

4 NEMOTAM Meeting 24/5/2007

A meeting was held in LOCEAN the 24th of may, were present: Arthur Vidard (INRIA-Moise), Laurent Hascoet (INRIA-Tropics), David Ferreira (MIT), Rachid Benshila (LOCEAN), Charles Deltel (LOCEAN), Gurvan Madec (LOCEAN), Claude Talandier (LOCEAN)

Agenda

- Description du fonctionnement de NEMO (gestion des versions, etc.)
- Description du fonctionnement de TAPENADE
- Choix à court terme d'une version de démonstration à valeur pédagogique (tutorial)
- Prospectif : quelle stratégie pour faire évoluer le linéaire tangent et l'adjoint en même temps que le modèle direct?
- ANR calcul scientifique (prochain AO jan/févr prochain), pour obtenir un post-doc ou un IR (3ans) à Grenoble

4.1 Status of Tangent and Adjoint Models

4.1.1 Current Status

Le modèle direct utilisé actuellement est une version de NEMO, antérieure à la release `nemo_v2`, dans la configuration de référence ORCA2 (sans glace de mer). La physique utilisée pour le mélange vertical est simplifiée, car le schéma TKE n'est pas différentiable (la clef `key_zdfcst` est activée). Les codes linéaire tangent (TL) et adjoint (ADJ) ont pu être générés en utilisant le dérivateur automatique TAPENADE. Les deux modèles générés ont été validés. Pour le modèle ADJ, deux validations sont effectuées : le test du produit scalaire et la comparaison d'un gradient obtenu par différences finies au gradient fourni par l'adjoint (jusqu'à la 14ème décimale). Le coût calcul de l'ADJ est 6.8 fois celui du modèle direct. Ce ratio est raisonnable, étant donné qu'il peut difficilement être inférieur à 4 (dérivation des produits).

4.1.2 Code details

Quelques modifications mineures, relatives aux routines `flinjet/flinput` de IOIPSL (fermeture systématique de fichiers) ont été nécessaires dans le code. Ces modifications devraient être superflues (ou différentes) à partir de `nemo_v2`, avec l'utilisation de IOM (Input/Output Manager).

Les clefs CPP utilisées sont les suivantes : `key_partial_steps` `key_trabbl_dif` `key_lim_fdd` `key_dynspgflt` `key_traldf_c2d` `key_dynldf_c3d` `key_dtatem` `key_dtasal` `key_tau_monthly` `key_flx_forced_daily` `key_tradmp` `key_trabbc` `key_zdfcst` `key_orca_r2`

4.2 Providing it to the users

Une possibilité est d'avoir deux approches pour NEMOTAM: une première purement Tapenade, dans la continuité de ce qui a été fait jusqu'à présent, où NEMO est fourni avec Tapenade (éventuellement un peu adapté) et des instructions pour obtenir TL et ADJ. La seconde approche est de développer un code TL et ADJ en appliquant Tapenade routine par routine et en raffinant à la main. Les deux pistes :

- "Full TAPENADE": on ne peut pas envisager de fournir directement les codes TL et ADJ, car ils ne peuvent être générés qu'après le pré-processing CPP, et dépendent donc de la physique choisie par l'utilisateur. TAPENADE est écrit en JAVA (compilé). Il ne s'agit pas d'un logiciel libre, mais il peut être mis gratuitement à la disposition de la communauté recherche (il faudra probablement faire signer aux utilisateurs un engagement d'utilisation dans ce sens, non commerciale, qui pourrait être adossé au texte de NEMO??) .
- "Mix TAPENADE-Handwriting" : l'application directe de Tapenade est malheureusement difficilement compatible avec la formulation incrémentale. Cette incompatibilité est due à aux stratégies utilisées pour gérer la trajectoire autour de laquelle les modèles sont linéarisés :
 - Pour TL: le modèle non linéaire est intégré simultanément au TL. Or en 4D-Var incrémental on effectue plusieurs réalisations du TL (avec des perturbations de la CI différentes) autour de la même trajectoire du modèle non linéaire.
 - Pour ADJ: Tapenade utilise la stratégie de checkpointing pour stocker la trajectoire directe (on sauve quelques états et on utilise le modèle non linéaire pour recalculer la trajectoire entre les checkpoints). Ceci rend très difficile l'utilisation de résolutions différentes entre le modèle non linéaire et les TL et ADJ (approche multi-incrémentale). L'approche incrémentale est pourtant souhaitable quand on adopte une approche "physique simplifiée" (cela permet de garder la physique complète lors du calcul de l'écart aux observations). Ici la gestion de la trajectoire du modèle non linéaire serait faite "à la OPAVAR".

La seconde approche, qui n'est pas exclusive de la première, requiert évidemment plus de travail mais permet plus de flexibilité et peut fournir des TL et ADJ avant pré-processing CPP. À terme, si la force de travail est disponible, on peut espérer fournir cette deuxième version de TL et ADJ avec chaque release de Nemo. Même dans cette éventualité il serait cependant souhaitable de garder l'approche "full Tapenade" pour les utilisateurs qui voudraient 'ajouter' leur version modifiée de NEMO sans mettre la main à la pâte.

