-1}\delta\mathbf{x}_0 + \frac{1}{2}\left(\mathbf{H}_k\mathbf{M}_{k|0}\delta\mathbf{x}_0 + \mathbf{d}_k^{o-b}\right)^T\mathbf{R}_k^{-1}\left(\mathbf{H}_k\mathbf{M}_{k|0}\delta\mathbf{x}_0 - \mathbf{d}_k^{o-b}\right)$$

Cost function of analysis time k

$$J(\delta\mathbf{x}_k) \approx \frac{1}{2}\delta\mathbf{x}_k^T\mathbf{B}_k^{-1}\delta\mathbf{x}_k + \frac{1}{2}\left(\mathbf{H}_k\delta\mathbf{x}_k + \mathbf{d}_k^{o-b}\right)^T\mathbf{R}_k^{-1}\left(\mathbf{H}_k\delta\mathbf{x}_k - \mathbf{d}_k^{o-b}\right)$$

Characteristics of Cost

Preparation

a simple 3DVAR case is considered

$$\mathbf{d} \equiv \mathbf{y}^o - H(\mathbf{x}^b) \qquad H(\mathbf{x}^a) - \mathbf{y}^o = \mathbf{H}(\mathbf{x}^a - \mathbf{x}^b) - \mathbf{d} = -\mathbf{R}(\mathbf{HBH}^T + \mathbf{R})^{-1}\mathbf{d}$$

$$\langle \mathbf{dd}^T \rangle = \mathbf{HBH}^T + \mathbf{R} \qquad \mathbf{x}^a - \mathbf{x}^b = \mathbf{BH}^T(\mathbf{HBH}^T + \mathbf{R})^{-1}\,\mathbf{d}$$

Cost Function w.r.t. analysis (scalar)

$$J(\mathbf{x}^a) = \frac{1}{2}(\mathbf{x}^a - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}^a - \mathbf{x}^b) + \frac{1}{2}(H(\mathbf{x}^a) - \mathbf{y}^o)^T\mathbf{R}^{-1}(H(\mathbf{x}^a) - \mathbf{y}^o)$$

$$= \frac{1}{2}\mathbf{d}^T(\mathbf{HBH}^T + \mathbf{R})^{-1}\mathbf{H}\cancel{\mathbf{BB}^{-1}}\mathbf{BH}^T(\mathbf{HBH}^T + \mathbf{R})^{-1}\,\mathbf{d}$$

$$+ \frac{1}{2}\mathbf{d}^T(\mathbf{HBH}^T + \mathbf{R})^{-1}\cancel{\mathbf{RR}^{-1}}\mathbf{R}(\mathbf{HBH}^T + \mathbf{R})^{-1}\,\mathbf{d}$$

$$= \frac{1}{2}\mathbf{d}^T(\mathbf{HBH}^T + \mathbf{R})^{-1}\,\mathbf{d} = \frac{1}{2}\mathrm{tr}[(\mathbf{HBH}^T + \mathbf{R})^{-1}\mathbf{dd}^T]$$

$$\langle J(\mathbf{x}^a)\rangle = \frac{1}{2}\mathrm{tr}[\mathbf{I}] = p/2$$

where p is the number of observations

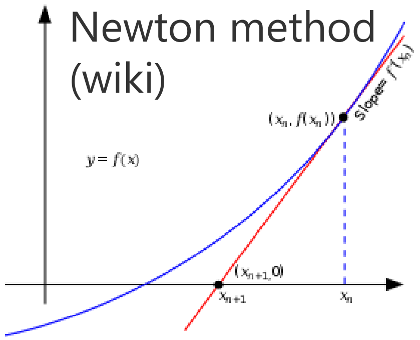
Iterative Solver
(BFGS method)

How to solve 4DVAR?

Problem to be solved

$$\delta\mathbf{x}_0^a = \text{argmin}\, J(\delta\mathbf{x}_0) \qquad \text{subject to} \frac{\partial J}{\partial(\delta\mathbf{x}_0)} = 0$$

Newton method (wiki)



Newton Method

after jth iteration ($\delta\mathbf{x}_0^j$), to obtain $\mathbf{s}_j$ that satisfies: $\nabla J\left(\delta\mathbf{x}_0^j + \mathbf{s}^j\right) = 0$

Talyer expansion gives $\nabla J\left(\delta\mathbf{x}_0^j\right) + J''\left(\delta\mathbf{x}_0^j\right)\mathbf{s}^j \approx 0 \Leftrightarrow \mathbf{s}^j \approx -\left[J''\left(\delta\mathbf{x}_0^j\right)\right]^{-1} \nabla J\left(\delta\mathbf{x}_0^j\right)$

$$\underline{\delta\mathbf{x}_0^{j+1} = \delta\mathbf{x}_0^j + \mathbf{s}^j = \delta\mathbf{x}_0^j - \left[J''\left(\delta\mathbf{x}_0^j\right)\right]^{-1} \nabla J\left(\delta\mathbf{x}_0^j\right)}$$

to update $\delta\mathbf{x}_0^j$ iteratively

And obtain $\delta\mathbf{x}_0^a$ when $\left|\delta\mathbf{x}_0^{j+1} - \delta\mathbf{x}_0^j\right| \approx 0$

Here, Hessian matrix ($J''\left(\delta\mathbf{x}_0^j\right) \in \mathbb{R}^{n\times n}$) is given by

$$J''\left(\delta\mathbf{x}_0^j\right) \approx \mathbf{B}_0^{-1} + \sum_{i=1}^{k} \mathbf{M}_0^T \cdots \mathbf{M}_{i-2}^T \mathbf{M}_{i-1}^T \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i \mathbf{M}_{i-1} \cdots \mathbf{M}_1 \mathbf{M}_0 = \mathbf{A}_0^{-1}$$

➔ **Quasi-Newton method**:

(1) without having $\mathbf{A}_0^{-1}$ explicitly

(2) to approximate $\mathbf{A}_0^{-1}$by a positive definite matrix $\mathbf{Q}$ for inversion

Quasi-Newton Method

Secant Conditions

Let $\mathbf{Q}^j$ be a positive definite matrix that approximates the Hessian matrix (i.e., $\mathbf{Q}^j \approx J''\left(\delta\mathbf{x}_0^j\right) \approx \mathbf{A}_0^{-1}$)

Suppose we have $\mathbf{Q}^j$, and would like to update $\mathbf{Q}^{j+1}$ for the next iteration

Taylor Approximation

$$\nabla J\left(\delta\mathbf{x}_0^j\right) = \nabla J\left(\delta\mathbf{x}_0^{j+1} - \mathbf{s}^j\right) \approx \nabla J\left(\delta\mathbf{x}_0^{j+1}\right) - J''\left(\delta\mathbf{x}_0^{j+1}\right)\mathbf{s}^j \approx \nabla J\left(\delta\mathbf{x}_0^{j+1}\right) - \mathbf{Q}^{j+1}\mathbf{s}^j$$

Secant Equation

Secant condition $\mathbf{Q}^{j+1}\mathbf{s}^j = \mathbf{u}^j$ where

$$\mathbf{u}^j = \nabla J\left(\delta\mathbf{x}_0^{j+1}\right) - \nabla J\left(\delta\mathbf{x}_0^j\right)$$

$$\mathbf{s}^j = \delta\mathbf{x}_0^{j+1} - \delta\mathbf{x}_0^j$$

There are many matrices that satisfy the Secant Condition (e.g., DFP, BFGS, SR1).

BFGS method (widely used in 4DVAR for NWP)

$$\mathbf{Q}^{j+1} = \mathbf{Q}^j - \frac{\mathbf{Q}^j\mathbf{s}^j\left(\mathbf{Q}^j\mathbf{s}^j\right)^T}{(\mathbf{s}^j)^T\mathbf{Q}^j\mathbf{s}^j} + \frac{\mathbf{u}^j(\mathbf{u}^j)^T}{(\mathbf{s}^j)^T\mathbf{u}^j}$$

Usually $\mathbf{Q}^0 = \mathbf{I}$

Algorithm of BFGS method

1	to set initial condition ($\delta\mathbf{x}_0^0 = 0$ and $\mathbf{Q}^0 = \mathbf{I}$)
2	to compute the update direction by $\mathbf{s}^j = -\left(\mathbf{Q}^j\right)^{-1}\nabla J\left(\delta\mathbf{x}_0^j\right)$
3	to compute the update parameter α^j by Armijo Condition (there may be other efficient condition such as Wolfe's condition)
4	to update $\delta\mathbf{x}_0^{j+1} = \delta\mathbf{x}_0^j + \alpha^j\mathbf{s}^j$
5	to update $\mathbf{Q}^{j+1} = \mathbf{Q}^j - \frac{\mathbf{Q}^j\mathbf{s}^j\left(\mathbf{Q}^j\mathbf{s}^j\right)^T}{(\mathbf{s}^j)^T\mathbf{Q}^j\mathbf{s}^j} + \frac{\mathbf{u}^j(\mathbf{u}^j)^T}{(\mathbf{s}^j)^T\mathbf{u}^j}$
6	to repeat steps 2-5 until $\left\|\nabla J\left(\delta\mathbf{x}_0^{j+1}\right)\right\| < \varepsilon$ e.g. $\varepsilon = 10^{-4}$

iteration

Armijo Condition

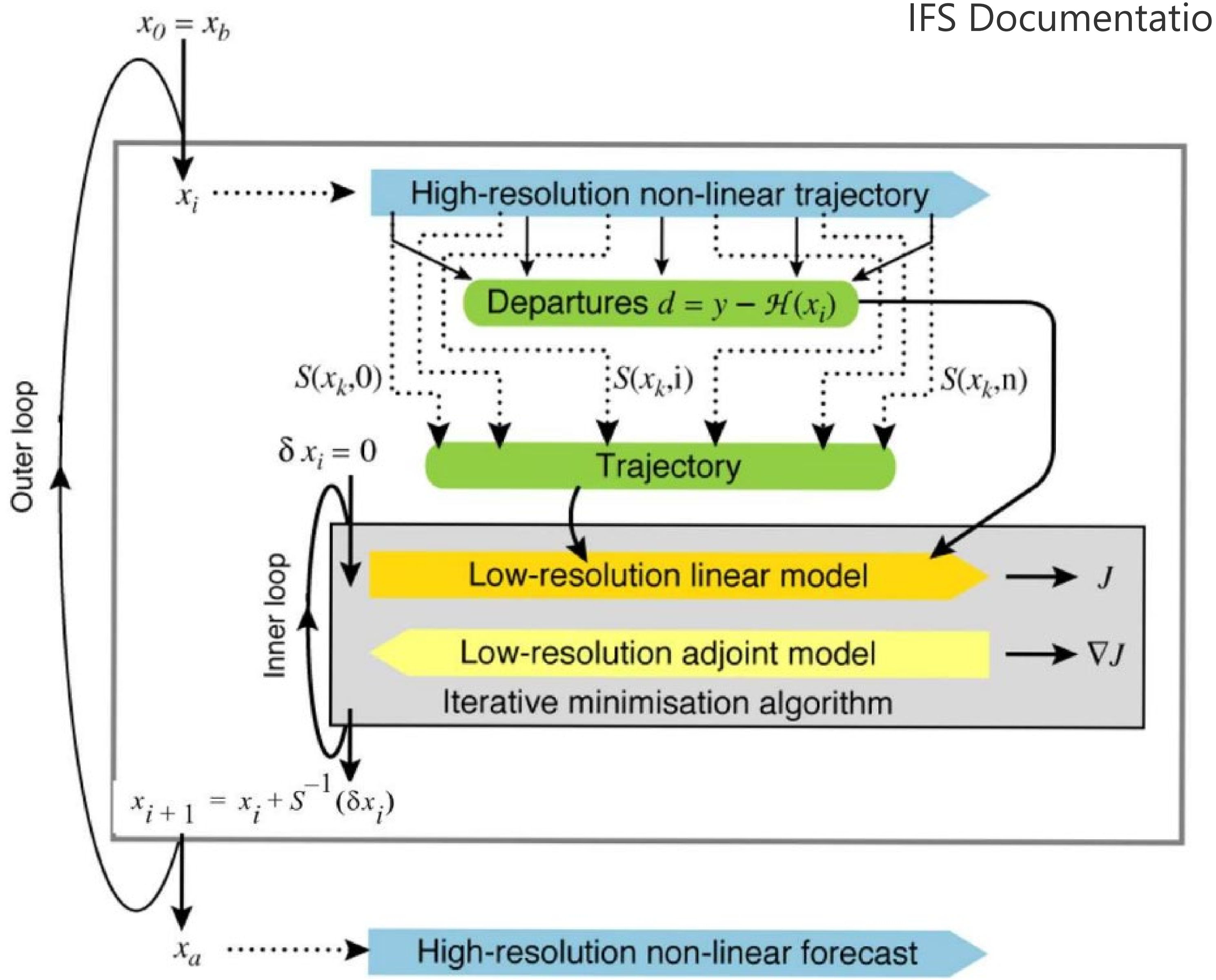
To obtain an appropriate update parameter α^j

