Projet de fin d'études

Réalisation d'un modèle couplé océan-atmosphère avec le coupleur dynamique de codes parallèles Open-PALM

Auteur :
Pierre Trespeuch

Maître de stage :
Dr. Sophie Valcke

Tuteur de stage :
Dr. Denis Lapoire

6 septembre 2011

Remerciements

Je remercie toute l'équipe global change du CERFACS, et plus particulièrement Sophie Valcke et Thierry Morel qui ont encadré mon stage et l'ont rendu si enrichissant. Je remercie également Laure Coquart et Florent Duchaîne pour leur aide précieuse et Marie Piazza et Julien Boé pour m'avoir si bien accueilli dans leur bureau.

Table des matières

1 Introduction **6**
1.1 Raisons de ce choix de stage 6
1.2 Problématique générale 6
1.3 Plan du rapport 8

2 Présentation de l'entreprise **9**
2.1 Présentation du CERFACS 9
2.2 L'équipe "Global Change" 11
2.3 Le coupleur PALM 12
2.4 Les encadrants 13

3 Objectifs de la mission **14**
3.1 Sujet de stage 14
3.2 Cahier des charges fonctionnel 15
3.3 Critères de validation du projet 15
3.4 Moyens mis à disposition du stagiaire 16
3.5 Déroulement du stage 17

4 Réalisation du projet **19**
4.1 Analyse du cahier des charges et de ses contraintes 19
4.2 Réalisation d'un couplage à l'aide de OpenPALM/CWIPI . . . 20
4.3 Modèle NEMO 25
4.3.1 Structure de la grille 25
4.3.2 Partitionnement de la grille 25
4.3.3 Construction du maillage 26
4.4 Modèle ARPEGE 27
4.4.1 Structure de la grille 27
4.4.2 Partitionnement de la grille 28
4.4.3 Construction du maillage 31
4.5 Qualité de l'interpolation 34
4.5.1 Cartes et graphiques d'erreurs d'interpolation 34

4.6 Problèmes rencontrés et solutions apportées 38
4.6.1 Traitement du domaine de complétion 38
4.6.2 Besoin d'une définition d'un halo sur 2 points pour ARPEGE 43
4.6.3 Points isolés 45
4.7 Performance du couplage 49
4.7.1 Méthode 49
4.7.2 Résultats obtenus sur Octopus 49
4.8 Grilles icosaèdres 51

5 Conlusion **52**

Table des figures

4.1 En-têtes des fichiers implémentant les modèles NEMO/ORCA et ARPEGE. 22
4.2 Interface du coupleur OpenPALM 23
4.3 Grille ORCA 2 degrés. 25
4.4 Grille ARPEGE basse résolution. 27
4.5 Partitionnement de la grille gaussienne réduite. 28
4.6 Mailles triangulaires. 31
4.7 Maillage obtenu sur la grille gaussienne réduite basse résolution pour 3 décompositions est-ouest et 4 décompositions nord- sud (soit un total de 12 partitions). L'algorithme étant paramétré nous l'avons utilisé sur différentes résolutions allant jusqu'à 181724 points. 33
4.8 Erreurs d'interpolation sur les grilles Arpège et Orca avec échelle adaptative, les points masqués ont une valeur de -5. . . 35
4.9 Erreurs d'interpolation sur les grilles Arpège et Orca à échelle fixée, les points en blanc sont les points masqués ou hors de l'échelle de couleur. 36
4.10 Erreurs d'interpolation en fonction de l'indice du point. 37
4.11 Localisation des différences entre les résultats mono-processus et parallèles. 43
4.12 Un halo de deux points en latitude est nécessaire pour la construction de cette maille. 44
4.13 Différences de maillage avant et après traitement des points isolés(toutes les différences ne sont pas encadrées). 46
4.14 Cartes des erreurs d'interpolation avant et après traitement des points isolés. 47
4.15 Différences de la valeur absolue de l'erreur avant et après traitement des points isolés (les améliorations correspondent aux valeurs négatives). 48
4.16 Performances observées sur Octopus. 50
4.17 grilles icosaèdres. 51

4.18 grilles icosaèdres superposées. 51

Chapitre 1

Introduction

1.1 Raisons de ce choix de stage

Ayant choisi en $3^{ème}$ année l'option PRCD (Parallélisme Régulation et Calculs Distribués) j'ai pu découvrir le calcul parallèle et notamment le couplage de codes qui est le sujet de ce stage. De plus mon stage de deuxième année au CEA(Commissariat à l'Énergie Atomique et aux Énergies Alternatives) dans le domaine de l'informatique scientifique m'avait beaucoup plu. Le fait de pouvoir travailler dans un laboratoire où chercheurs et ingénieurs de différents domaines travaillent ensemble sur de nombreux projets m'a semblé le lieu idéal pour apprendre mais aussi utiliser les connaissances acquises lors de mon cursus à l'Enseirb-Matmeca. Ce stage s'est déroulé au CERFACS du 14/02/2011 au 16/09/2011.

1.2 Problématique générale

Les modèles de climat utilisés aujourd'hui en production tournent sur un nombre réduit de coeurs de calcul (de l'ordre de la dizaine) et certains encore sur des machines vectorielles. Au sein d'ENES (European Network pour Earth System Modelling, www.enes.org/) un récent atelier a permis de fixer les objectifs à 15 ans de la modélisation climatique en Europe.

A l'horizon 2022 et 2027, ce sont des ensembles de simulations d'une dizaine de membres chacun ayant une résolution de 10 km qui sont visés. Ces résolutions permettront de simplifier les hypothèses faites actuellement dans le traitement des nuages et des aérosols microphysiques dans l'atmosphère et de représenter les tourbillons de méso-échelle océanique. Pour atteindre ces résolutions, il semble aujourd'hui indispensable que les cœurs dynamiques des

composantes des modèles de climat évoluent vers d'autres types de maillages que les grilles structurées traditionelles. Les grilles non-structurées de type icosaédrique, qui présentent l'avantage de discrétiser la sphère de manière uniforme, commencent à s'imposer comme voie d'avenir[5].

Techniquement, il existe aujourd'hui deux approches principales en modélisation du climat pour réaliser le couplage de codes[4]. La première approche vise la fusion des codes des composantes originales en un seul et même code intégré. Pour ce faire les composantes originales doivent être séparées en trois unités regroupant respectivement l'initialisation, les calculs effectués à chaque pas de temps, et la finalisation. Les structures de données et l'interface d'appel de ces unités doivent être modifiés de façon à respecter les normes de codage de l'infrastructure logicielle de couplage utilisée qui sert alors pour reconstruire un code unique intégrant les unités des composantes originales. Cette approche est adoptée par plusieurs groupes aux Etats-Unis, en particulier par les développeurs du « Earth System Modeling Framework » (ESMF, `http://www.earthsystemmodeling.org`), par le Geophysical Fluid Dynamics Laboratory (GFDL) avec leur Flexible Modeling System (FMS, `http://www.gfdl.noaa.gov/fms`) et par le NCAR pour la constitution de CESM1 (`http://www.cesm.ucar.edu/models/cesm1.0`). Le principal avantage de cette approche est d'être particulièrement efficace pour exécuter séquentiellement les composantes sur un même ensemble de processeurs car les champs de couplage peuvent alors être partagés au sein de la mémoire du processeur. Le désavantage est d'imposer des contraintes de codages fortes (séparation en trois unités, structures de données, interface d'appel) ce qui est difficilement envisageable avec des codes existants développés par des groupes différents sans autorité dirigeante commune ce qui est le cas de la modélisation climatique en Europe.

