

Impact de non-linéarité pour la configuration actuelle
du 3d-var ALADIN

Rapport de stage

Météo France – CNRM/GMAP

Du 09 Nov. au 18 Déc. 2009

Office National de la Météorologie, Algérie
SADOKI Khaled : k.sadoki@gmail.com

Remerciements

Je tiens à remercier mon encadreur M. Ludovic Auger ainsi M. Claude Fisher. Mme Véronique, M. Escalière, M. Jean et tous ceux qui m'ont aidé de près ou de loin lors de mon stage au GMAP/CNRM.

Introduction

Ce stage a pour but d'investiguer la possibilité d'obtenir une analyse « meilleure » $X = X_g + \delta X_1 + \delta X_2$, soit parce qu'elle donne une meilleure prévision ensuite (critère le plus objectif pour évaluer la qualité d'une analyse) soit parce qu'elle est réalisée plus efficacement (moins d'itérations globalement).

L'algorithme d'assimilation du 3d-var consiste à trouver l'état initial X qui minimise la fonction coût suivante :

$$J(X) = (X - X_g)^t B^{-1} (X - X_g) + (Y - \mathcal{H}(X))^t R^{-1} (Y - \mathcal{H}(X))$$

Avec X la variable de contrôle qui regroupe un sous-ensemble des champs 3D/2D du modèle : T, Hu, Div, Vor, Ps . X_g est l'état de départ dit *guess* ou *background*, Y sont les observations, \mathcal{H} l'opérateur non-linéaire permettant de passer des variables du modèle aux observations simulées. B et R sont les matrices de variance-covariance des erreurs du modèle et des observations. Cette fonction coût peut se réécrire sous une forme différente en linéarisant l'opérateur d'observation :

$$J_1(\delta X_1) = (\delta X_1)^t B^{-1} (\delta X_1) + (I - H_g(\delta X_1))^t R^{-1} (I - H_g(\delta X_1)) \quad (1)$$

Avec $X = X_g + \delta X_1$ et $\mathcal{H}(X) = \mathcal{H}(X_g + \delta X) = \mathcal{H}(X_g) + H_g(\delta X)$ et $I = Y - \mathcal{H}(X_g)$ (innovation). Bien sur on réalise ici une linéarisation de la fonction coût initiale (à savoir de l'opérateur d'observation H_g) autour de l'état X_g supposons qu'on linéarise cette même fonction autour d'un autre état X_0 :

$$J_2(\delta X_2) = (\delta X_2 + \delta X_0)^t B^{-1} (\delta X_2 + \delta X_0) + (I_0 - H_0(\delta X_2))^t R^{-1} (I_0 - H_0(\delta X_2)) \quad (2)$$

Avec $\delta X_2 = X - X_0$, $\delta X_0 = X_0 - X_g$, $I_0 = Y - \mathcal{H}(X_0)$, H_0 la linéarisation de \mathcal{H} autour de X_0 une version approchée de cette fonction coût est :

$$J_2'(\delta X_2) = (\delta X_2)^t B'^{-1} (\delta X_2) + (I_0 - H_0(\delta X_2))^t R^{-1} (I_0 - H_0(\delta X_2)) \quad (3)$$

Le moyen pour y parvenir est de tester la réalisation dans ALADIN 3d-var de l'adjoint d'une 2^{ème} étape de la minimisation de $J(X)$: après avoir trouvé un minimum δX_1 et $J(X_1)$ on veut relinéariser $J(X)$ et minimiser cette fonction autour de $X_0 = X_g + \delta X_1$ et ainsi obtenir une valeur δX_2 pour laquelle un minimum de $J_2(X)$ est atteint. A cet effet on testera les deux formulations :

$J_2(\delta X_2)$ et $J_2'(\delta X_2)$

Etapes de déroulement

Après avoir pris en main de l'outil OLIVE de gestion d'expériences ALADIN/France, on a pu arriver à faire:

- création d'une première expérience modifiée en utilisant la formulation (3),
Analyse les listings de retour d'expérience de la tâche minimisation
- création d'une deuxième expérience modifiée en utilisant la formulation (2),
Analyse les listings de retour d'expérience de la tâche minimisation

1^{ère} expérience

On effectue uniquement la minimisation de la fonction J_1 pour obtenir l'état analysé $X_g + \delta X_1$, c'est le 3d-var actuel