Armijo Condition	$J\left(\delta\mathbf{x}_0^j + \alpha^j \mathbf{s}^j\right) \leq J\left(\delta\mathbf{x}_0^j\right) + \xi\alpha^j \nabla J\left(\delta\mathbf{x}_0^j\right)^T \mathbf{s}^j$ $0 < \xi < 1$
Algorithm	1． set parameters $0 < \xi < 1$, $0 < \tau < 1$
	2． set initial condition: $\alpha_0^j = 1$
	3． end the algorithm if α_k^jsatisfies the condition
	4． If not, update$\alpha_{k+1}^j = \tau\alpha_k^j$and go back Step 3

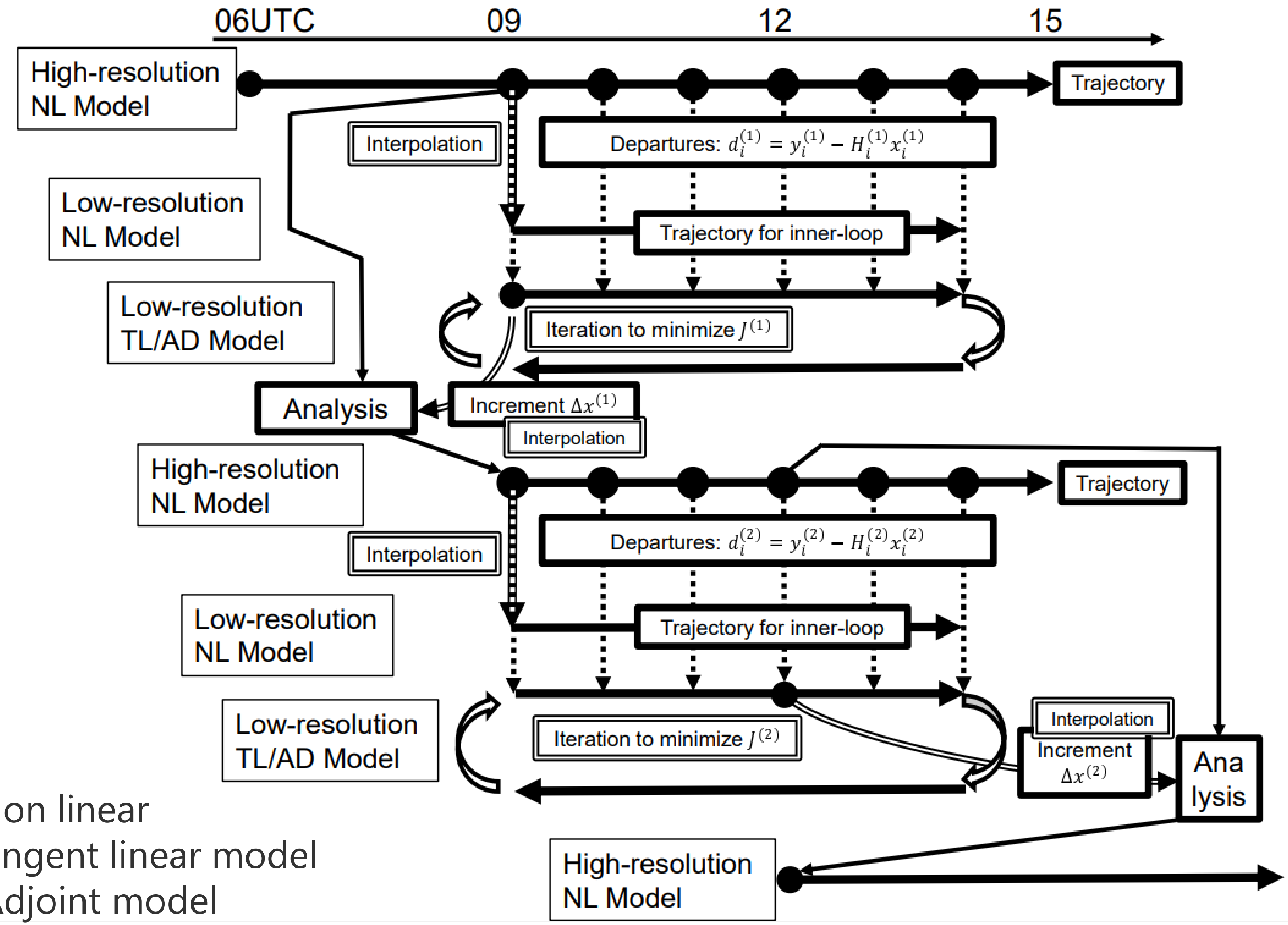
Outer and Inner Loops

ECMWF 4DVAR (2015)

IFS Documentation (2015)



JMA 4DVAR (2022)



NL: Non linear
TL: Tangent linear model
AD: Adjoint model

https://www.jma.go.jp/jma/jma-eng/jma-center/nwp/outline2022-nwp/pdf/outline2022_02.pdf

Solver of JMA 4DVAR (2022)

- The limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm (Liu and Nocedal 1989)
  - with Veersé's preconditioner (Veersé et al. 2000)

$$J^{(j)}\left(\Delta x_0^{(j)}\right) = \frac{1}{2}\left(\sum_{l=1}^{j} \Delta x_0^{(l)}\right)^{\mathrm{T}} \mathbf{B}^{-1}\left(\sum_{l=1}^{j} \Delta x_0^{(l)}\right) + \frac{1}{2}\sum_{i=1}^{n}\left(\mathbf{H}_i^{(j)}\Delta x_i^{(j)} - d_i^{(j)}\right)^{\mathrm{T}} \mathbf{R}_i^{(j)^{-1}}\left(\mathbf{H}_i^{(j)}\Delta x_i^{(j)} - d_i^{(j)}\right) + J_C^{(j)} \tag{2.5.1}$$

$$\Delta x_{i+1}^{(j)} = \mathbf{M}_i^{(j)}\Delta x_i^{(j)} \qquad (i = 0, \ldots, n-1)$$

The penalty term, which is the third term of Eq. (2.5.1), is given by

$$J_C^{(j)} = \frac{1}{2}\alpha\left(\left|\mathrm{N}_G \sum_{l=1}^{j} \Delta x_0^{(l)}\right|^2 + \sum_{i=2}^{\mathrm{n}}\left|\mathrm{N}_G \sum_{l=1}^{j} \Delta x_i^{(l)}\right|^2\right) \tag{2.5.7}$$

where N_G denotes an operator used to calculate the tendency of the gravity wave mode based on Machenhauer (1977). α is an empirically determined constant $3.0 \times 10^{-2}[\mathrm{s}^4/\mathrm{m}^2]$. Although this penalty term is primarily introduced to suppress gravity waves in the analysis increment, it is also effective in stabilizing calculation.

to be updated

JMS's NWP System (2022)

Table 2.1.1: Specifications of 4D-Var in Global Analysis (GA)

Analysis time	00, 06, 12, and 18 UTC
Analysis scheme	Incremental hybrid 4D-Var using LETKF
Data cut-off time	2 hours and 20 minutes for early run analysis at 00, 06, 12, and 18 UTC 11 hours and 50 minutes for cycle run analysis at 00 and 12 UTC 7 hours and 50 minutes for cycle run analysis at 06 and 18 UTC
First guess	6-hour forecast by the GSM
Domain configuration (Outer step)	Globe TL959, Reduced Gaussian grid, roughly equivalent to 0.1875 ° (20 km) [1920 (tropic) – 60 (polar)] × 960
(Inner step)	TL319, Reduced Gaussian grid, roughly equivalent to 0.5625 ° (55 km) [640 (tropic) – 60 (polar)] × 320
Vertical coordinate	σ-p hybrid
Vertical levels	128 forecast model levels up to 0.01 hPa + surface
Analysis variables	Wind, surface pressure, specific humidity and temperature
Observation (as of 31 March 2021)	SYNOP, METAR, SHIP, BUOY, TEMP, PILOT, Wind Profiler, AIREP, AMDAR, Typhoon Bogus; atmospheric motion vectors (AMVs) from Himawari-8, GOES-16 and Meteosat-[8, 11]; MODIS polar AMVs from Terra satellite; AVHRR polar AMVs from NOAA and Metop satellites; LEO-GEO AMVs; ocean surface wind from Metop-[A, B, C]/ASCAT and ScatSat-1/OSCAT; radiances from NOAA-15/AMSU-A, NOAA-[18, 19]/ATOVS, Metop-[A, B, C]/ATOVS, Aqua/AMSU-A, DMSP-F[17, 18]/SSMIS, Suomi-NPP/ATMS, NOAA-20/ATMS, GCOM-W/AMSR2, GPM-core/GMI, Megha-Tropiques/SAPHIR, Aqua/AIRS, Metop-[A, B]/IASI, Suomi-NPP/CrIS, and NOAA-20/CrIS; clear sky radiances from the water vapor channels (WV-CSRs) of Himawari-8, GOES-16 and Meteosat-[8, 11]; GNSS RO bending angle data from Metop-[A, B]/GRAS and TerraSAR-X/IGOR; zenith total delay data from ground-based GNSS
Assimilation window	6 hours

Table 2.1.2: Specifications of the Mesoscale Analysis (MA)

Analysis time	00, 03, 06, 09, 12, 15, 18, and 21 UTC
Analysis scheme	Incremental 4D-Var using a nonlinear forward model in the inner step with low resolution
Data cut-off time	50 minutes for analysis at 00, 03, 06, 09, 12, 15, 18, and 21 UTC
First guess	3-hour forecast produced by ASUCA
Domain configuration	Japan and its surrounding area
(Outer step)	Lambert projection: 5 km at 60°N and 30°N, 817 × 661 Grid point (1, 1) is at the northwest corner of the domain. Grid point (565, 445) is at 140°E, 30°N
(Inner step)	Lambert projection: 15 km at 60°N and 30°N, 273 × 221 Grid point (1, 1) is at the northwest corner of the domain. Grid point (189, 149) is at 140°E, 30°N
Vertical coordinate	z-$z^\star$ hybrid
Vertical levels	(Outer step) 76 levels up to 21.8 km (Inner step) 38 levels up to 21.8 km
Analysis variables	Wind, potential temperature, surface pressure, pseudo-relative humidity, soil temperature and soil volumetric water content
Observations (as of 31 March 2021)	SYNOP, SHIP, BUOY, TEMP, PILOT, Wind Profiler, Weather Doppler radar (radial velocity, reflectivity), AIREP, AMDAR, Typhoon Bogus; AMVs from Himawari-8; ocean surface wind from Metop-[A, B]/ASCAT; radiances from NOAA-15/AMSU-A, NOAA-[18, 19]/ATOVS, Metop-[A, B]/ATOVS, Aqua/AMSU-A, DMSP-F[17, 18]/SSMIS, GCOM-W/AMSR2 and GPM-core/GMI; clear sky radiances from the water vapor channels (WV-CSRs) of Himawari-8; Radar/Raingauge-Analyzed Precipitation; precipitation retrievals from DMSP-F[17, 18]/SSMIS, GCOM-W/AMSR2 and GPM-core/GMI; GPM-core/DPR; GNSS RO refractivity data from Metop-[A, B]/GRAS, TerraSAR-X/IGOR and TanDEM-X/IGOR; Total Precipitable Water Vapor from ground-based GNSS
Assimilation window	3 hours

