Adaptive and Non-Global MMSE of Covariance for Meteorologic Data Assimilation

Alexandre Guirland Nowosad

DMD-CPTEC-INPE
Av. dos Astronautas, 1758
12201-970 São José dos Campos-SP
alex@cptec.inpe.br

Summary:

- **Introduction**
- **Development and Results**
- **Conclusions**

Introduction

- **Numerical Weather Prediction requires the initial state** w^i **of the atmosphere to be known or estimated (Kalnay, 2003; Daley, 1991; Cohn, da Silva, Guo, Sienkiewicz e Lamich, 1998).**
- **Usually this state vector is estimated using a composition of observed data and the output of a meteorological model, as solution to the minimization problem:**

$$\min_{w^i} J\left(w^i\right) = \left(w^i - w^f\right)^T \left(P^f\right)^{-1} \left(w^i - w^f\right) + \left[w^o - H\left(w^i\right)\right]^T R^{-1} \left[w^o - H\left(w^i\right)\right]$$

w^f **is given by the model,** P^f **is the covariance of the error in** w^f**,** w^o **the observation,** H **the observation function and** R **the covariance of the observation error.**

- **This work concerns fast and optimal calculation of** P^f**.**

- **Let**

$$e^f = w^f - w^r, \quad e^o = w^o - H\left(w^r\right)$$

w^r **is the <u>real</u> atmosphere (inaccessible!)**

- **Definition of P^f:**

$$P^f \equiv E\left\{\left(e^f - E\left\{e^f\right\}\right)\left(e^f - E\left\{e^f\right\}\right)^T\right\}$$

- **In practice one uses linear model for observation and:**

$$e \equiv H^+ w^o - w^f$$

(Dee and daSilva, 1999)

- **Another Problem: Computational complexity**

$$\vec{e}_j = \begin{bmatrix} \vec{e}_u & \vec{e}_v & \cdots & \vec{e}_\phi \end{bmatrix}^T_{1\times L}$$

$$P = \begin{bmatrix} \vec{e}_1\vec{e}_1^T & \vec{e}_1\vec{e}_2^T & \cdots & \vec{e}_1\vec{e}_{N_x\cdot N_y\cdot N_z}^T \\ \vec{e}_2\vec{e}_1^T & \vec{e}_2\vec{e}_2^T & \cdots & \vec{e}_2\vec{e}_{N_x\cdot N_y\cdot N_z}^T \\ \vdots & \ddots & \ddots & \vdots \\ \vec{e}_{N_x\cdot N_y\cdot N_z}\vec{e}_1^T & \vec{e}_{N_x\cdot N_y\cdot N_z}\vec{e}_2^T & \cdots & \vec{e}_{N_x\cdot N_y\cdot N_z}\vec{e}_{N_x\cdot N_y\cdot N_z}^T \end{bmatrix}$$

$$O\left(\left[N_x \cdot N_y \cdot N_z \cdot L\right]^2\right) = 10^{12} \quad !!!$$

- **Aproximation 1: Vertical covariance is** $s(z_1, z_2) = \sigma^2 \delta(z_1 - z_2)$ **and independent from horizontal covariance**

- **Aproximation 2: Homogeneous horizontal covariance**

$$P^f(z, \Delta x, \Delta y) = s(z)\rho(\Delta x, \Delta y)$$

If uses FFT results $O([N_x.\log N_x.N_y.\log N_y].N_z.L)=10^9$

Development and Results

- **Calculus of vertical variance in practice**

$$\overline{e}_{xy}(z) = \overline{e_{xy}(z)} = \frac{1}{T}\sum_{t} e_{xyzt}$$

$$s_{xy}(z) = \langle s_{xy}(z)\rangle = \frac{1}{T}\sum_{t}\left\{\left[e_{xyt}(z) - \overline{e}_{xy}(z)\right]^2\right\}$$

$$s(z) = \langle s_{xy}(z)\rangle = \frac{1}{N_x \cdot N_y - 1}\sum_{xy} s_{xy}(z)$$

- **Calculus of horizontal covariance in practice**

$$\rho_z(\Delta x, \Delta y) = \overline{\left\langle \begin{matrix} \left[e_{zt}(x+\Delta x, y+\Delta y) - \overline{e_{zt}}(x+\Delta x, y+\Delta y)\right] \times \\ \left[e_{zt}(x, y) - \overline{e_{zt}}(x, y)\right] \end{matrix} \right\rangle}$$

$$\rho(\Delta x, \Delta y) = \left\langle \frac{\rho_z(\Delta x, \Delta y)}{\rho_z(0,0)} \right\rangle$$

- **Space and time mean- and variance-ergodicity is assumed, therefore restrictions on autocovariance function of e^f apply.**

(Papoulis, 1991)