La deuxième approche adopte la non-intrusivité comme contrainte forte, c'est-à-dire de ne devoir faire qu'un minimum de modifications dans les codes existants. L'outil de couplage agit en fait comme une librairie de communication liée aux codes à coupler et comme un exécutable séparé qui gère les composantes originales et éventuellement effectue les interpolations des champs de couplage. Les coupleurs OASIS et PALM développés au CERFACS suivent cette deuxième approche. Environ 35 groupes de modélisation du climat utilisent aujourd'hui OASIS en Europe mais aussi un peu partout dans le monde comme au Canada, aux États-Unis, au Japon, en Australie. Le succès d'OASIS est très certainement dû à sa caractéristique principale de non-intrusivité. La majorité des groupes utilisent toujours la version non-parallèle du coupleur, OASIS3 , bien qu'une première version complètement parallèle

soit aujourd'hui disponible, OASIS4. OASIS4 présente cependant une limitation importante qui est de ne pas traiter les maillages non-structurés. Ceci semble à moyen terme rédhibitoire avec l'arrivée des grilles icosaédriques dans le paysage de la modélisation du climat. C'est pourquoi, lors de ce stage, on se propose d'étudier la faisabilité d'un couplage océan-atmosphère avec Open-PALM qui, grâce à CWIPI ne présentera pas ces limitations. De plus, OpenPALM permet de gérer plusieurs instances d'une même composante au sein d'une application couplée (contrairement à OASIS) ce qui est particulièrement intéressant dans le cadre d'une simulation d'ensemble interactif.

1.3 Plan du rapport

Dans un premier temps je vais vous présenter le CERFACS, l'équipe Global Change dans laquelle j'ai travaillé et le coupleur OpenPALM. Ensuite je décrirai les objectifs du stage et son déroulement. Le problème du couplage avec OpenPALM ainsi que les spécificités des modèles à coupler seront alors présentés. Cette partie débouchera sur la description des problèmes rencontrés et des solutions apportées lors de ce stage. Les méthodes et résultats concernant la qualité de l'interpolation et les temps de calculs feront l'objet des deux dernières sections. L'accent sera mis sur les choix algorithmiques effectués ainsi que sur les méthodes de validation du couplage. Enfin une conclusion fera le bilan de ce stage de 7 mois au CERFACS.

Chapitre 2

Présentation de l'entreprise

2.1 Présentation du CERFACS

Le CERFACS (Centre Europeen de Recherche et de Formation Avancée en Calcul Scientifique) est un centre de recherche dont le rôle est de développer des méthodes de calcul avancées pour les simulations numériques et la résolution algorithmique de problèmes scientifiques et technologiques aussi bien pour la recherche que pour l'industrie. C'est pourquoi le CERFACS nécessite un accès aux machines de calculs les plus puissantes actuellement disponibles. Celles-ci seront présentées par la suite et nous reviendrons plus en détail sur celles utilisées pendant ce stage.

Le CERFACS est dirigé par un Conseil de Gérance constitué par les représentants de ses actionnaires, et un Conseil Scientifique préconise les orientations à suivre. Les sept actionnaires du CERFACS sont : le CNES (Centre National d'Etudes Spatiales), EADS France (European Aeronautic and Defence Space Company), EDF (Electricité de France), Météo-France, l'ONERA (Office National d'Etudes et de Recherches Aérospatiales), SAFRAN (groupe international de haute technologie), TOTAL (multinationale dans le domaine de l'énergie).

Le CERFACS regroupe plusieurs équipes interdisciplinaires, à la fois pour la recherche et pour la formation, dans les domaines suivants : physique, mathématiques appliquées, analyse numérique, dévelopement de logiciels.

Environ 115 personnes y travaillent, dont plus de 95 chercheurs et ingénieurs venant de dix pays différents. Ils travaillent sur des projets spécifiques dans neuf grands domaines de recherche : l'algorithmique parallèle, le couplage de codes, l'aérodynamique, les écoulements dans les turbines, la combustion, l'électromagnétisme, l'impact environnemental, l'assimilation de données et le climat.

Une partie de l'activité de recherche du CERFACS est réalisée en association avec le CNRS (Centre National de la Recherche Scientifique, www.cnrs.fr) en tant qu' "Unité de Recherche Associée" (SUC, URA 1875). Le CERFACS et l'INRIA (Institut National de Recherche en Informatique et Automatique (www.inria.fr) ont également joint une partie de leurs activités. Le CERFACS fait partie du pôle de laboratoires TVE (Terre Vivante et Espace, http ://tve.omp.obs-mip.fr/). C'est aussi un membre du RTRA/STAE (Réseau Thématique de Recherche Avancée "Sciences et Technologies pour l'Aéronautique et l'Espace", www.fondation-stae.net/fr/) et il participe aux activités de l'AESE (Pôle de Compétitivité "Aéronautique, Espace et Systèmes Embarqués", www.aerospace-valley.com).

La mission statutaire du CERFACS a été définie lors de la création de la Société Civile en 1996. Ses objectifs sont :

- de développer des recherches scientifiques et techniques visant à l'amélioration des méthodes de calcul avancé, incluant une meilleure prise en compte des processus physiques concernés et la mise au point d'algorithmes performants pour les nouvelles architectures de calculateurs ;
- de transférer ces connaissances scientifiques et ces méthodes techniques, pour l'application à de grands secteurs et domaines industriels ;
- de former des personnels hautement qualifiés et d'offrir une formation avancée à ces méthodes.

L'activité du CERFACS peut se résumer ainsi : "*Le CERFACS transfère les connaissances et le savoir-faire entre le milieu de la recherche fondamentale, où il est reconnu et considéré comme un acteur à part entière, et le monde des applications, où il trouve à la fois ses domaines d'activités et ses principaux objectifs, pour le bénéfice de ses Associés et de ses autres partenaires. Il prend par ailleurs en charge, plus directement si nécessaire, les besoins prospectifs des Associés dans des domaines insuffisamment explorés, tels aujourd'hui ceux de la simulation numérique avancée, de l'exploitation optimale des nouvelles architectures de calcul et de la préparation des nouvelles stratégies d'utilisation de ces architectures*"[1].

[1] d'après le Rapport de 2009 du Comité d'Entreprise

2.2 L'équipe "Global Change"

L'équipe "Modélisation du climat et de son changement global" ("Global Change") a été créée en 1990 et fait partie, depuis 1998, du Centre de national la Recherche Scientifique (CNRS) comme une unité de recherche associée URA1875. L'équipe conduit des recherches fondamentales et appliquées dans le domaine de l'étude de climat. Son objectif principal est d'apporter une contribution significative à la compréhension et à la prévision du climat et de l'environnement aux échelles globales et régionales. Sa stratégie repose fortement sur une approche duale basée sur les études théoriques et des travaux de modélisation ainsi que sur le développement de logiciels de haut niveau nécessaires pour traiter les couplages variés intervenant en science du climat. Ceci s'effectue à travers la coordination et la participation à des programmes de recherche nationaux et internationaux.