4DVARs in ECMWF & JMA

Environmental Prediction Science Laboratory

ECMWF (2015)

Outer loop: High-res. NL

(O1) high-res. NL model

(O2) departure: $\mathbf{d}_i^{o-b} = \mathbf{y}_i^o - H_i(M_{i|0}(\mathbf{x}_0))$

(O3) get TLM and ADJ models based on the trajectory of high-res. NL model

Inner loop: Low-res. TLM & ADJ

(I1) Cost and its gradient

$$J(\delta\mathbf{x}_0) \approx \frac{1}{2}\delta\mathbf{x}_0^T\mathbf{B}_0^{-1}\delta\mathbf{x}_0 - \sum_{i=1}^{k}\frac{1}{2}\left(\mathbf{H}_i\mathbf{M}_{i|0}\delta\mathbf{x}_0 - \mathbf{d}_i^{o-b}\right)^T\mathbf{R}_i^{-1}\mathbf{d}_i^{o-b}$$

$$\nabla J \approx \mathbf{B}_0^{-1}(\delta\mathbf{x}_0) - \sum_{i=1}^{k}\mathbf{M}_{i|0}^T\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{d}_i^{o-b}$$

(I2) BGGS to update $\delta\mathbf{x}_0$

① iteration

(O4) update analysis $\mathbf{x}_0 = \mathbf{x}_0 + \delta\mathbf{x}_0$

② iteration

JMA (2022)

Outer loop: High- & Low-res. NL

(O1) high-res. NL model

(O2) departure: $\mathbf{d}_i^{o-b} = \mathbf{y}_i^o - H_i(M_{i|0}(\mathbf{x}_0))$

(O3) low-res. NL model

(O4) get TLM and ADJ models based on the trajectory of low-res. NL model

Inner loop: Low-res. TLM & ADJ

(I1) Cost and its gradient

$$J(\delta\mathbf{x}_0) \approx \frac{1}{2}\delta\mathbf{x}_0^T\mathbf{B}_0^{-1}\delta\mathbf{x}_0 - \sum_{i=1}^{k}\frac{1}{2}\left(\mathbf{H}_i\mathbf{M}_{i|0}\delta\mathbf{x}_0 - \mathbf{d}_i^{o-b}\right)^T\mathbf{R}_i^{-1}\mathbf{d}_i^{o-b}$$

$$\nabla J \approx \mathbf{B}_0^{-1}(\delta\mathbf{x}_0) - \sum_{i=1}^{k}\mathbf{M}_{i|0}^T\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{d}_i^{o-b}$$

(I2) BGGS to update $\delta\mathbf{x}_0$

① iteration

(O4) update analysis $\mathbf{x}_0 = \mathbf{x}_0 + \delta\mathbf{x}_0$

② iteration

Use Low-res. NL that is consistent with TLM & ADJ

Cost Function

Andersson et al. (2005)

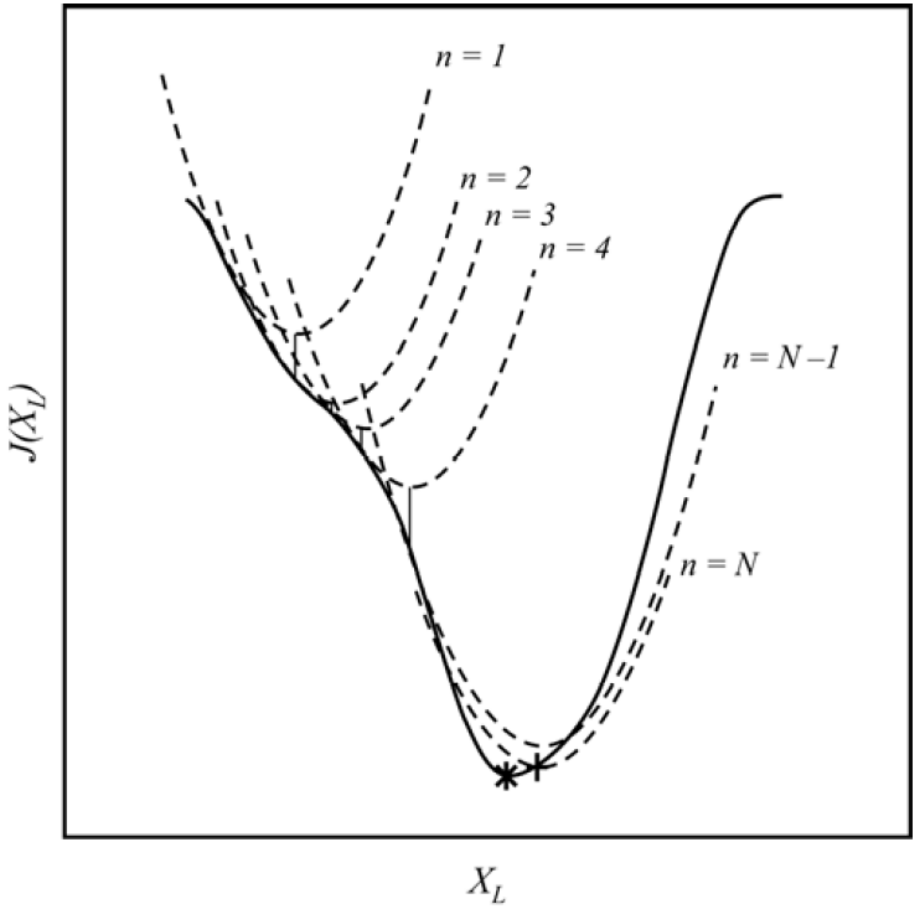


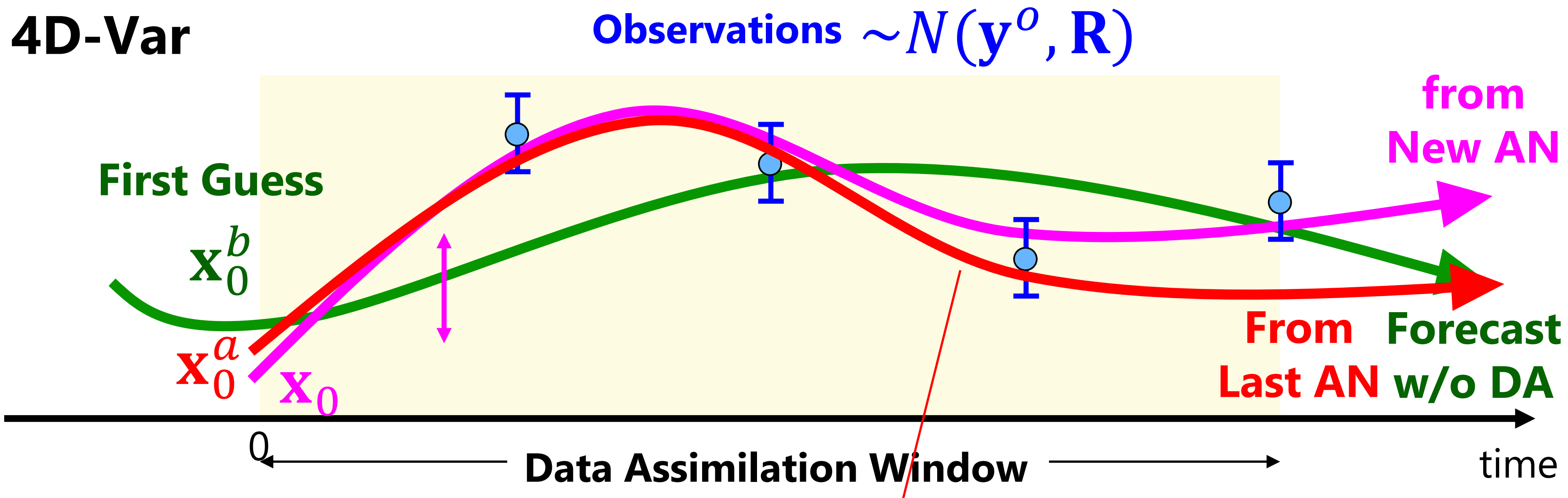
Figure 1. Schematic of the inner/outer solution algorithm, in which the full nonlinear problem, with a non-quadratic cost-function (full line) is solved through a succession of N linear problems, with quadratic cost-functions (dashed), approximating the original problem (from Laroche and Gauthier, 1998). The N re-linearizations constitute the outer-loop, while iterative minimisation of each of the linear problems constitutes the inner loops.

Regularization

- (1) Penalty term for gravity wave
  - Not to regulate small increments for fitting to obs

- (2) incremental form
  - Smoothed TLM in the inner loop
    - Low-resolution
    - Regulated non-linear terms
  - →the smoothed TLM makes cost function smoother

Standard 4DVAR



Cost Function

$$J(\mathbf{x}_0) = \frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^b\right)^T \mathbf{B}_0^{-1}\left(\mathbf{x}_0 - \mathbf{x}_0^b\right) + \sum_{i=1}^{k} \frac{1}{2}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)^T \mathbf{R}_i^{-1}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)$$

Jacobian from the latest trajectory
(updated by outer loop)

Gradient

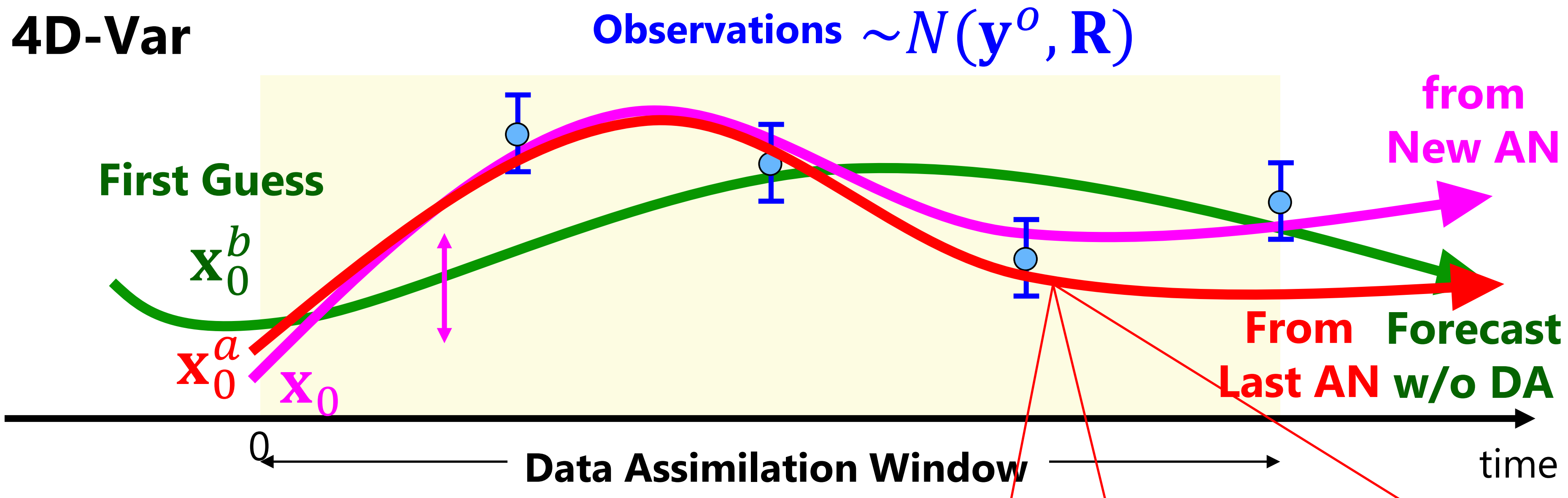
$$\frac{\partial J}{\partial(\mathbf{x}_0)} \approx \mathbf{B}_0^{-1}(\mathbf{x}_0) + \sum_{i=1}^{k} \underline{\mathbf{M}_0^T \cdots \mathbf{M}_{i-2}^T \mathbf{M}_{i-1}^T \mathbf{H}_i^T} \mathbf{R}_i^{-1}\left(H_i(M_{i|0}(\mathbf{x}_0)) - \mathbf{y}_i^o\right)$$