- **Parameterizations (in order of calculus):**

- **Parameterization of variance uses a polynomial of degree *m* conformal to the number of calculated variances valued at least *10 %* of the maximum of the set.** Dynamic memory allocation **is used in FORTRAN 90 to define the dimension *m* of the MMSE problem**

$$\hat{s}_{norm}(z) = \frac{1}{\hat{s}_{\max}} \hat{\sigma}^2(z) = \sum_{i=0}^{m} a_i (z/m)^i \qquad \vec{a} = \begin{bmatrix} a_0 & \cdots & a_m \end{bmatrix}^T$$

To recover variances $\hat{s}(z) = s_{\max} \cdot \hat{s}_{norm}(z)$

- **Remaining parameterizations also use MMSE discarding calculated values smaller in amplitude than *10⁻⁶***

$$\hat{\rho}_{\Delta y}(\Delta x) = \exp\left(-\alpha_{\Delta y}(\Delta x)\right)$$

$$\hat{\rho}_{\Delta x}(\Delta y) = \exp\left(-\alpha_{\Delta x}(\Delta y)\right)$$

(see Purser-Wu-Parrish-Roberts, 2003)

$$\hat{\rho}(\Delta x, \Delta y) = \hat{\rho}_x(\Delta x)\hat{\rho}_y(\Delta y)$$

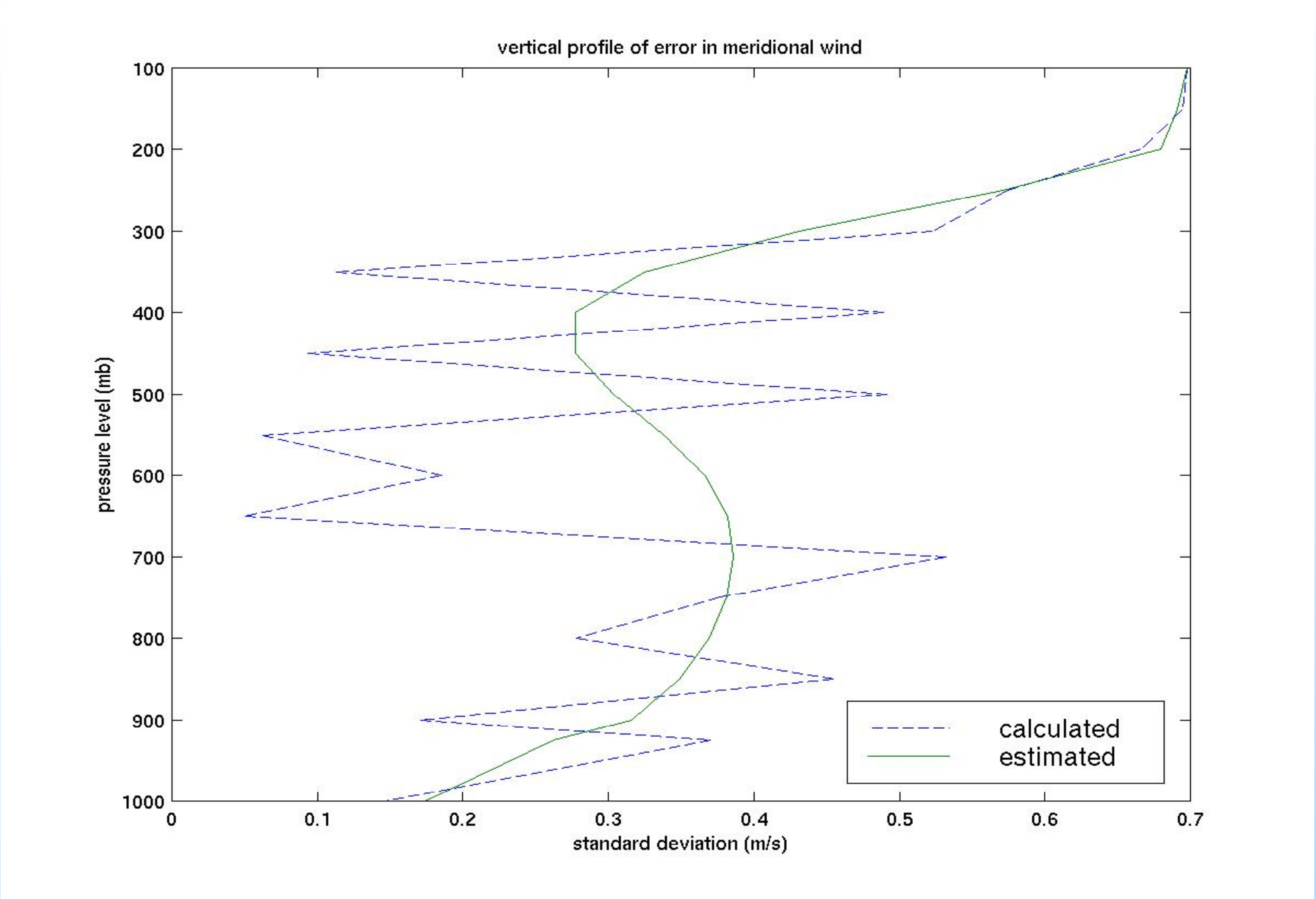

Figura 1: Vertical standard deviation.