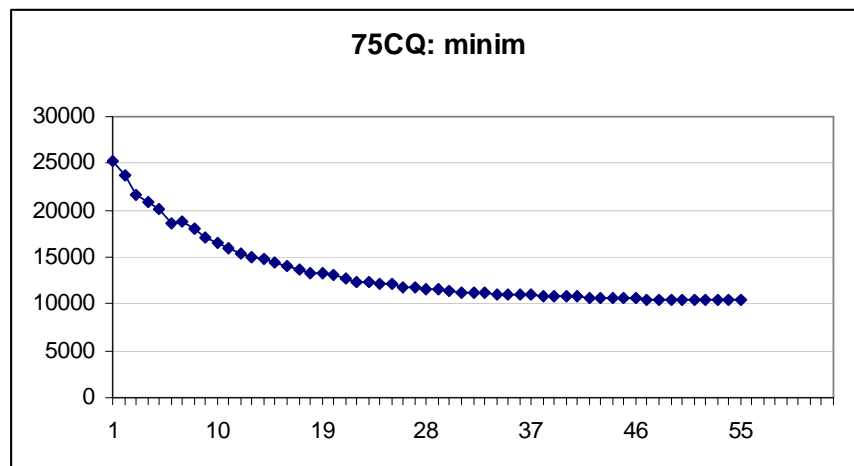


Fig.1 : Tracé du J_0 d'une simple minimisation

2^{ème} expérience

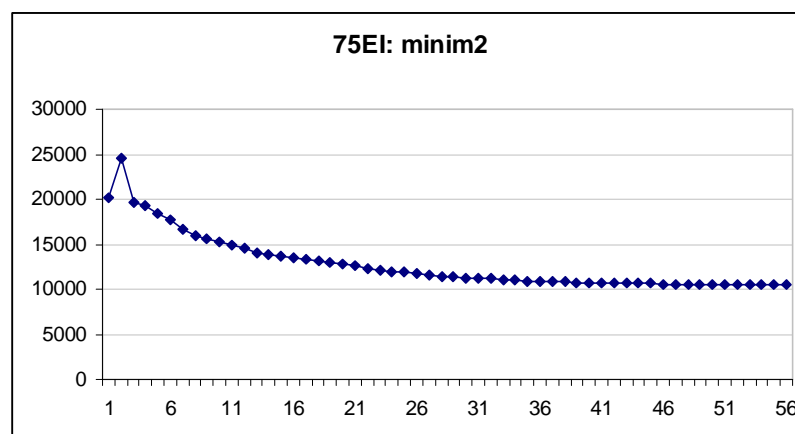


Fig.2 : Tracé du J_0 de la 2^{ème} minimisation

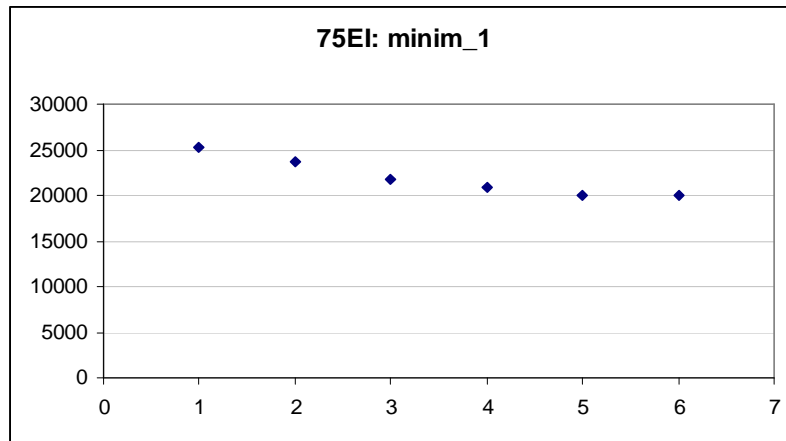


Fig.3 : Tracé du J_0 de la 1^{ère} minimisation
Avec 5 itérations

3^{ème} expérience

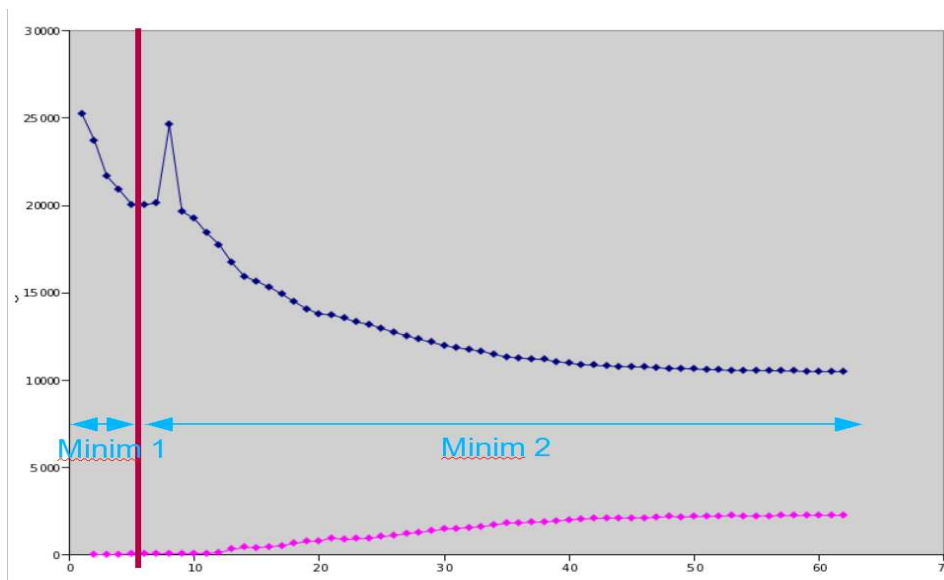


Fig. : Tracé J_0 et J_B

Remarques et conclusion

En faisant les diagnostics des sorties de 3d-var ALADIN/France en utilisant GREPCOST pour obtenir J_0 et J_b pour chaque expérience/minimisations, figures ci-dessus. Pour la 1^{er} expérience (Fig. 1), le tracé montre le terme J_0 qui diminue fortement dans les dix premières itérations, puis lentement pour les 20 itérations suivantes puis très faible minimisation malgré le nombre d'itérations. En faisant une 2^{ème} minimisation autour de X_g , le tracé (Fig. 2) montre que le nouveau minimum n'est pas assez différent par rapport au premier, même résultat constaté avec le changement du nombre d'itérations. Il est intéressant de rappeler que d'autres résultats pouvaient être éventuellement tirés avec la reconduction des expériences précédentes sur une période longue (1, 2 semaines) et on les analyse par le biais des diagnostics appropriés tels que les outils d'analyse de base ODB (mandaodb)

Annexes

Les expériences ont été exécutés à partir l'interface OLIVE/SWAPP

Interface OLIVE

The screenshot shows the OLIVE interface in a Mozilla Firefox browser. The interface is divided into several sections:

- Left Sidebar:** Contains 'Olive::Task' with a bug icon, 'Action [Browse]' (Edit, Configure), 'SMS Commands' (update, OK), and a 'Sequence' list including tasks like 'Increment from previous minim', 'Errgribvor', 'NameIst IAS1', etc.