$M_{i|0}()$: nonlinear model from t=0 to t=i, $H_i()$: nonlinear obs. operator at t=i

Outer & inner-loop 4DVAR

Environmental Prediction Science Laboratory

4D-Var



Cost Function

$$\mathbf{d}_i^{o-b} = \mathbf{y}_i^o - H_i(M_{i|0}(\underline{\mathbf{x}_0^a}))$$

$$J(\delta\mathbf{x}_0) \approx \frac{1}{2}(\delta\mathbf{x}_0^a + \delta\mathbf{x}_0)^T \mathbf{B}_0^{-1}(\delta\mathbf{x}_0^a + \delta\mathbf{x}_0) + \sum_{i=1}^{k} \frac{1}{2}\left(\mathbf{H}_i \mathbf{M}_{i|0} \delta\mathbf{x}_0 - \mathbf{d}_i^{o-b}\right)^T \mathbf{R}_i^{-1}\left(\mathbf{H}_i \mathbf{M}_{i|0} \delta\mathbf{x}_0 - \mathbf{d}_i^{o-b}\right)$$

from the latest trajectory
(updated by outer loop)

Gradient

$$\frac{\partial J}{\partial(\delta\mathbf{x}_0)} \approx \mathbf{B}_0^{-1}(\delta\mathbf{x}_0^a + \underline{\delta\mathbf{x}_0}) + \sum_{i=1}^{k} \underline{\mathbf{M}_0^T \cdots \mathbf{M}_{i-2}^T \mathbf{M}_{i-1}^T \mathbf{H}_i^T} \mathbf{R}_i^{-1}\left(\underline{\mathbf{H}_i \mathbf{M}_{i|0}}\,\underline{\delta\mathbf{x}_0} - \mathbf{d}_i^{o-b}\right)$$

updated over inner loop

$M_{i|0}()$: nonlinear model from t=0 to t=i, $H_i()$: nonlinear obs. operator at t=i

Training Course

DA Study w/ 40-variable Lorenz-96

Lorenz-96 model (Lorenz 1996)

For $j=1,\ldots,N$, $X_j=X_{j+N}$

$$dX_j\,/dt = \underbrace{\left(X_{j+1} - X_{j-2}\right)X_{j-1}}_{\text{Advection term}} - \underbrace{X_j}_{\text{Dissipation term}} + \underbrace{F}_{\text{Forcing term}}$$

力学系モデル・データ同化基礎技術の速習コース

Training Course of Dynamical Model and Data Assimilation

January 31, 2020, Shunji Kotsuki

updated 2020/03/19, 2020/06/29, 2021/07/15

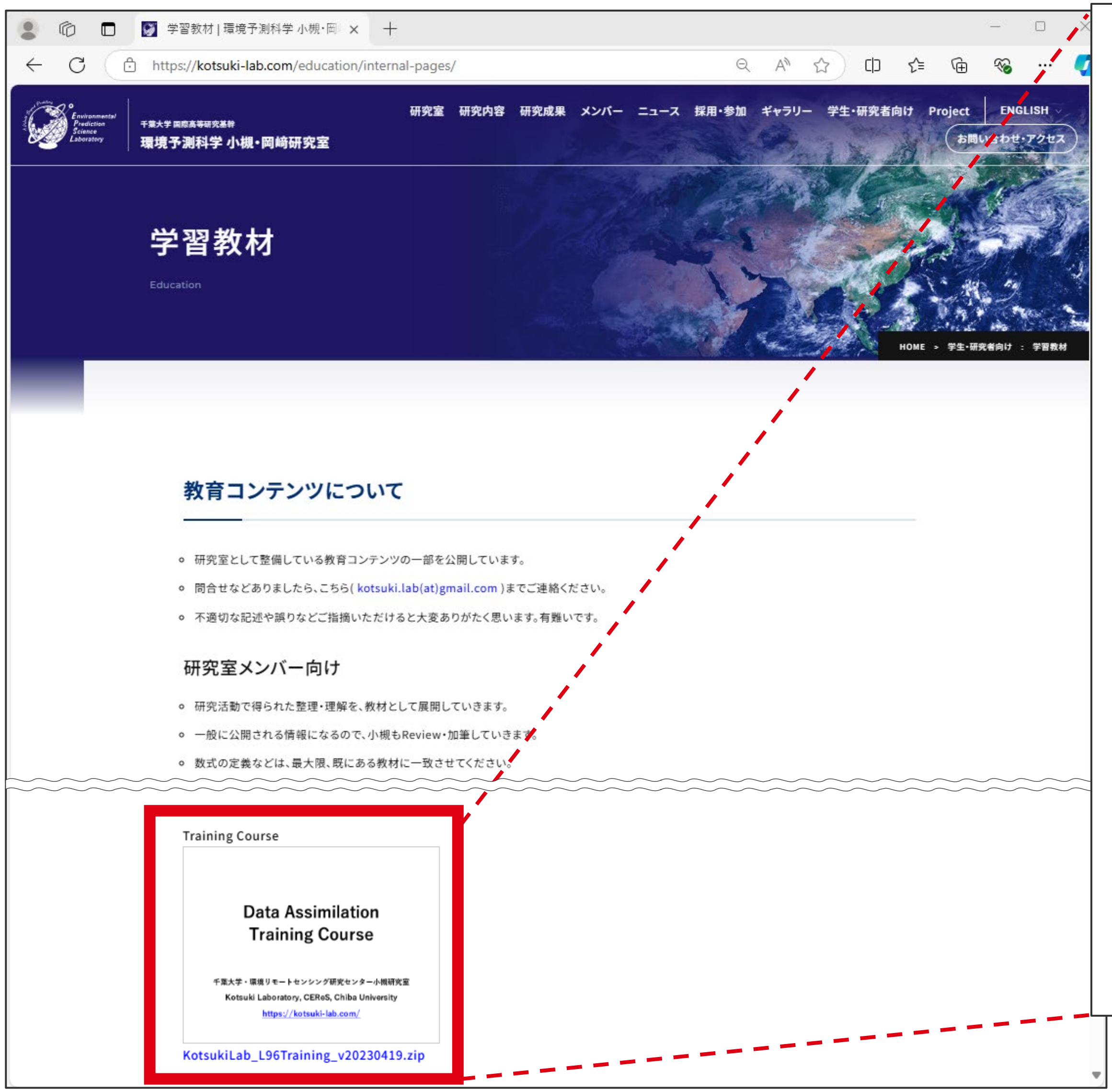
目的： 簡易力学モデル Lorenz の 40 変数モデル（以下 L96; Lorenz 1996）を使って複数のデータ同化手法を自ら実装し、様々な実験を行う。データ同化システムを実際に、0 からコーディングすることで、力学モデリングやデータ同化に関する実践的な「使える」基礎技術を体得する。

Purpose: Using the 40-variable dynamical a.k.a. Lorenz-96 (L96; Lorenz 1996), we are going to perform various experiments with multiple data assimilation (DA) methods. By actually coding a data assimilation system from scratch, you will acquire practically "usable" basic techniques related to mechanical modeling and data assimilation.

Text Books

① Training Description

pswd: ceres



力学系モデル・データ同化基礎技術の速習コース

Training Course of Dynamical Model and Data Assimilation

January 31, 2020, Shunji Kotsuki
updated 2020/03/19, 2020/06/29, 2021/07/15

目的： 簡易力学モデル Lorenz の 40 変数モデル（以下 L96; Lorenz 1996）を使って複数のデータ同化手法を自ら実装し、様々な実験を行う。データ同化システムを実際に、0 からコーディングすることで、力学モデリングやデータ同化に関する実践的な「使える」基礎技術を体得する。

Purpose: Using the 40-variable dynamical a.k.a. Lorenz-96 (L96; Lorenz 1996), we are going to perform various experiments with multiple data assimilation (DA) methods. By actually coding a data assimilation system from scratch, you will acquire practically "usable" basic techniques related to mechanical modeling and data assimilation.

方法： 以下の課題を自ら実装し、解決していく。使用言語やプラットフォームは問わない。研究室の MTG において、各自が進捗を報告し、問題点を解消していく。質問は MTG の他も、居室で適宜受け付ける。使用言語については、特に拘りがなければ、行列演算の容易な python が扱いやすい。また、単精度ではなく倍精度でコーディングする事。でないと、既往研究と比較して正しく動作しているか確認できない。

Method: Implement and solve the following problems yourself. Any programing languages or platforms can be used in this exercise. At the Kotsuki Lab. mtg, each personnel will report the progress, and try to solve the problems. Questions are accepted during the MTG as well as at the office when necessary. As for the programing language, python, which is easy to perform matrix operations, is recommended unless specific language is preferred. Also, you should code in double precision instead of single precision. Otherwise, confirming whether performing properly or not compared to the previous studies will not be possible.

▸ https://kotsuki-lab.com/internal-pages/

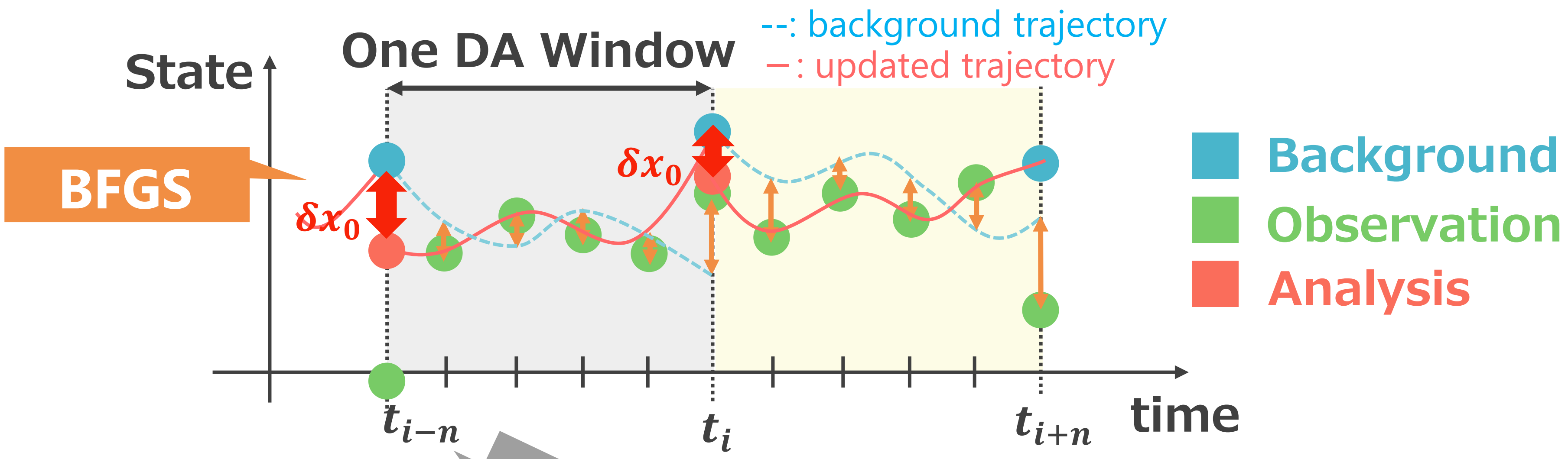
Advanced Task 4

Advanced Task 4

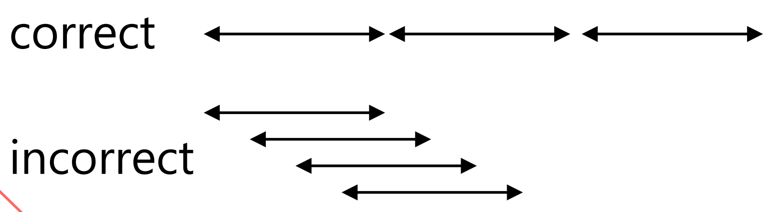
4. **難易度 S, 研究発展性 C [4 次元変分法]:** 4 次元変分法を実装し、EnKF と比較する。4 次元変分法には、アジョイントモデルを構築する他、近似的に 40×40 行列の線形モデルを生成する方法もある。もしアジョイントモデルを構築すれば、近似的な線形モデル行列との違いを調べてみるのも面白いかもしれない。

