

TD SEQUOIA
Janvier 2010
P. De Mey

Notes préliminaires:

- Pas de panique: on ne passera pas à l'étape suivante tant que tout le monde n'a pas fini l'étape en cours!
- Désolé d'avance du caractère peu graphique, et pas toujours océanographique, du TD. L'idée de base est de rester le plus simple possible.
- Pour la prochaine fois (2011?), on propose de se concentrer davantage sur l'interface avec SYMPHONIE, qui devrait être plus présentable à ce moment-là, et sur les différences apportées par la version 1.3sirocco2.
- Par défaut le répertoire courant n'est peut-être pas inclus dans le PATH sur votre machine. Ceci empêche certains Makefiles de fonctionner correctement. Le remède est simple: en cas de problème, dans chaque nouveau shell, entrer: PATH=.:\$PATH
- Pour avoir l'arborescence à gauche dans Dolphin: Affichage / Panneau / Dossiers.

A. TELECHARGEMENT, INSTALLATION ET CONFIGURATION DE SEQUOIA

A1. Afficher <http://sirocco.omp.obs-mip.fr> dans un navigateur, télécharger sequoia-sirocco1.tgz dans le répertoire Documents/ et le décompresser (demander le mot de passe à Pierre ou un membre du POC)

A2. Afficher le Users Guide HTML dans un navigateur, et suivre les instructions d'installation (3. INDIVIDUAL TARBALLS)
(Note: pour décompresser d'un coup tous les tarballs dans components/, faire "make unpackall")

A3. Suivre les instructions du Users Guide pour configurer et compiler SEQUOIA avec les options suivantes:

- Noyau d'analyse: beluga
- Bibliothèque utilisateur: ulib1 ("coquille vide")
- Version identique pour tous les modules: 1.2.1sirocco1
- Compilateur: ifort avec optimisation (-O)
- Bibliothèques BLAS et LAPACK: laisser par défaut

B. EXPLORATION DU CODE , PARAMETRES, FONCTIONS DE BASE

B1. Faire "cd work; beluga test-beluga.prm" et examiner la sortie standard. Recommencer avec la bonne option pour un fonctionnement scalaire (hors ensemble). Dans la suite de B., on va regarder où les paramètres listés dans la sortie standard sont définis.
(Noter une petite déviation par rapport à la rubrique Command Line du Users Guide: désolé, ça sera réparé dans la prochaine version... mais de toute façon le mode hors ensemble ne sert pas à grand chose avec ce noyau)

B2. Examen du fichier de paramètres, interprétation à l'aide du Users Guide (section Parameter File)

B3. Examen de la structure de l'arborescence et de modules particuliers (voir aussi le Users Guide,

section Parameter Modules):

- Noyau d'analyse (lien akern/, examen de ak_params_mod.f)
- Bibliothèque noyau partagée (lien shak/, examen de ak_event.f, ak_event_mod.f, ak_kdtree_lib.f, ak_obs_mod.f)
- Séquenceur (lien sap/, examen de sap_enkf_drv.f)

B4. Examen de la bibliothèque utilisateur

- Affichage du fichier PDF doc/sequoia-sirocco-jan2010.pdf : slides 3, 4, 12
- Users Guide : section SEQUOIA/model coupling reference
- Lien ulib/, examen de u_ingridFD.f, u_nullmodel.f, u_io_ak_prm.f, u_io_data.f, u_launch.f, u_setstats.f, u_setup_mod.f, u_stats_lib.f

B5. L'interface avec SYMPHONIE: présent et futur proche

- Interface avec SYMPHONIE 2007 disponible en mode « recherche » et encore en chantier... mis en place lors de plusieurs thèses au POC. Contact actuel principal: Julien Lamouroux (via Pierre ou Nadia). Fonctionne bien, encore utilisée assez couramment, mais version plus présentable dispo vers avril, avant ou avec la version 1.3sirocco2 de SEQUOIA.
- cd components/sequoia-beluga-ulib-symphonie-1.2.1sirocco1
- Examen rapide de u_interface_symphonie.f, u_correct_symphonie.f, u_ingrid_FD.f, u_xfyf.f .

B6. Un avant-goût d'une exécution d'ensemble!

- Dans work/ , créer un fichier nommé « mlist », contenant deux lignes:

1
2

- Ouvrir un second shell, aller dans work/
- Dans le premier shell, exécuter beluga en mode maître – noter l'attente de « quelque chose »...
- Dans le second shell, examiner le contenu du fichier « runinfo.0 ». Puis taper « touch abort » – noter le résultat dans le premier shell. Réexaminer le contenu de « runinfo.0 ».
- Dans le premier shell, effacer le fichier « abort », exécuter beluga en mode « esclave n°1 ». que se passe-t-il? Noter ensuite le contenu de « runinfo.1 ».
- Examiner shak/aseya_lib.f , et en particulier les time-outs du début et la fonction aseya_signal()
- Affichage du fichier PDF doc/sequoia-sirocco-jan2010.pdf : slide 11
- Examiner la définition des états ASEYA au début de sap/sap_enkf_drv.f

C. UN EXEMPLE D'ASSIMILATION DANS UNE DYNAMIQUE FORTEMENT NON-LINEAIRE : RESOLUTION DE L'OSCILLATEUR DE LORENZ PAR FILTRAGE DE KALMAN D'ENSEMBLE

C1. A la racine de l'arborescence SEQUOIA, faire « make cleanall ». Modifier sequoia.conf pour utiliser la bibliothèque utilisateur « ulib-lorenz ». Comme après tout changement de configuration, faire « make config ». Compiler.

C2. « cd ulib ». Noter que le contenu a changé. Examen de u_model-lorenz.f, u_ingrid.f .

C3. « cd work ». Nous allons faire tourner le filtre de Kalman d'ensemble avec 8 membres + 1 processus maître.

- Affichage du fichier PDF doc/sequoia-sirocco-jan2010.pdf : slide 9

- Rappel des objectifs: estimation en ligne des covariances d'erreur et de leur évolution dynamique.
- Les fichiers README contiennent les caractéristiques des simulations.
- Affichage du fichier PDF doc/sequoia-sirocco-jan2010.pdf : slide 11
- Exécuter une simulation d'ensemble de référence, sans assimilation: « runens lorenz 1 »
 - Prend quelques minutes
 - Observer les sorties du processus maître, qui est à l'avant-plan, et qui synchronise les membres
 - Observer le nombre de processus, tous identiques, en mémoire
 - Quand c'est fini on peut comparer les fichiers de référence du processus maître (terminés par un « -- ») aux fichiers fraîchement créés par un « diff »: log (stdout), niv (norme de l'innovation), tra (trajectoire du modèle) – aucune chance que ce soit strictement identique cependant, il s'agit d'un oscillateur dit « chaotique »!
- Exécuter de la même manière une simulation d'ensemble de référence, avec assimilation: « runens lorenz 2 »
- Lancer les scripts gnuplot (*.gnu), interpréter les résultats et comparer aux sorties de référence.

C4. Modifications pour l'exécution sur cluster (grappe) de Pcs

- Examen du script « ulib/work/runens » pour exécution sur multicore
- Examen du script « doc/runens_clust » pour exécution sur cluster
 - Avantage d'une mise à jour synchrone du cache NFS dans la configuration actuelle
 - Padding à 512 octets de l'aseyafile si nécessaire (aseya_signal())
 - La version 1.3 et certaines versions 1.2.x comportent une paramétrisation du répertoire « commun » de SEQUOIA → possibilité de lancer l'exécutable SEQUOIA dans un répertoire spécifique au noeud ou à SYMPHONIE