Le projet "climat" conduit des recherches appliquées et fondamentales dans le domaine des études de climat. Son objectif principal est d'apporter une contribution significative à la compréhension de la variabilité du climat aux échelles globales et régionales dans le but d'améliorer les projections climatiques ainsi que les études d'impact aux échelles saisonnières à décennales. Sa stratégie est fondée sur une approche duale basée sur les études théoriques et fondamentales ainsi que bien que sur le développement de logiciels de haut niveau nécessaires aux sciences du climat (logiciels de couplage, modélisation couplée à haute résolution, calcul de grilles, compression et réduction de données).

Le projet "Assimilation de données océaniques" est concentré sur le développement du système NEMOVAR qui devient la référence européenne pour l'assimilation de données océanique variationnelle. Le développement conjoint des thèmes algorithmiques et scientifiques est une de la force principale de ce projet.

Le projet "Diffusion de l'assimilation de données" se préoccupe du développement de techniques d'assimilation de données dans les champs différents comme la modélisation des cœurs de centrales nucléaires ou l'hydrologie. Un effort important sur la formation est développé dans le but de diffuser les approches d'assimilation de données dans de nouveaux champs.

Le projet "Logiciel Oasis" à pour but de permettre l'échange de champs de couplages entre des codes numériques de façon synchronisée. Alors que le coupleur Oasis3 est opérationnel, une version pleinement parallèle nommée Oasis4 est actuellement en développement au CERFACS.

Le projet "Logiciel PALM" ayant comme objectif le développement d'un coupleur dynamique généraliste est décrit en détail à la prochaine section.

2.3 Le coupleur PALM

Le projet de prévision opérationnelle de l'océan MERCATOR est à l'origine en 1996 du projet PALM.

Le directeur de MERCATOR de l'époque, Philippe Courtier, suggère qu'une chaîne d'assimilation de données pourrait être construite et implémentée comme un couplage. Les différentes tâches - faire tourner une prévision, appliquer l'opérateur d'observation - calculer l'erreur - approximer l'erreur statistique - inverser les matrices, minimiser une fonction de coût etc -seraient des pièces indépendantes de codes assemblées par un logiciel portable et flexible. Comme la plupart des algorithmes d'assimilation de données peuvent êtres implémentés comme différentes séquences de mêmes unités fonctionnelles, cette approche était la plus appropriée pour un contexte de recherche et développement, où de nombreuses méthodes sont à implémenter et à comparer.

L'équipe PALM au CERFACS était en charge du développement du logiciel. C'était un challenge technologique : développer un coupleur dynamique parallèle et son interface utilisateur qui seraient utilisés sur des plateformes variées sous les contraintes d'applications opérationnelles.

Le développement s'est déroulé en deux phases :

1. Un prototype complet a été implémenté en utilisant la technologie de la fin des années 1990. A l'issue de cette phase une version basée sur MPI1 a été achevée portant tout d'abord le nom de PALM_RESEARCH puis de PALM_SP.
2. De l'expertise née du développement du prototype, une version finale du coupleur basée sur MPI2 a été délivrée sous le nom de PALM_MP.

Grâce à la modularité de l'approche, le formalisme non spécialisé utilisé pour décrire les interfaces de couplage et la portabilité du logiciel, PALM s'est progressivement imposé comme un choix naturel pour toute sorte de couplages dynamiques parallèles ou d'assemblage de code.

En particulier, PALM est aujourd'hui utilisé par une quarantaine de groupes pour réaliser des simulations multi-physiques dans divers domaines tels la mécanique des fluides (l'aérodynamique, la combustion, etc). Pour répondre aux prérequis des futures simulations hautes performances en mécanique des fluides, PALM a interfacé la libraire de communication et d'interpolation CWIPI développée par l'ONERA. Pour rendre cette collaboration ainsi que celles à venir plus aisées, PALM est actuellement distribué comme une application open source sous license LGPL v3 sous le nom d'OpenPALM.

2.4 Les encadrants

Voici la liste des personnes qui ont participé à l'encadrement de mon stage :

- Sophie Valcke : Ingénieur de recherche, dirige le projet OASIS.
- Thierry Morel : Ingénieur de recherche, dirige le projet PALM depuis 2003.
- Florent Duchaîne : Chercheur, participe au développement de PALM.
- Laure Coquart : Ingénieur d'Études au CNRS.

Chapitre 3

Objectifs de la mission

3.1 Sujet de stage

Le sujet du stage porte sur la réalisation d'un modèle couplé océan-atmosphère avec le coupleur OpenPALM. Le code numérique de circulation générale océanique est NEMO du LOCEAN (voir `http://www.nemo-ocean.eu/`) et le code numérique d'atmosphère est ARPEGE de Météo-France (voir `http://www.cnrm.meteo.fr/gmgec/arpegeclimat/ARPCLI-V5.1/index.html`).

Ces deux codes sont actuellement déjà couplés avec le coupleur OASIS. OASIS, contrairement à OpenPALM,ne peut cependant pas traiter les maillages non-structurés ce qui à terme sera limitant pour le couplage climatique. Il s'agira d'évaluer la faisabilité d'utiliser le coupleur OpenPALM pour réaliser ce couplage. En particulier, il faudra tester la capacité de la librairie d'interpolation CWIPI développée à l'ONERA et en cours d'interfaçage dans OpenPALM à effectuer les transformations de champs requises pour exprimer sur le maillage du code atmosphérique les champs de couplages donnés par le code d'océan sur son propre maillage et vice-versa. Pour ce faire, le travail commencera dans un environnement Linux avec des modèles "jouets", i.e. des codes qui n'effectuent pas de réels calculs mais qui reproduisent des échanges de couplage réalistes. Le modèle jouet reproduisant le couplage ARPEGE-NEMO avec OASIS existe déjà et il s'agira donc de l'adapter au coupleur OpenPALM. Il sera alors possible d'identifier les fonctionnalités manquant à OpenPALM pour reproduire exactement le couplage OASIS.

Il s'agira ensuite de porter les deux modèles jouets sur les machines parallèles du CERFACS (IBM Nehalem cluster, HP AMD cluster) ainsi que sur

la machine SGI Altix ICE du CINES (`http://www.cines.fr`) et d'effectuer des tests de performances comparatifs des deux coupleurs en augmentant le nombre de processus.

3.2 Cahier des charges fonctionnel

Il existe de nombreux modèles jouets couplant des modèles océan-atmosphère sur des grilles différentes avec OASIS. Il faudra reproduire le couplage ARPEGE-NEMO avec OpenPALM et développer des outils permettant d'analyser à la fois la performance du couplage en termes de rapidité d'exécution mais aussi en terme de qualité d'interpolation. Il faudra aussi identifier les fonctionnalités manquantes dans OpenPALM et dans la bibliothèque d'interpolation CWIPI pour produire un couplage océan atmosphère satisfaisant. Ces différents points feront l'objet de rapports. Les fonctionnalités manquantes pourront êtres implémentées.

Il faudra implémenter différents algorithmes de "maillage" des grilles, selon le type de grille et l'algorithme d'interpolation que l'on voudra utiliser. De plus ces grilles doivent être découpées en différents domaines qui seront chacun traités par des processus différents. Différents types de partitionnement devront être implémentés et testés.

La scalabilité du couplage devra alors être testée en augmentant le nombre de processus sur les machines parallèles du CERFACS et en utilisant des grilles de résolutions de plus en plus importantes.

Afin de mener à bien la mission il sera nécessaire de collaborer avec de nombreuses personnes dont notamment les développeurs et les utilisateurs de PALM et d'OASIS ainsi que ceux de la bibliothèque d'interpolation CWIPI.