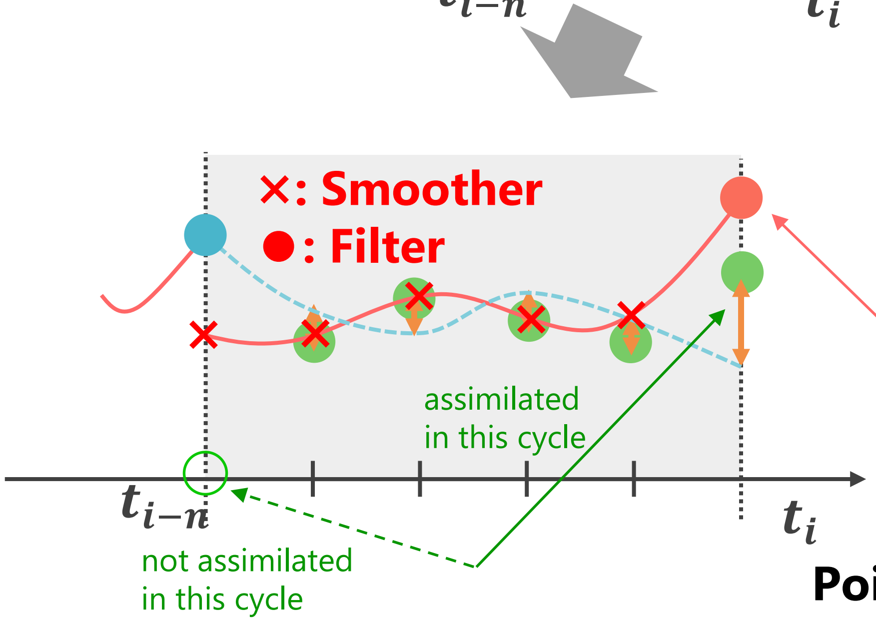
4. **Difficulty S, Scientific Extensionality C [4D variational method]:** Implement the 4D variational method and compare it with EnKF. In addition to constructing an adjoint model, the 4D variational method also includes a method of approximately generating a linear model of a 40 × 40 matrix. When building an adjoint model, it may be interesting to see how it differs from an approximate linear model matrix.

4DVAR Implementation



Point 1: the observation should not be assimilated more than once

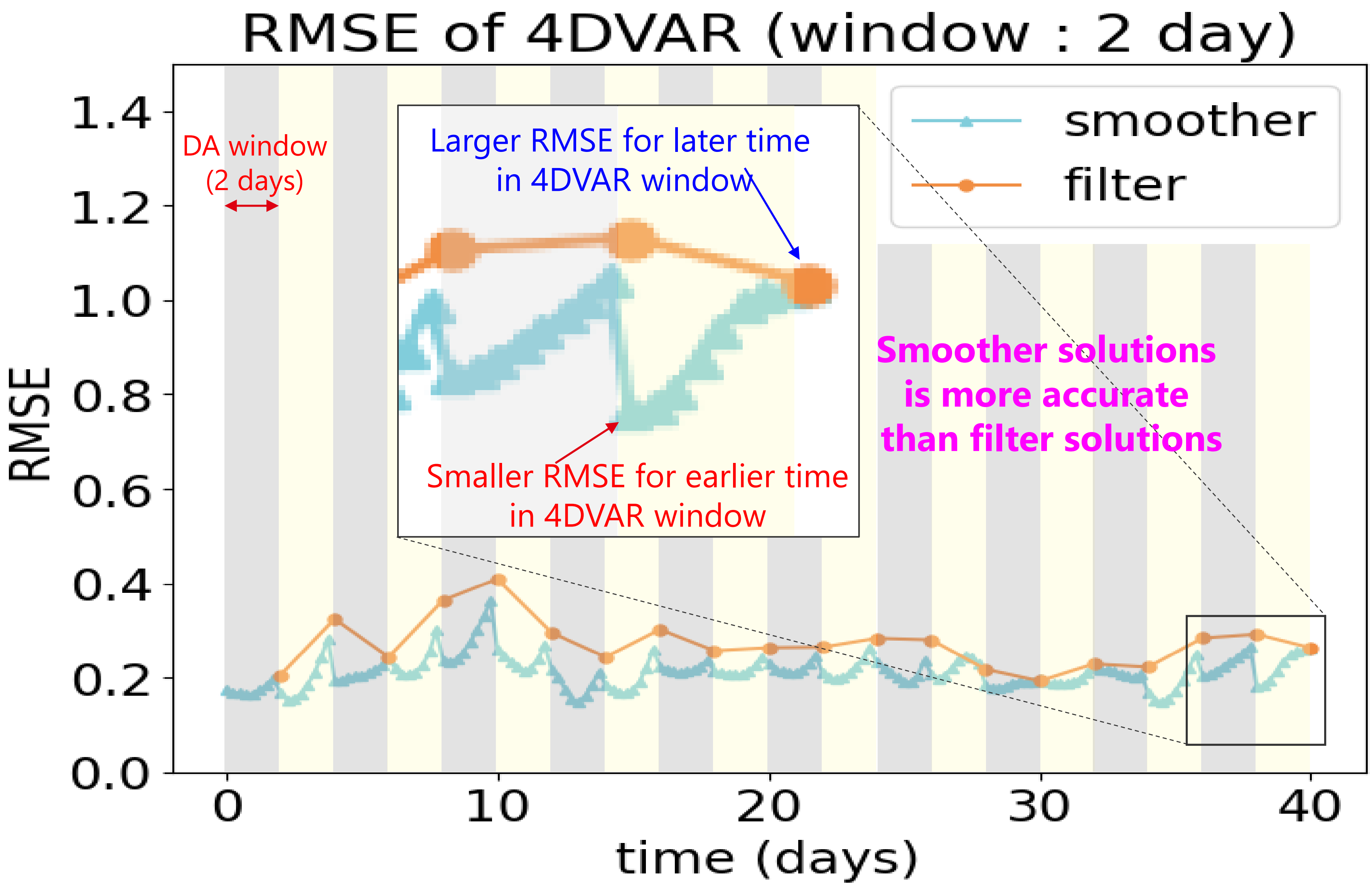




Point 2: Usually obs at t=0 is DA'ed by the last cycle

Point 3: Smoother is more accurate than filter
Analysis RMSEs of filter solutions should be compared w/ other filters (e.g. KF)

RMSEs in DA window



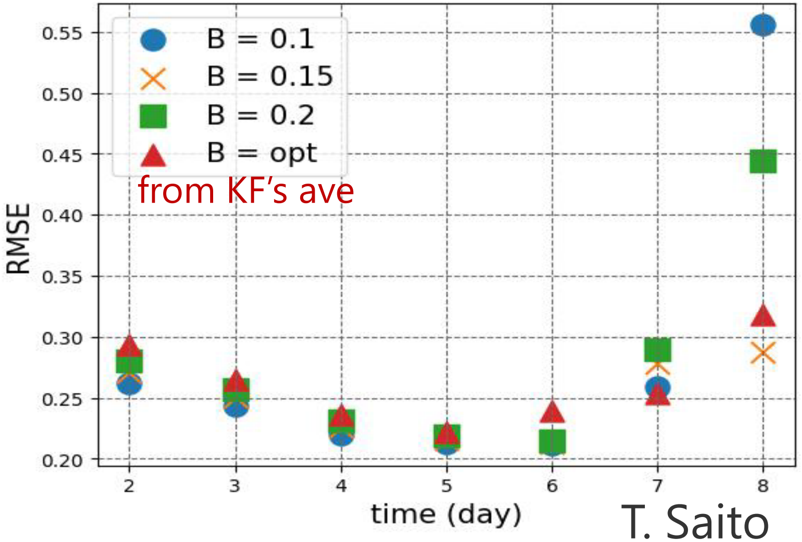
DA window: 2 days, B_{ii}=0.15, BFGS of Scipy

Sensitivity to B and window

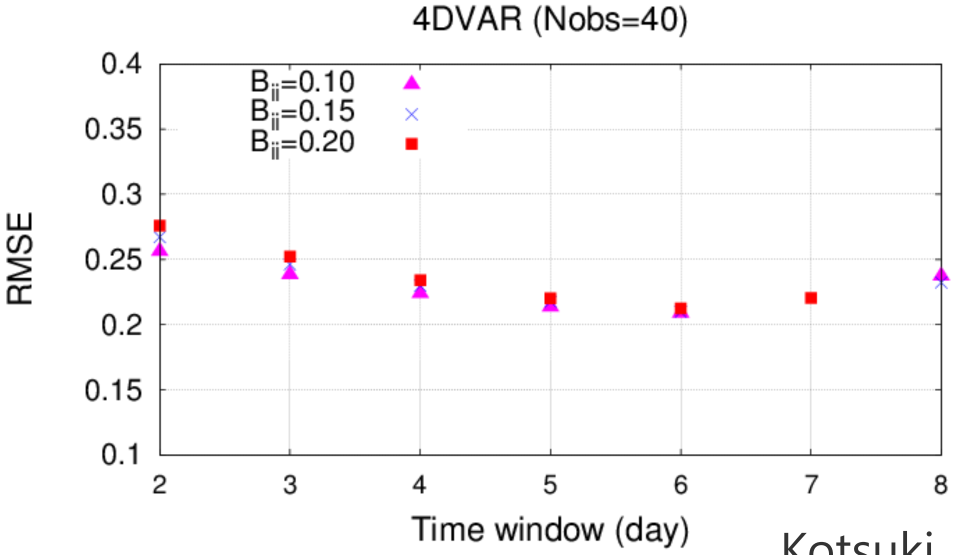
when B is diagonal matrix.

$$B = \begin{pmatrix} b_{11} & & & & \\ & \ddots & & 0 & \\ & & b_{ii} & & \\ & 0 & & \ddots & \\ & & & & b_{nn} \end{pmatrix}$$