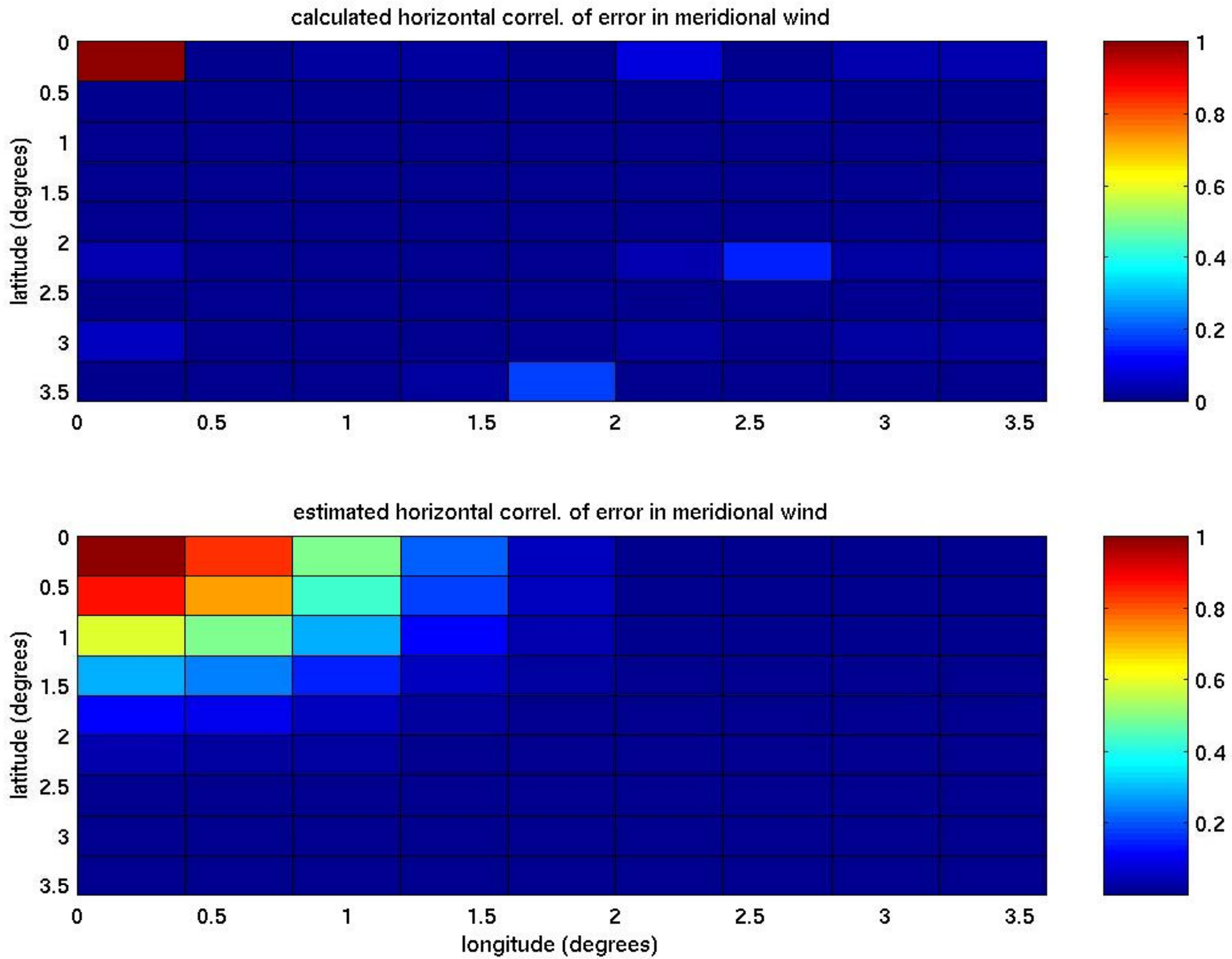

Figure 2: Horizontal covariance.

Conclusions

- Dynamic memory allocation **in FORTRAN 90 makes this software self-adaptive to available data providing smooth estimate (see Sztipanovits and Karsen, 2000).**
- **Results show estimated covariance is more realistic than calculated covariance.**
- **Computational complexity of problem is reduced.**