3.3 Critères de validation du projet

Le projet devra aboutir sur un interfaçage fonctionnel d'Open-PALM dans le modèle jouet reproduisant le couplage ARPEGE-NEMO du CNRM-CERFACS. Des analyses sur la qualité de l'interpolation ainsi que des comparaisons avec le coupleur OASIS devront être produites. De plus des analyses

des différences entre cas séquentiels et cas parallèles ainsi que des observations de performances sur différentes machines permettront de valider le projet.

3.4 Moyens mis à disposition du stagiaire

La majeure partie des tests ont été effectués sur la machine suivante sur laquelle il m'était possible d'utiliser jusqu'à 4 nœuds soit 32 cœurs.

L'IBM IDATAPLEX octopus est configuré de la manière suivante :
- 1 nœud de login bi-processeur quad core Intel Xeon "Nehalem" 2.93GHz 32 GO de mémoire.
- 1 nœud d'administration.
- 2 nœuds serveurs GPFS des répertoires utilisateur.
- 82 nœuds de calcul.

Chaque nœud possède les caractéristiques suivantes :
- Deux processeurs quad cœur Intel Xeon "Nehalem" 2.66 Ghz (4 flop par cycle par cœur soit 42.72 GFlops de performance crête par processeur).
- 24 GO de mémoire par nœud, (mémoire DDR3 cadencée à 1333 MHz).
- Caches L1 (instruction et données) 32 KO , cache L2 256 KO par cœur.
- Cache L3 de 8 M0 partagé par les 2 processeurs.

Le réseau d'interconnexion Infiniband offre une bande passante de 1.45 GO/s entre nœuds. La latence MPI est de 1.3 micro-secondes. Le système d'exploitation installé sur les nœuds est Suse SLES10SP2.

3.5 Déroulement du stage

Voici les principales étapes de mon stage :

du 14/02/2011 au 18/02/2011	Implémentation en fortran90 d'un nouvel algorithme de partitionnement pour le modèle jouet ARPEGE-NEMO couplé avec Oasis.
du 21/02/2011 au 25/02/2011	Ajout d'options activées par des clés de précompilation au modèle jouet afin de pouvoir ne tester qu'un seul champ de couplage. Ainsi les comparaisons avec le futur couplage OpenPALM seront plus aisées et les calculs plus rapides. Formation OpenPALM et formation de 2 jours en Fortran90.
du 28/02/2011 au 11/03/2011	Couplage d'un modèle jouet plus simple que le toy Arepege-Nemo à l'aide de OpenPALM. Début du couplage du toy Arpege-Nemo en fortran90 avec OpenPALM.
du 14/03/2011 au 25/03/2011	Mise au point et implémentation de l'algorithme de maillage à l'aide triangles. Ajout des sorties Netcdf. Début d'un wiki sur lequel seront mis l'ensemble des résultats obtenus pour les différents modèles jouets et algorithmes utilisés.
du 28/03/2011 au 01/01/2011	Étude du code de la bibliothèque d'interpolation CWIPI afin d'y apporter des modifications. Implémentation d'un prototype fortran90 permettant aux processus de récupérer les valeurs aux bordures sur les processus distants les ayant calculées.
du 04/04/2011 au 08/04/2011	Implémentation en C++ dans la bibliothèque CWIPI de cette nouvelle fonctionnalité.
du 11/04/2011 au 22/04/2011	Tests de performance sur la machine halong pour les modèles jouets couplés avec OpenPALM.Tests sur le calculateur Corail du modèle jouet Arpège-Nemo couplé avec OASIS.
du 25/04/2011 au 06/05/2011	Tests comparés de qualité d'interpolation et améliorations/corrections des algorithmes. Mises à jour du wiki avec de nombreuses figures.

du 09/05/2011 au 20/05/2011	Portage du toy couplé avec OASIS4 sur octopus. Tests de performances sur octopus. Différences mono-processus/multi-processus observées. Tests afin d'en détecter l'origine.
du 23/05/2011 au 03/06/2011	Toujours des tests sur les différences mono-processus/multi-processus. Installation de OpenPALM sur la machine Jade du CINES. Séminaire sur les entrées et sorties parallèles.
du 06/06/2011 au 24/06/2011	Corrections des différences mono-processus/multi-processus en utilisant un halo de 2 points autours des partitions. Installation d'une nouvelle version de CWIPI. Ajout des nouvelles fonctionnalités à cette version. Nouvel algorithme permettant la création de petits triangles en ajoutant des points initialement non présents sur la grille du modèle au niveau des points isolés.
du 27/06/2011 au 08/07/2011	Abandon de l'algorithme des "petits triangles" au profit d'une extension des mailles au niveau des cotes.
du 11/07/2011 au 22/07/2011	Mise au propre du modèle jouet le plus simple (Orca-LMDZ) pour l'envoyer au DKRZ(Deutsches KlimaRechenZentrum). Rédaction du rapport de stage.
du 25/07/2011 au 05/08/2011	Rédaction du rapport de stage. Visite de la salle machines.
du 08/08/2011 au 19/08/2011	Extensions des mailles seulement au niveau des points isolés. Comparaison des résultats avec les anciennes méthodes.
du 22/08/2011 au 02/09/2011	Rédaction du Rapport de stage et préparation de la soutenance.
du 05/09/2011 au 16/09/2011	Finitions.

Chapitre 4

Réalisation du projet

4.1 Analyse du cahier des charges et de ses contraintes

Le cahier des charges amène les contraintes suivantes :

- Performances « scalables ». Diminution du temps de calcul proportionnel au nombre de processus.
- L'étude de la qualité de l'interpolation nécessite de définir ce que représente une « bonne interpolation » et de choisir des outils pour la calculer/visualiser.
- L'analyse des fonctionnalités manquantes à CWIPI doit passer par l'étude du code de la bibliothèque afin de repérer toutes les fonctionnalités existantes et d'en déduire celles qui manquent. Il n'existe pas encore de documentation pour la bibliothèque d'interpolation mis à part le code source et les commentaires qu'il contient. De plus CWIPI est basé sur la bibliothèque FVM (Finite Volume Mesh) développée par EDF qu'il faudra donc aussi étudier.
- Les modèles jouets doivent être codés en fortran90. Les bibliothèques CWIPI et FVM sont écrites en C++ et OpenPALM est implémenté en C.
- Les modèles jouets couplés avec OpenPALM doivent reproduire exactement ceux couplés avec OASIS4. Il faudra donc utiliser les modèles jouets couplés avec OASIS4 et les modifier pour effectuer les comparaisons demandées. Cela nécessite d'apprendre à se servir du coupleur OASIS4[2].
- Ces modèles jouets doivent produire des résultats reproductibles sans différences mono-processus/multi-processus.
- Un tel travail suppose des rapports réguliers qui devront être accessibles

facilement(création d'un wiki : https ://oasistrac.cerfacs.fr/wiki/palm1).
- Il faudra faire correspondre les modèles jouets aux modèles réels en ce qui concerne le découpage des grilles climatiques. Il faudra donc implémenter les algorithmes de partitions parallèles en conséquence.
- Les compétences techniques sur les logiciels OpenPALM,OASIS4 et CWIPI sont détenues par des personnes différentes au CERFACS et à l'ONERA. Ces différentes personnes étant intéressées par les résultats obtenus lors du stage, il conviendra de prévoir des réunions a intervalle régulier pour informer et réorienter mon travail sur de nouvelles pistes si nécessaire.