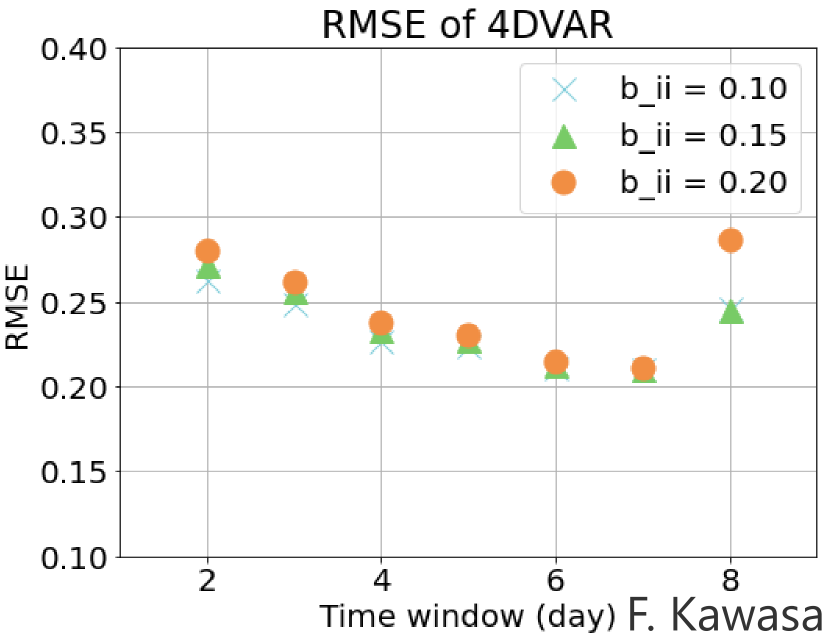
Analysis RMSE of filters are compared



T. Saito



Kotsuki



F. Kawasaki

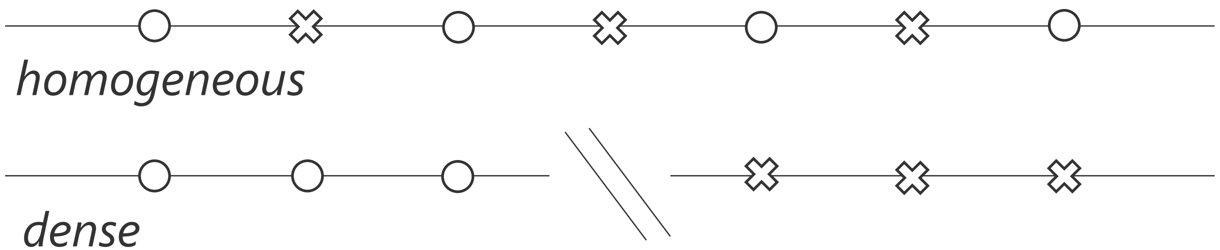
Sensitivity to Obs. Network

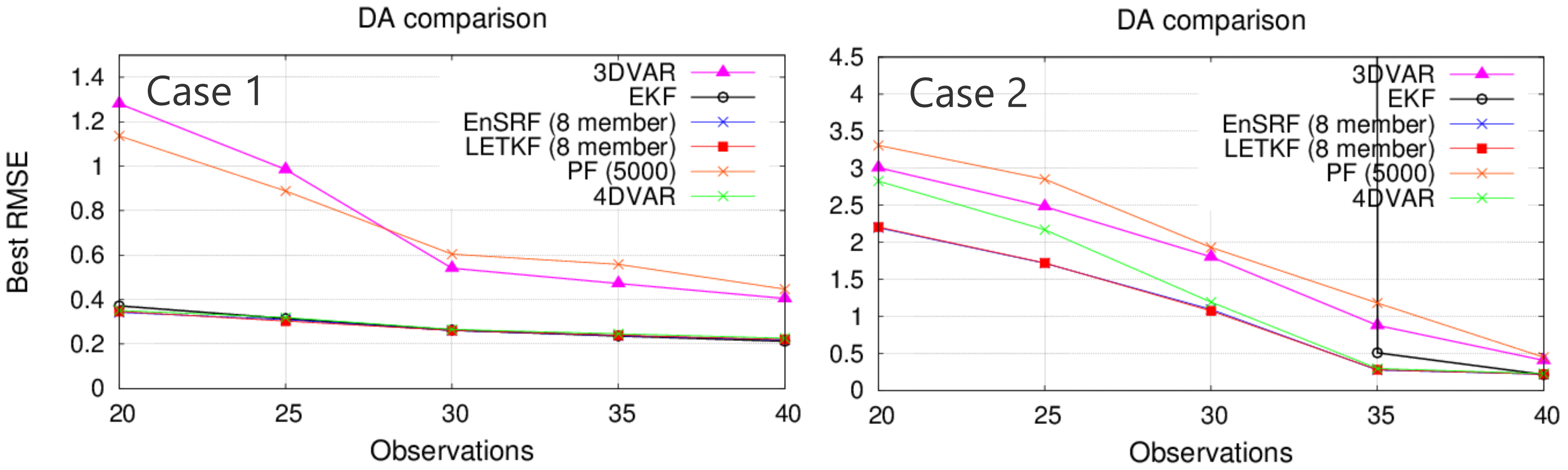
Case 1: The Num. Obs. = X

homogeneous

Case 2: The Num. Obs. = X

dense





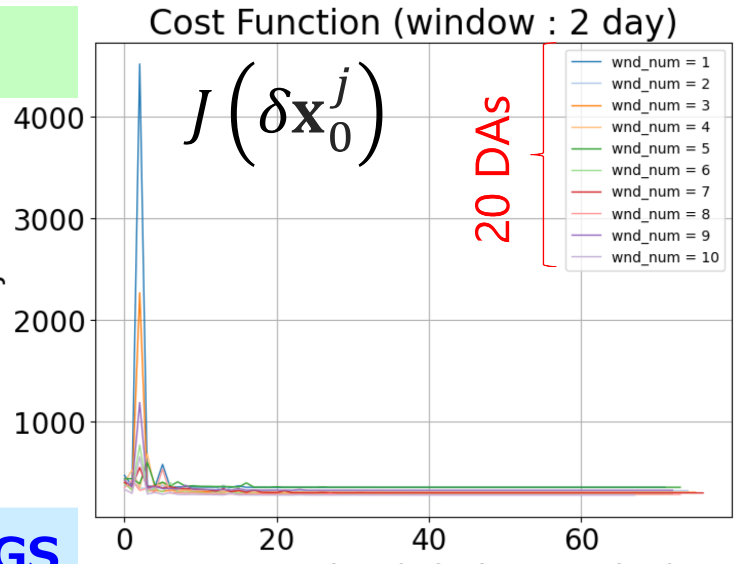
Iterations of 4DVAR (20 DAs)

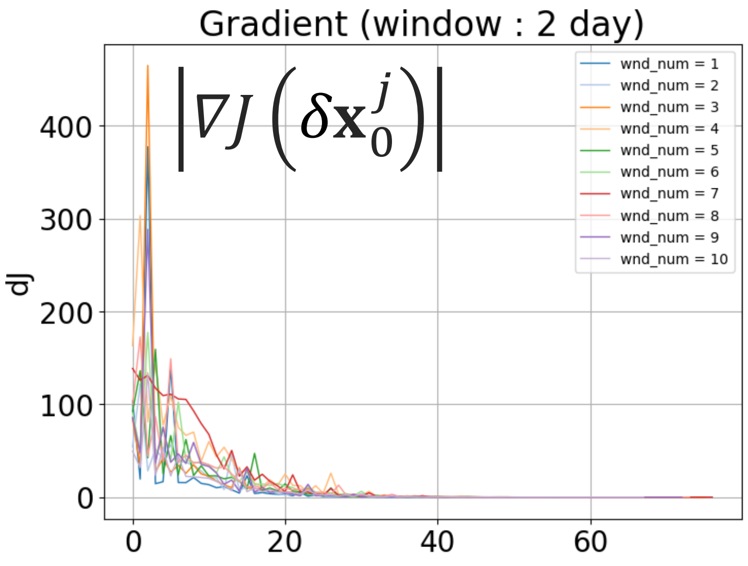
By F. Kawasaki

Environmental Prediction Science Laboratory

Scipy BFGS

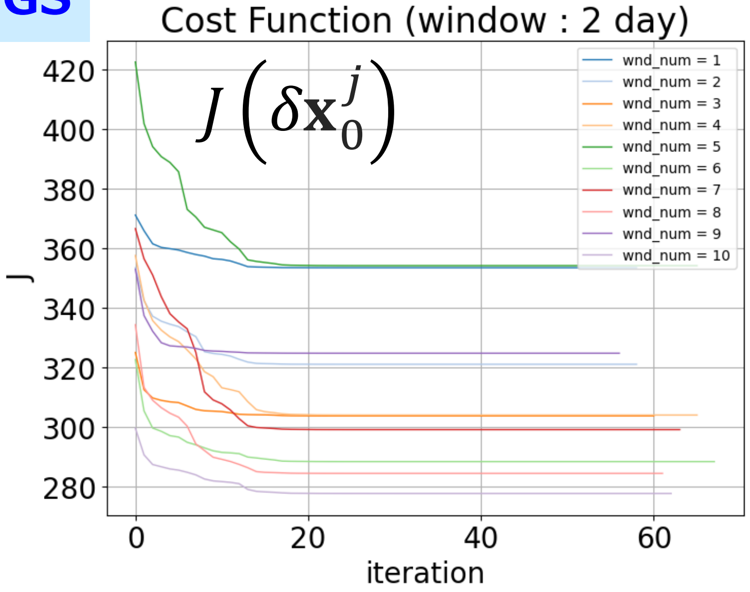
terminated when
$\left|\delta\mathbf{x}_0^{j+1} - \delta\mathbf{x}_0^{j}\right|$

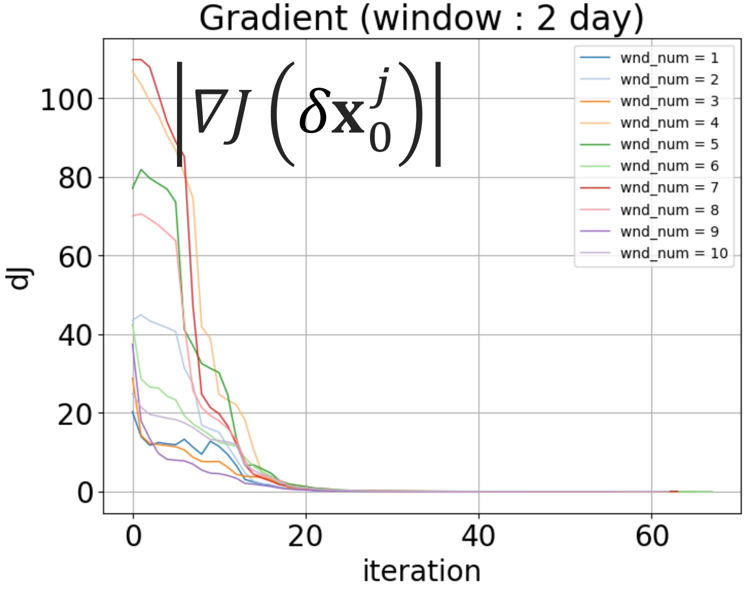




Hand-write BFGS

terminated when
$|\nabla J| < 10^{-4}$





DA window: 2 days, B_{ii}=0.15 Hand-write BFGS used Armijo ($\alpha_0^j = 1, \xi = 0.2, \tau$=0.1)

Visualization of Cost Function

F. Kawasaki

Environmental
Prediction
Science
Laboratory

Experimental Setting

- Focus on one DA by 4DVAR w/ Lorenz 96

NL Experiment

Nonlinear Model

(N1) employ L96 model fcst

(N2) departure: $\mathbf{d}_i^{o-b} = \mathbf{y}_i^o - H_i(M_{i|0}(\mathbf{x}_0))$

(N3) get TLM and ADJ models

Linearized Model

(L1) Cost and its gradient

$$J \approx \frac{1}{2}\delta\mathbf{x}_0^T\mathbf{B}_0^{-1}\delta\mathbf{x}_0 + \sum_{i=1}^{k}\frac{1}{2}\left(\mathbf{d}_i^{o-b}\right)^T\mathbf{R}_i^{-1}\mathbf{d}_i^{o-b}$$

$$\nabla J \approx \mathbf{B}_0^{-1}(\delta\mathbf{x}_0) - \sum_{i=1}^{k}\mathbf{M}_{i|0}^T\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{d}_i^{o-b}$$

(L2) BGGS to update $\delta\mathbf{x}_0$

(N4) update analysis $\mathbf{x}_0 = \mathbf{x}_0 + \delta\mathbf{x}_0$

iteration

QUO: 4DVAR

Nonlinear Model

(N1) employ L96 model fcst

(N2) departure: $\mathbf{d}_i^{o-b} = \mathbf{y}_i^o - H_i(M_{i|0}(\mathbf{x}_0))$

(N3) get TLM and ADJ models

$$\tilde{\mathbf{d}}_i^{o-b} = \mathbf{d}_i^{o-b} - \mathbf{H}_i\mathbf{M}_{i|0}\delta\mathbf{x}_0$$

Linearized Model

(L1) Cost and its gradient

$$J \approx \frac{1}{2}\delta\mathbf{x}_0^T\mathbf{B}_0^{-1}\delta\mathbf{x}_0 + \sum_{i=1}^{k}\frac{1}{2}\left(\tilde{\mathbf{d}}_i^{o-b}\right)^T\mathbf{R}_i^{-1}\tilde{\mathbf{d}}_i^{o-b}$$

$$\nabla J \approx \mathbf{B}_0^{-1}(\delta\mathbf{x}_0) - \sum_{i=1}^{k}\mathbf{M}_{i|0}^T\mathbf{H}_i^T\mathbf{R}_i^{-1}\tilde{\mathbf{d}}_i^{o-b}$$