- Main Content Area:** Displays configuration for 'minim'.
 - Variables:** NPROC: 1, PROFILE: size:20000 cpu:1500 wrap:ko, CLASS: minia_2.
 - SMS functions:** trigger: screening==complete, flag: "varbc_\$CUTOFF".
 - Sequence:** 'Increment from previous minim' with format: unknown and local: traj00+0000000. Below it, 'Errgribvor' with format: unknown and local: errgrib.
- Right Panel:** A file explorer showing a directory tree for '75EI'. It includes sub-directories like 'YYYYMMDD', 'HH', 'CUTOFF', and various task families like 'fc_init', 'coupling', 'forecast', 'fullpos', 'obsprep', 'pregpssol', 'batodb', 'merge', 'merge_varbc', 'surfan', 'extract_surf', 'c931', 'canari', 'analyse', 'screening', 'obscma', 'minim', 'pseudotraj', 'matchup', 'analyse_2', 'screening', 'minim', 'pseudotraj', 'cplinit', 'coupling', and 'dfi'.

The screenshot shows the XCDP interface displaying a task execution graph. The graph is organized into a tree structure with nodes representing tasks and their dependencies. The tasks are color-coded (yellow and red) and include job IDs. The graph shows a sequence of tasks starting from 'assim' and 'obsprep', leading to 'surfan', 'analyse', 'cplinit', 'dfi', 'fc_I', and 'fc_J'. This is followed by a large block of tasks including 'pregpssol', 'batodb', 'merge', 'merge_varbc', 'surfan', 'batodb', 'canari', 'analyse', 'screening', 'obscma', 'minim', 'pseudotraj', 'matchup', 'analyse_2', 'screening', 'minim', 'pseudotraj', 'cplinit', 'coupling', 'bias_idf', 'inc_idf', 'forecast', and 'fullpos'. Each task node is associated with a job ID, such as 'jobid: 0.3942 yuki-backup.mstep_000333'.