4.2 Réalisation d'un couplage à l'aide de OpenPALM/CWIPI

La première étape du stage a consisté à prendre en main le coupleur OpenPALM et la bibliothèque d'interpolation CWIPI sur des exemples de couplage simples. Afin de simplifier la tâche ce sont des modèles « jouets » qui ont été utilisés.

Voici les étapes de définition d'un couplage à l'aide de OpenPALM/CWIPI :

- Dans les commentaires du code des deux modèles inclure des cartes d'identités[1], voir figure 4.1. Ces cartes d'identités permettent au coupleur d'avoir des informations sur les codes à exécuter. Ainsi c'est grâce à ces métadonnées que OpenPALM prend connaissance du couplage CWIPI et des champs impliqués dans ce couplage. On peut éventuellement définir plusieurs couplages pour un même modèle.
- Dans les deux codes, appeler la fonction PCW_Create_coupling qui permettra à CWIPI d'instancier un objet « couplage » et de définir les communicateurs MPI qui serviront lors des envois et réceptions des champs de couplages. Cet objet est contenu dans une table de hachage. On peut donc ensuite accéder à un objet couplage à l'aide de son nom.
- Dans les codes sources, lire les grilles climatiques au format de fichier auto-descriptif Netcdf(Network Data Form) et définir les maillages associés (plus de détails sont donnés page 24).
- Dans une boucle temporelle, implémenter l'appel aux fonctions PCW_Send et PCW_Recv permettent d'échanger des champs qui seront interpolés de la grille source à la grille cible. La fonction PCW_Send appelle

une méthode de l'objet « couplage » de la bibliothèque d'interpolation CWIPI. Voici le prototype simplifié de la fonction PCW_Send (le prototype de la méthode CWIPI est similaire) :

```
PCW_Send(nom du couplage,
         nom de l'échange,
         stride,(permet d'envoyer plus d'une valeur par point du maillage)
         temps/date de l'échange, &
         identifiant de l'échange, &
         nom du champ, &
         tableau 1d das valeurs en chaque point du maillage)
```

- Utilisation de l'interface Prepalm pour créer deux branches parallèles ce qui indique à OpenPALM de faire tourner en parallèle les deux modèles, voir figure 4.2(c) et 4.2(b).
- Utilisation de l'interface Prepalm pour définir les échanges de couplage, voir figure 4.2(a).

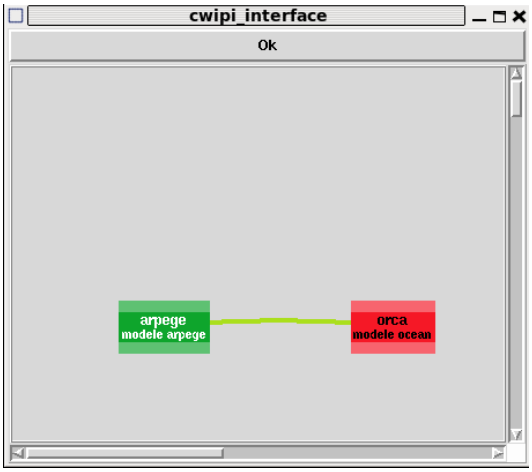
Orca

```
!PALM_UNIT -name orca\
!          -functions {f90 orca}\
!          -parallel mpi\
!          -minproc 1\
!          -maxproc 100000\
!          -object_files {Write_Result.o orca_domain.o orca.o}\
!          -comment {modele ocean}
!
!PALM_CWIPI_COUPLING -name ocean\
!
!PALM_CWIPI_OBJECT -name surface_downward_water_flux\
!                  -coupling ocean\
!                  -intent OUT
!
!PALM_CWIPI_OBJECT -name sea_surface_temperature\
!                  -coupling ocean\
!                  -intent IN
```

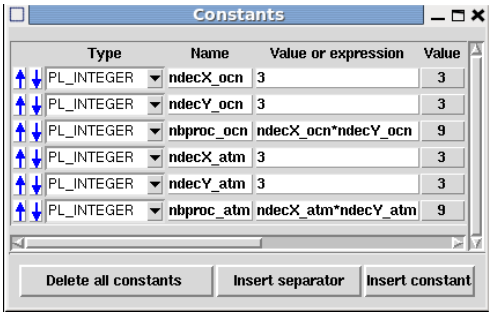
Arpège

```
!PALM_UNIT -name arpege\
!          -functions {f90 arpege}\
!          -parallel mpi\
!          -minproc 1\
!          -maxproc 100000\
!          -object_files {Write_Result_gauss.o arpege_domain.o arpege.o gather_sort.o}\
!          -comment {modele arpege}
!
!PALM_CWIPI_COUPLING -name atm\
!
!PALM_CWIPI_OBJECT -name sea_surface_temperature\
!                  -coupling atm\
!                  -intent OUT
!
!PALM_CWIPI_OBJECT -name surface_downward_water_flux\
!                  -coupling atm\
!                  -intent IN
```

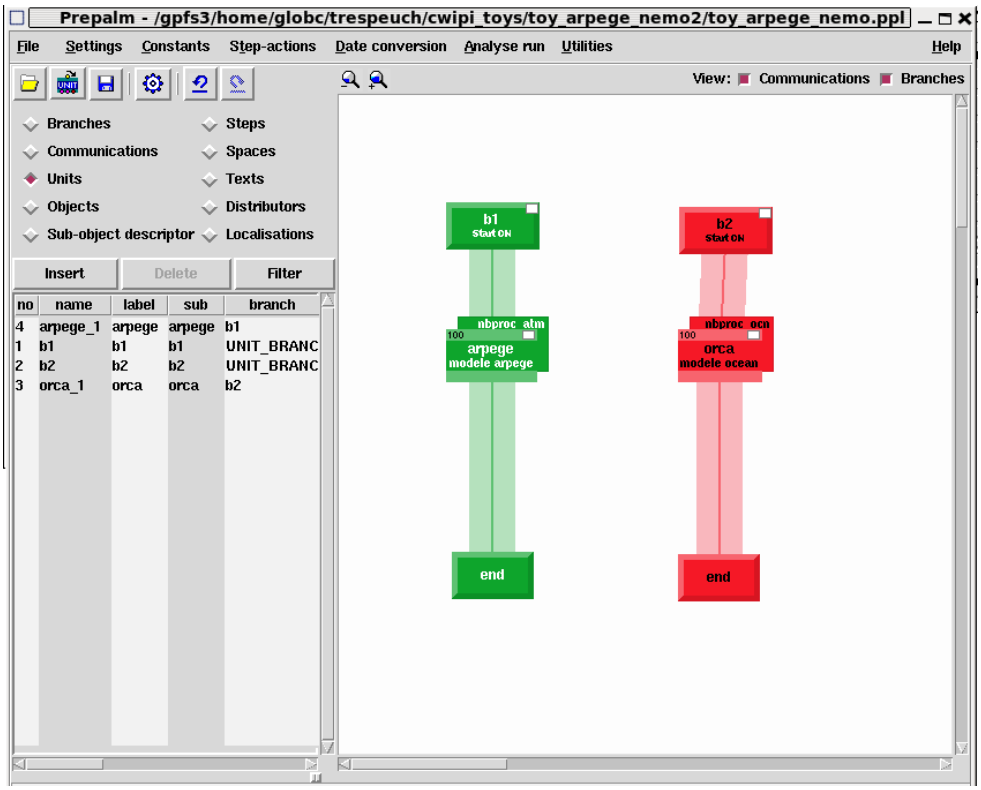
FIG. 4.1 – En-têtes des fichiers implémentant les modèles NEMO/ORCA et ARPEGE.