(L2) BGGS to update $\delta\mathbf{x}_0$

iteration

(N4) update analysis $\mathbf{x}_0 = \mathbf{x}_0 + \delta\mathbf{x}_0$

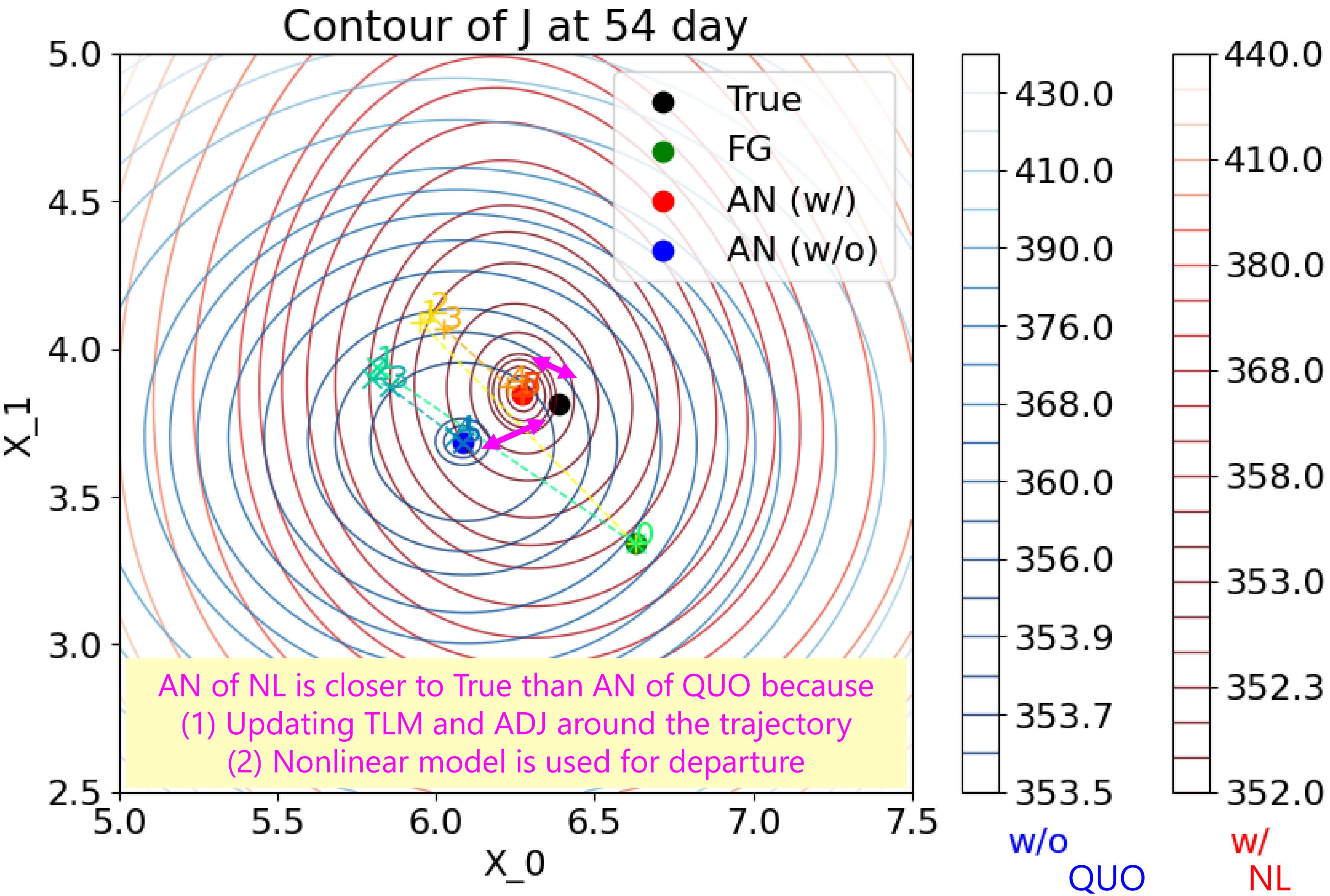
QUO: Quadratic unconstrained optimization

Estimating only two vars

- we estimate only x０ and x1 among 40 vars
  - For x2-39, truth is used for computing J and grad J
  - While analysis increment is obtained for 40 vars, only x０ and x1 are updated over the iteration

- メモ:
  - X2=x39はtruthを使って、コスト＆勾配を計算
  - Grad(J)は40変数だけど、x0とx1だけを更新

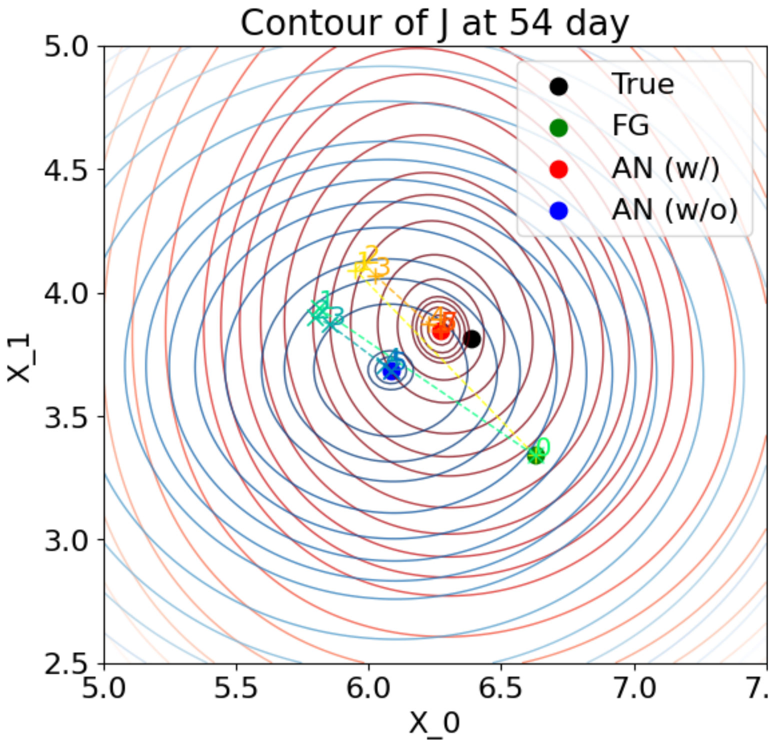
Cost function & Iteration



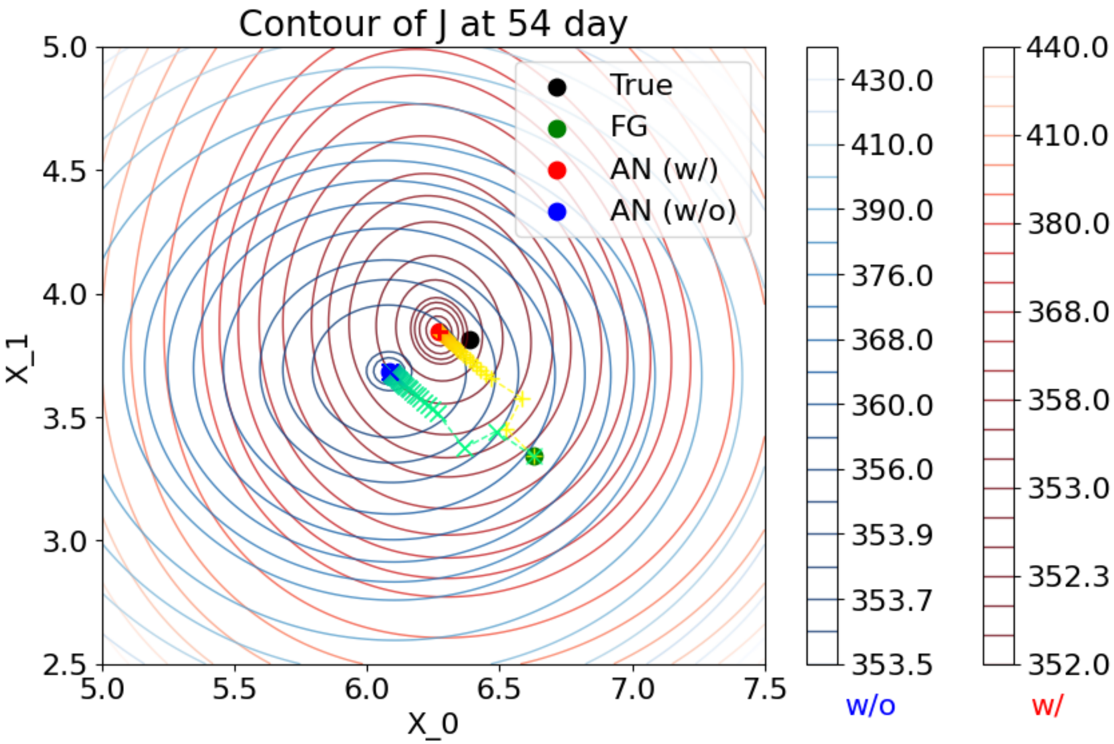
AN of NL is closer to True than AN of QUO because
(1) Updating TLM and ADJ around the trajectory
(2) Nonlinear model is used for departure

Comparison of solvers

Scipy BFGS (default)

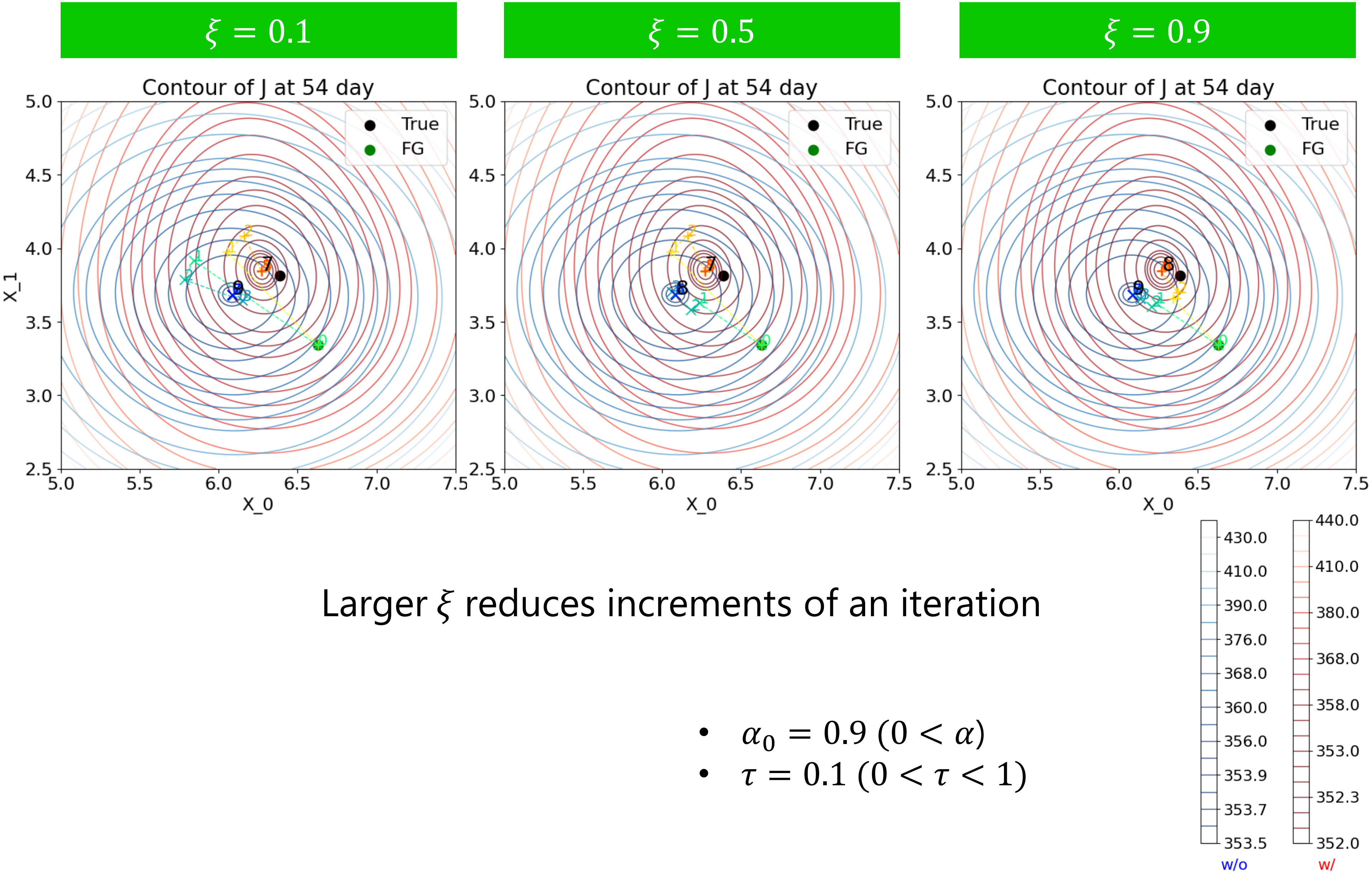


Hand-write BFGS w/ Armijo



- $\xi = 0.2 \ (0 < \xi < 1)$
- $\alpha_0 = 0.1 \ (0 < \alpha)$
- $\tau = 0.1 \ (0 < \tau < 1)$