4.2.1 Maintenance and user support

Il faudrait auparavant répondre aux questions suivantes : faut-il choisir une physique simplifiée (autre que de ne pas activer TKE)? quels sont les blocs de code auto-adjoints (au moins le solveur Red/Black SOR)

et ajouter à TAPENADE la fonctionnalité de générer automatiquement des routines de test des codes TL et ADJ. Ces tests, comme pour l'interface intégré par A. Weaver aux codes TL et ADJ codés manuellement, seraient les suivants : test du TL : calcul comme fonction du temps, de $\| M(x + dx) - M(x) \|$ et $\| M'.dx \|$ test du produit scalaire pour valider ADJ test du gradient pour valider l'ensemble de NEMO-TAM

Une première version "full Tapenade" pourrait être disponible pour fin octobre selon les disponibilités. **C'est pas un peu trop tôt ?** La nécessité pour les utilisateurs d'exécuter TAPENADE à chaque nouveau choix de clefs CPP a comme implications : le fonctionnement de TAPENADE (binaire) sur les différentes plates-formes la modification des Makefile de NEMO pour ce pré-processing

D'autre part, à chaque évolution de NEMO, il faudra s'assurer que TAPENADE s'exécute correctement sur le nouveau code, et il faudra valider les modèles TL et ADJ obtenus (pour la configuration ORCA2 de référence avec physique simplifiée).

4.2.2 future developments

- optimisation vectorielle (ou pas) TL et ADJ : L'équipe ESOPA regardera les optimisations possibles du code produit par Tapenade. Ces optimisations pourront être envoyés à l'équipe TROPICS afin que Tapenade génère directement le code optimisé lorsque cela est possible.
- parallélisation MPI de TL et ADJ : pas de difficulté de principe (lib_mpp auto-adjoint)
- Proposition ANR incluant une demande de PostDoc ou d'IR pour la version hybride.

4.3 Demo version

Une première version de démonstration est visée pour fin octobre. Elle a pour but plus de visibilité autour de TAM, et devrait servir pour Hicham à la rédaction d'un papier d'ici la fin de son post-doc. Ce travail utiliserait dans un premier temps la version de NEMO actuellement testée à Sophia.

4.3.1 Stationary solution for ORCA2

finalement Hicham n'a pas eu le temps de s'attaquer à ça, on peut quand même le garder sous le coude On chercherait à optimiser principalement les forçages pour rechercher une solution stationnaire du modèle, proche de la climatologie Levitus annuelle, et forcé par une climatologie annuelle. Le contexte du modèle direct serait donc : d'utiliser la climatologie Levitus moyennée sur une année comme condition initiale, de forcer le modèle avec des vents et flux air-mer constants (il faut moyenner les 12 mois des fichiers flx.nc, taux_1m.nc, tauy_1m.nc). Cela nécessite d'adapter les routines flxmod.F90 et taumod.F90.

La fonction coût J_{obs} d'écart aux observations la plus simple consisterait à pénaliser les déviations de l'état moyen, par un écart aux moindres carrés entre l'état final du modèle (on pourrait aussi ajouter un état intermédiaire) et la climatologie annuelle, pondéré par des statistiques à définir.

On pourrait lui ajouter un terme d'écart à l'ébauche `J_bck`, constitué : d'un écart quadratique aux forçages (les variables qu'on cherche à optimiser, on peut se contenter du vent pour commencer) d'un écart quadratique aux conditions initiales (cela permettrait de laisser un petit degré de liberté d'ajustement aux conditions initiales)

Les variables que l'on chercherait à optimiser seraient les forçages : vent (taux, `tauy`), flux de chaleur, évaporation-précipitation. Dans une première étape, on pourrait se contenter de contrôler le vent. Par la suite, on pourrait aussi laisser les conditions initiales s'ajuster un peu, car un état même stationnaire doit être compatible avec la dynamique du modèle, ce qui n'est vraisemblablement pas le cas d'une climatologie moyenne.

4.3.2 Possible bibliography

Tziperman et Thacker (1992), traitant du problème de la recherche d'une solution océanique stationnaire. Il y a aussi Wunsch (1978) : The north Atlantic general circulation west of 50°W determined by inverse methods. *Rev. Geophys. Space phys.*, 16, 583-620. Il faut néanmoins faire une étude bibliographique plus poussée.

4.3.3 Minimisation algorithm

il n'y a pas de minimiseur fourni avec NEMO. Celui utilisé par A. Weaver dans le système d'assimilation 4D-VAR avec l'adjoint codé manuellement s'appelle `m1qn3`, il est développé à l'INRIA par Gilbert & Lemaréchal⁴ Mais un peu de travail serait nécessaire à sa mise en oeuvre.

Hicham utilisait jusqu'à présent un minimiseur TAO (toolkit of advanced optimization), qui utilise une variante de L-BFGS. Il pourrait continuer à l'utiliser pour la version de démonstration (Hicham/Laurent : ce produit est-il sous licence, ou pourra-t-on le mettre à disposition d'utilisateurs futurs?).

4.3.4 Maintenance and service to the users

Dans un second temps, cette configuration serait adaptée à la version la plus récente de NEMO et fournie sous CVS, afin que les utilisateurs disposent d'un exemple de système d'assimilation fonctionnel pouvant leur servir de point de départ pour d'autres études.

⁴<http://www-rocq.inria.fr/estime/modulopt/optimization-routines/m1qn3/m1qn3.html>