(a) Fenêtre permettant de définir un échange via la bibliothèque d'interpolation CWIPI intégrée au coupleur PALM



(b) Constantes qui déterminent le découpage des grilles de chaque modèle



(c) Écran principal où on peut voir deux branches permettant l'exécution en parallèle des modèles eux mêmes parallèles (branche verte pour ARPEGE et rouge pour ORCA).

FIG. 4.2 – Interface du coupleur OpenPALM

Comment définit on un maillage à CWIPI ?

CWIPI fonctionne avec des maillages non structurés. Les grilles structurées comme celles d'ARPEGE et NEMO peuvent être définies sous forme de maillages non structurés en décrivant des quadrilatères ou des triangles. Ceci se fait à l'aide d'une fonction fortran qui prend les paramètres suivant :

- le nombre d'éléments (donc ici le nombre de triangles ou de quadrilatères).
- le nombre et les coordonnées de chaque point (définis à l'origine dans le fichier netcdf de la grille).
- un tableau de connectvité contenant les indices concaténés des sommets des éléments.
- un tableau de taille du nombre d'élément+1 dont la $i^{ème}$ valeur est la somme cumulée du nombre de sommets des $(i-1)^{ème}$ premiers éléments (donc ici la $i^{ème}$ valeur du tableau sera $(i-1)*3$ car en fortran le premier indice d'un tableau est 1).

4.3 Modèle NEMO

4.3.1 Structure de la grille

Le modèle NEMO traite la partie océanique du système couplé. Il est basé sur une discrétisation en différences finies 3D de l'océan. La grille utilisée par le modèle NEMO pour les simulations climatiques est la grille ORCA (figure 4.3). Cette grille est « logiquement-rectangulaire », les latitudes et longitudes des points peuvent être exprimés dans deux tableaux distincts à deux dimensions (I,J). Cette grille est étirée : elle possède un pôle de convergence au Pôle Sud, mais le pôle de convergence habituel des méridiens au Nord a été divisé en deux pôles de convergence distincts déplacés l'un sur l'Asie, l'autre sur l'Amérique afin de ramener les singularités des pôles de convergence sur des continents qui sont masqués. Le couplage entre l'océan et l'atmosphère se fait uniquement à la surface de l'océan, donc sur un domaine à deux dimensions.

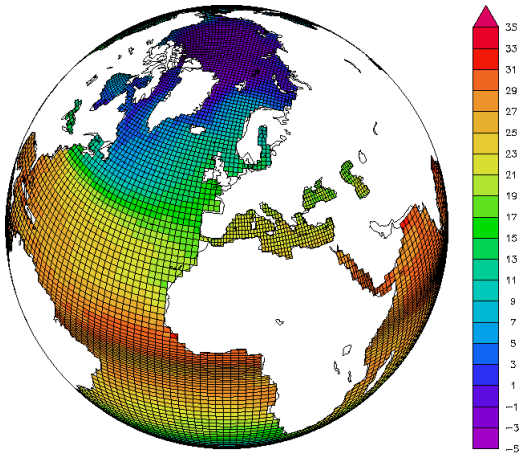


FIG. 4.3 – Grille ORCA 2 degrés.

4.3.2 Partitionnement de la grille

Le modèle NEMO tournant en parallèle, chaque processus calcule les champs d'une partition de la grille.

Ces partitions sont de simples divisions euclidiennes du maillage. La grille étant définie comme un tableau 2D (I,J) chaque partition sera définie par des

indices minimums et maximums dans chaque dimension. On obtient donc des partitions rectangulaires.

Du fait du fonctionnement de la bibliothèque d'interpolation il faut que les partitions se chevauchent c'est à dire qu'il faut qu'aux bordures, les points soient connus par les deux partitions adjacentes.

4.3.3 Construction du maillage

Les fichiers netcdf contiennent les coordonnées des « points » où sont définis les champs mais aussi les coordonnées de « coins » qui délimitent les surfaces sur lesquelles les valeurs des points sont valables.

L'interpolation testée pendant ce stage est l'interpolation bilinéaire. L'algorithme d'interpolation implémenté dans CWIPI est construit de façon à utiliser les valeurs des champs aux nœuds du maillage et donne de moins bons résultats lorsque l'on construit le maillage avec les coins (car les valeurs des champs aux coins sont dans ce cas tout d'abord interpolés à partir des valeurs connues aux points).

Afin d'avoir une qualité optimale d'interpolation, il a donc été choisi de définir les maillages avec les points où sont définis les champs et de ne pas tenir compte des « coins ».Néammoins lorsque des interpolations conservatives seront implémentées il sera nécessaire de tenir compte de la surface associée à chaque point délimitée par les coins.

Chaque colonne i et chaque ligne j du maillage contenant le même nombre de points, on peut aisément former des quadrangles constitués chacun de 4 points (i,j), (i+1,j), (i+1,j+1), (i,j+1) et se trouvant donc à cheval sur deux lignes et deux colonnes.

4.4 Modèle ARPEGE

4.4.1 Structure de la grille

Le modèle numérique ARPEGE est un modèle de circulation générale global et spectral développé par Météo-france en collaboration avec le Centre Européen de Prévision (CEP à Reading, U.K.). Ce modèle représente la composante atmosphérique du modèle couplé. La grille du modèle ARPEGE est une grille gaussienne réduite (voir figure 4.4). Le maillage est constitué de plusieurs cercles, chacun étant à une latitude constante. Le nombre de mailles par cercle de latitude diminue au fur et à mesure que l'on se rapproche des pôles de sorte que chaque maille couvre approximativement la même surface. Les latitudes et longitudes sont contenus dans des tableaux à une dimension couvrant l'ensemble des cercles.

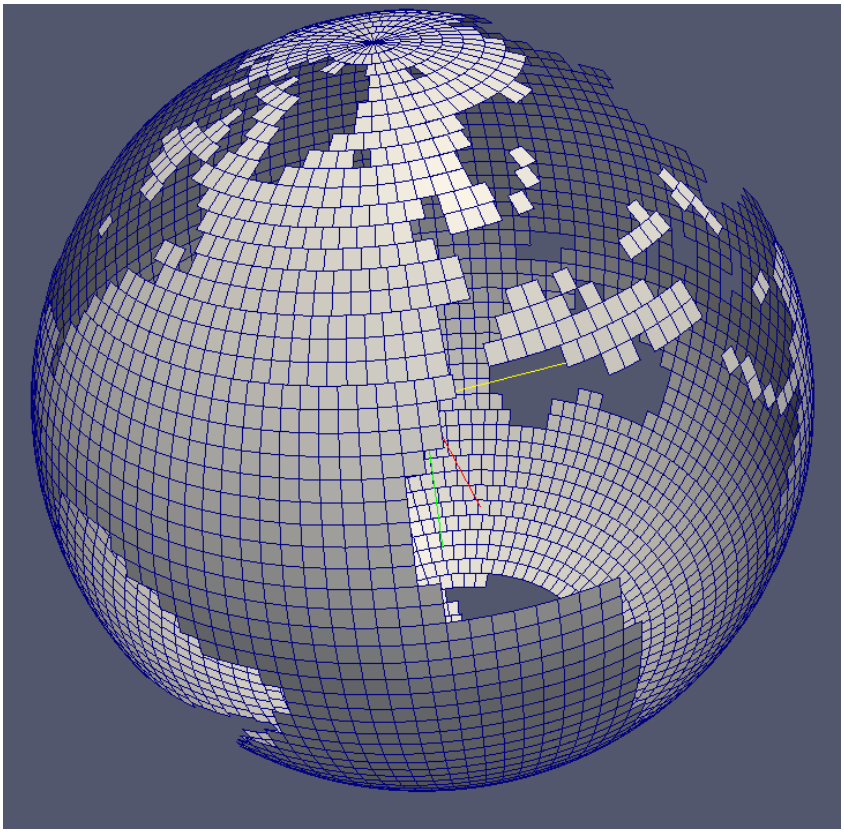


FIG. 4.4 – Grille ARPEGE basse résolution.