Sensitivity to Armijo params

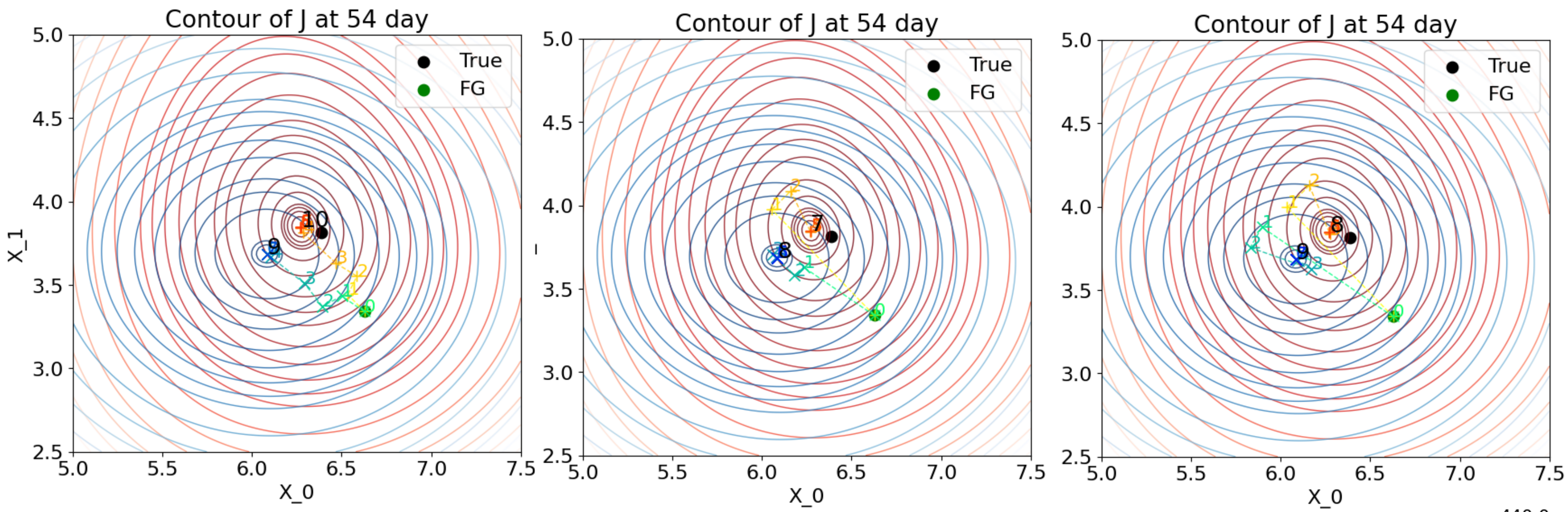


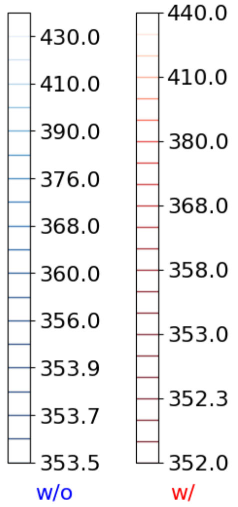
Larger ξ reduces increments of an iteration

- $\alpha_0 = 0.9\ (0 < \alpha)$
- $\tau = 0.1\ (0 < \tau < 1)$

Sensitivity to Armijo params

$\tau = 0.1$ | $\tau = 0.5$ | $\tau = 0.9$





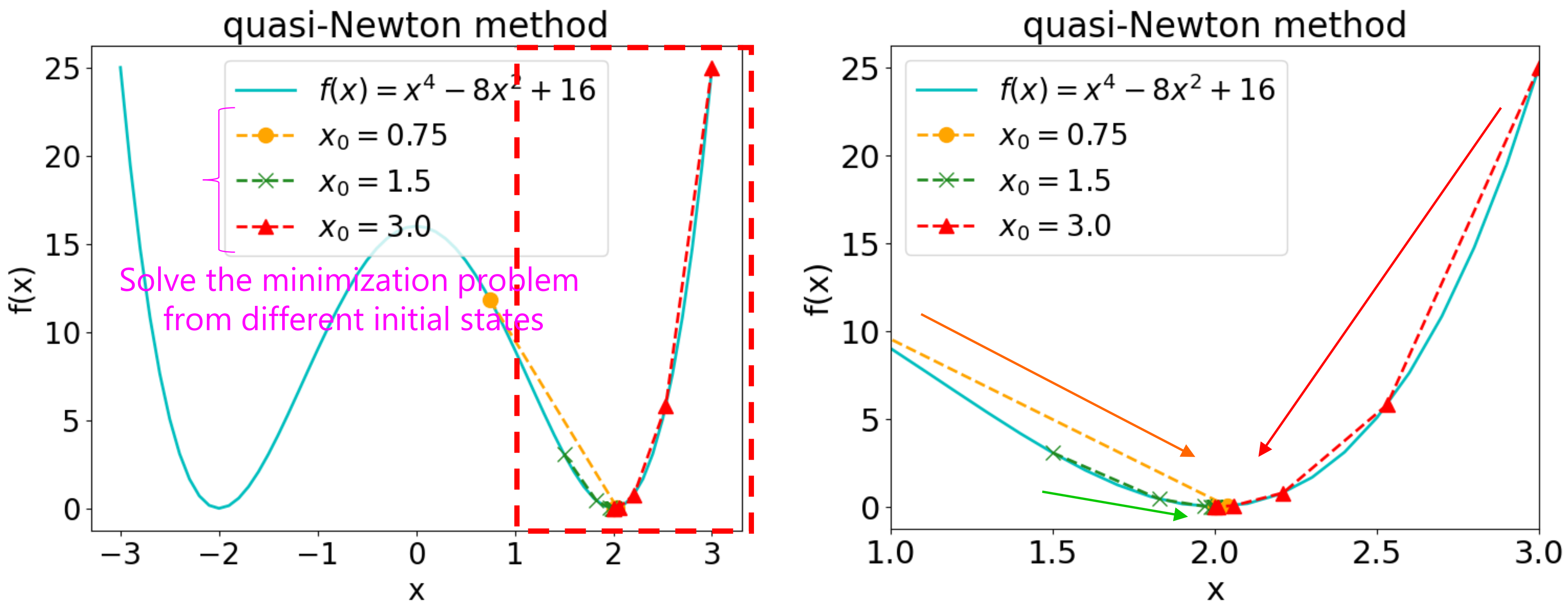
Larger τ increases increments of an iteration

- $\alpha_0 = 0.9\ (0 < \alpha)$
- $\xi = 0.5\ (0 < \xi < 1)$

Tips for implementations

For hand-write 4DVAR

- (1) to test BFGS w/ a simple problem



- (2) to test BFGS w/ 3DVAR

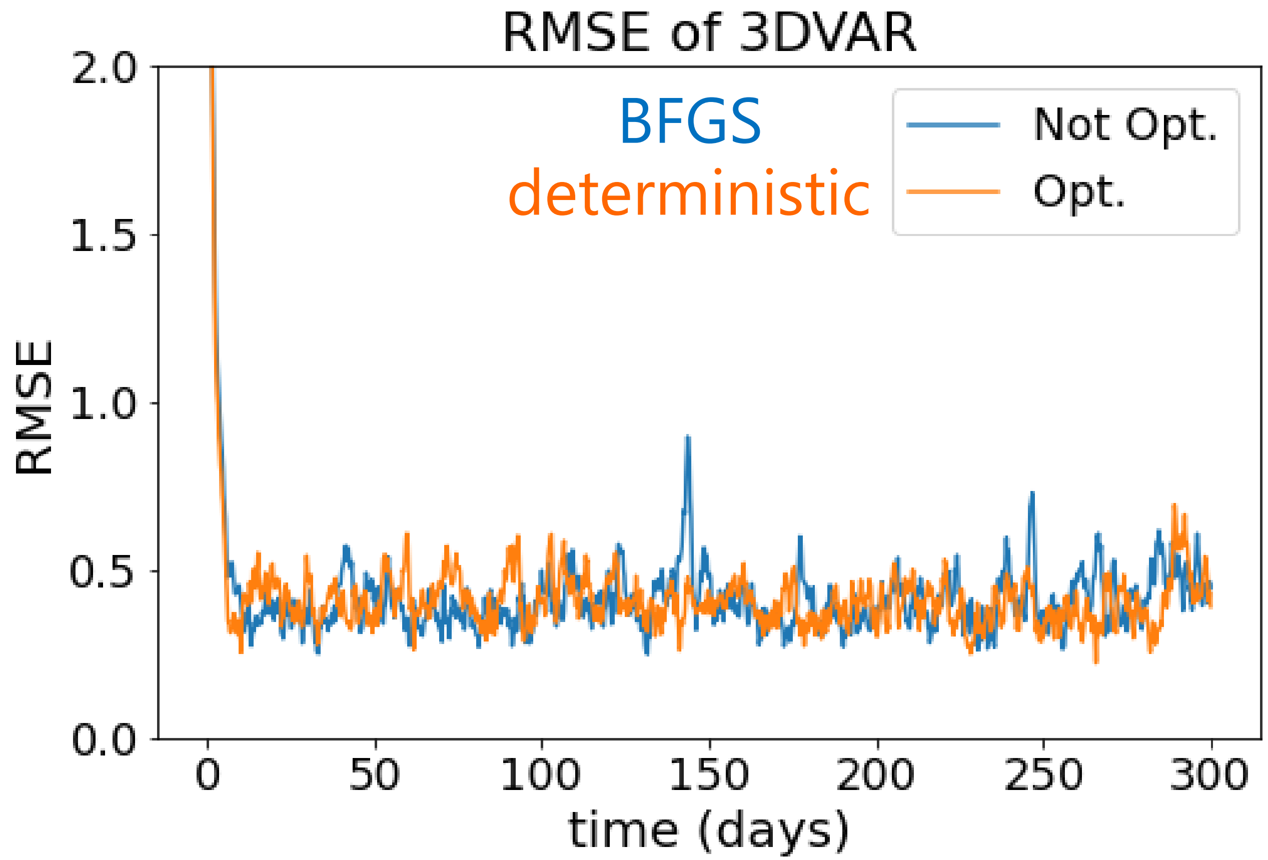
- (3) to test BFGS w/ 4DVAR

(2) BFGS for 3DVAR (L96)

3DVAR can be solved by deterministically or iteratively by BFGS

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_t^b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_t^b) + \frac{1}{2}(H(\mathbf{x}) - \mathbf{y}_t^o)^T \mathbf{R}^{-1}(H(\mathbf{x}) - \mathbf{y}_t^o)$$

$$\frac{\partial J(\mathbf{x})}{\partial x} = \mathbf{B}^{-1}\left(\mathbf{x} - \mathbf{x}_t^b\right) + \left(\frac{\partial H(\mathbf{x})}{\partial x}\right)^T \mathbf{R}^{-1}(H(\mathbf{x}) - \mathbf{y}_t^o)$$



Thank you for your attention!

Presented by Shunji Kotsuki
(shunji.kotsuki@chiba-u.jp)

Further information is available at
https://kotsuki-lab.com/

Environmental
Prediction
Science
Laboratory