4.4.2 Partitionnement de la grille

La première étape de mon stage a été de coder entièrement et d'implémenter un algorithme de décomposition de grille gaussienne réduite correspondant à celui utilisé par le ECMWF [3] (The European Centre for Medium-Range Weather Forecasts). Ceci afin de démontrer que le couplage suivant de telles partitions est possible dans le cas de l'utilisation du coupleur OASIS4 mais aussi du coupleur PALM/CWIPI. L'algorithme se décompose en deux étapes : une décomposition nord-sud puis une décomposition est-ouest de la grille ARPEGE recouvrant le globe terrestre. Nous verrons ensuite qu'une troisième étape est nécessaire dans le cas de l'utilisation du coupleur PALM/CWIPI.

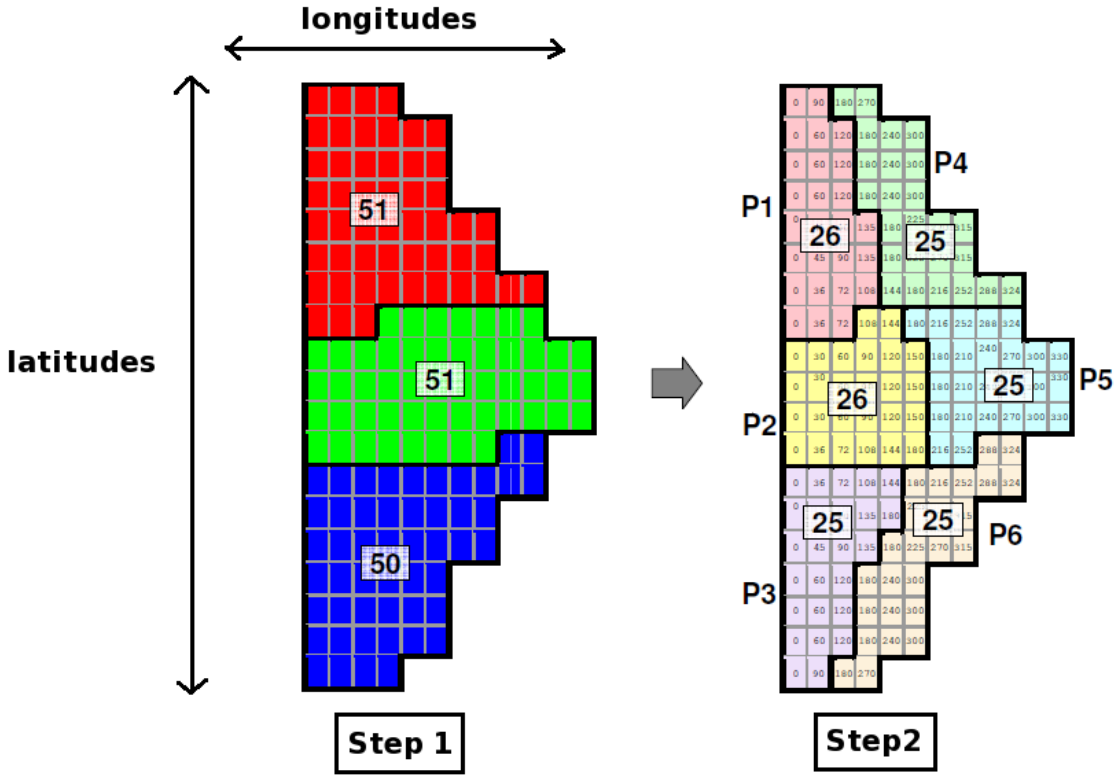


FIG. 4.5 – Partitionnement de la grille gaussienne réduite.

Entrées de l'algorithme :

- nblat : entier, le nombre de cercles de latitudes.
- npt_hor : entier, le nombre de points total.
- nlon_circle : un tableau d'entiers contenant le nombre de points par latitude.
- lon, lat : deux tableaux d'entier contenant respectivement les latitudes et longitudes de chaque point.
- nb_ew, nb_ns : deux entiers représentant respectivement le nombre de partitions dans la direction est-ouest et dans la direction nord-sud que l'on souhaite obtenir.

Sorties de l'algorithme :

- il_offset : un tableau d'entier qui pour chaque cercle de latitude contient l'indice du premier élément contenu par la partition du processus courant.
- il_extent : un tableau d'entier qui contient pour chaque cercle de latitude le nombre d'éléments contenu par la partition du processus courant.
- proc_mask : un tableau de booléens de la taille du nombre total de points contenant VRAI pour chaque élément appartenant à la partition du processus courant.

Première étape :

La première étape consiste à mettre en place des pré-partitions nord-sud qui contiennent toutes le même nombre de points. Une simple division euclidienne permet de déterminer les indices minimums et maximums de chaque pré-partition nord-sud. Le reste de la division sera attribué à la dernière partition. À noter qu'une pré-partition peux contenir des latitudes incomplètes.

Deuxième étape :

Lors de cette étape le but est de décomposer dans la direction est ouest les pré-partitions nord sud créées lors de la première étape.
Les offsets et les extents sont initialisés à 0 pour chaque latitude(cela représente une partition vide). Puis on trie les points de la partition obtenue à la première étape selon la longitude. On parcourt les points de la partition nord-sud du processus courant et on ajoute le point à la partition du processus de rang k si ce point se trouve entre le $((k-1) modulo (nb_ew))^{ème}(nb_ew)$-

quantile et le $((k)modulo(nb_ew))^{ème}(nb_ew)$-quantile.
L'ajout d'un point a lieu comme suit : on incrémente l'extent de 1 ; l'offset à cette latitude prend la valeur minimum entre l'offset du point et l'offset précédent.
Cette algorithme assure que les processus auront tous autant de points dans leurs partitions et que les partitions seront contiguës longitudinalement.

Par exemple, pour nb_ew=2 et nb_ns=1 on aura la répartition suivante :
- Processus 1 : tous les points situés avant la médiane. (On ajoutera ici le(s) point médians par convention).
- Processus 2 : tous les points situés après la médiane.

Extension des partitions pour le coupleur PALM/CWIPI

Pour la bibliothèque d'interpolation CWIPI les partitions doivent se chevaucher, les point aux bordures doivent êtres contenus par les deux partitions voisines. Pour cela l'algorithme précédent a du être légèrement modifié : à la première étape on ajoute la fin de la dernière latitude dans la direction Est-Ouest ainsi qu'une latitude supplémentaire aux partitions (hormis la dernière) dans la direction Nord-Sud.

4.4.3 Construction du maillage

Une grille gaussienne réduite ne contient pas le même nombre de points sur chaque latitude, on ne peut donc pas la « mailler » à l'aide de quadrangles dont les sommets seraient les points de la grille. La solution choisie à été de la mailler à l'aide de triangles.

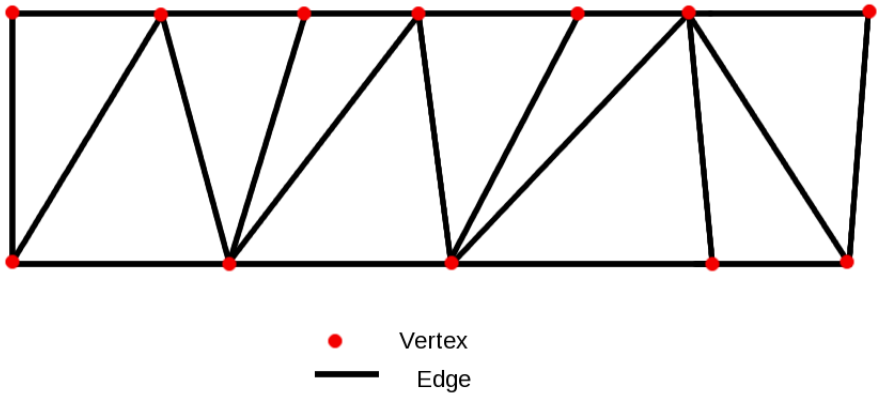


FIG. 4.6 – Mailles triangulaires.

On remarque que pour effectuer un tel maillage il faut alterner des triangles dont la base se trouve sur le cercle de latitude la plus basse et des triangles dont la base se trouve sur le cercle de latitude la plus haute (figure 4.6).

L'algorithme choisi construit d'ouest en est les triangles de façon à minimiser pour chaque triangle la longueur de son plus grand côté. Cette algorithme correspond à la façon la plus intuitive de résoudre ce problème. On peut voir le résultat sur la figure 4.7.

Algorithme :

Pour chaque couple de cercles de latitudes adjacentes

Tant que il reste des points sur une des deux latitudes

On ajoute au tableau des connectivités le premier point (longitude la plus faible) disponible de chacune des deux latitudes.

On ajoute au tableau des connectivités le point disponible d'une des deux latitudes dont la longitude est la plus faible qui ne fait pas partie des deux points précédemment ajoutés.

On retire le point précédent de la liste des points disponibles.

On incrémente le nombre d'élément.

Fin Tant que

Fin Pour

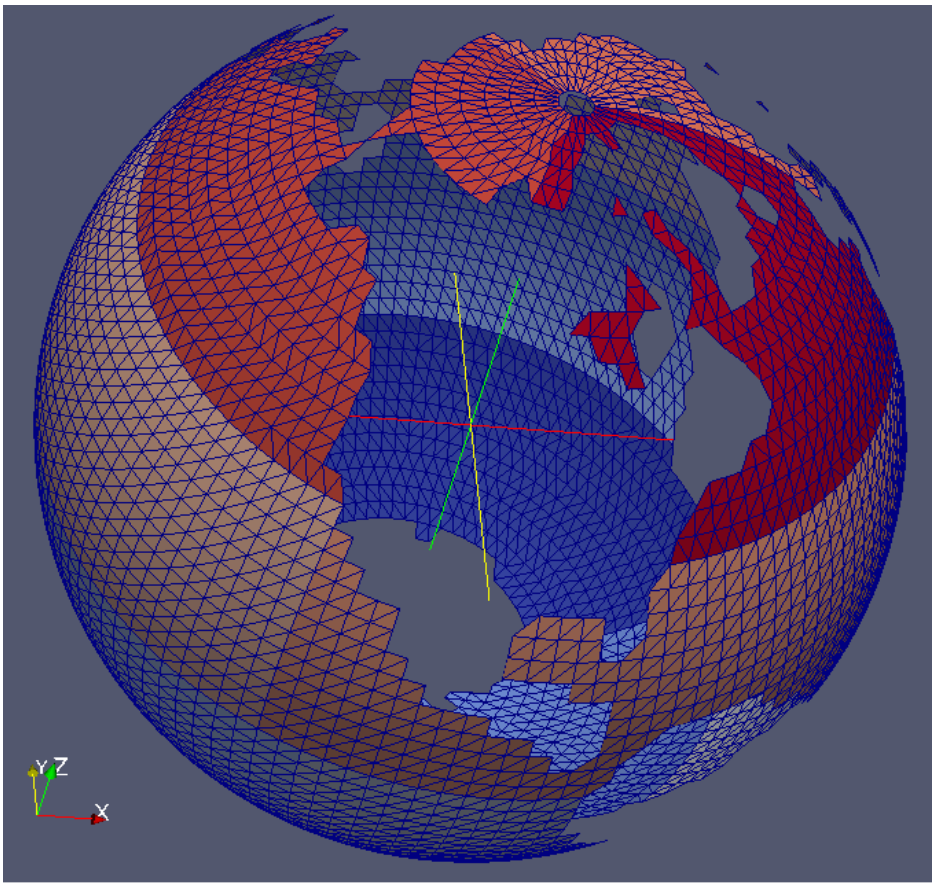


FIG. 4.7 – Maillage obtenu sur la grille gaussienne réduite basse résolution pour 3 décompositions est-ouest et 4 décompositions nord- sud (soit un total de 12 partitions). L'algorithme étant paramétré nous l'avons utilisé sur différentes résolutions allant jusqu'à 181724 points.

4.5 Qualité de l'interpolation

Dans cette section sont présentés des résultats qui ont été obtenus en fin de stage. La section qui la suit présente comment ces résultats ont étés obtenus et quelles solutions ont été apportés aux problèmes rencontrés.

4.5.1 Cartes et graphiques d'erreurs d'interpolation

Afin de tester la qualité de l'interpolation, on utilise une fonction analytique pour calculer les valeurs des champs à interpoler. On peut ainsi comparer le champ interpolé reçu par un modèle cible à la valeur de cette fonction analytique aux points du maillage cible pour déterminer la qualité de l'interpolation. L'erreur d'interpolation est calculée comme suit :

$$\boxed{\mathbf{Erreur} = \frac{\mathbf{fonction_analytique - champ_recu}}{\mathbf{champ_recu}} * \mathbf{100}}$$

La fonction analytique suivante à été utilisée pour les tests :

$$\boxed{\mathbf{f(x,y,z)} = \mathbf{2} - \mathbf{cos}(\pi(\frac{\mathbf{acos(cos(y)cos(x))}}{\mathbf{1.2}*\pi}))}$$

Les champs reçus par chacun des deux modèles ainsi que les erreurs associées sont écrits dans des fichiers netcdf. Grâce à l'outil de visualisation des fichiers netcdf ferret on obtient des figures comme la figure 4.8 qui permettent de comparer les qualités d'interpolation obtenues avec le coupleur PALM/CWIPI de celles obtenues avec le coupleur OASIS4.

La figure 4.8 montre peu de différences entre les deux couplages, les erreurs sont visualisées avec une échelle adaptative qui permet de ne pas sortir de l'échelle de couleur. On peut noter que pour les champs interpolés sur la grille ARPEGE une partie de l'Antarctique n'est pas géré par OASIS4 car c'est un domaine non couvert par la grille source (ORCA). Némmoins le couplage avec OpenPALM donne une valeur à ces points par recherche du plus proche voisin.

En fixant une plus petite échelle, comme sur la figure 4.9, on remarque que les résultats obtenus avec le couplage OpenPALM sont globalement meilleurs, notamment au niveau des pôles.

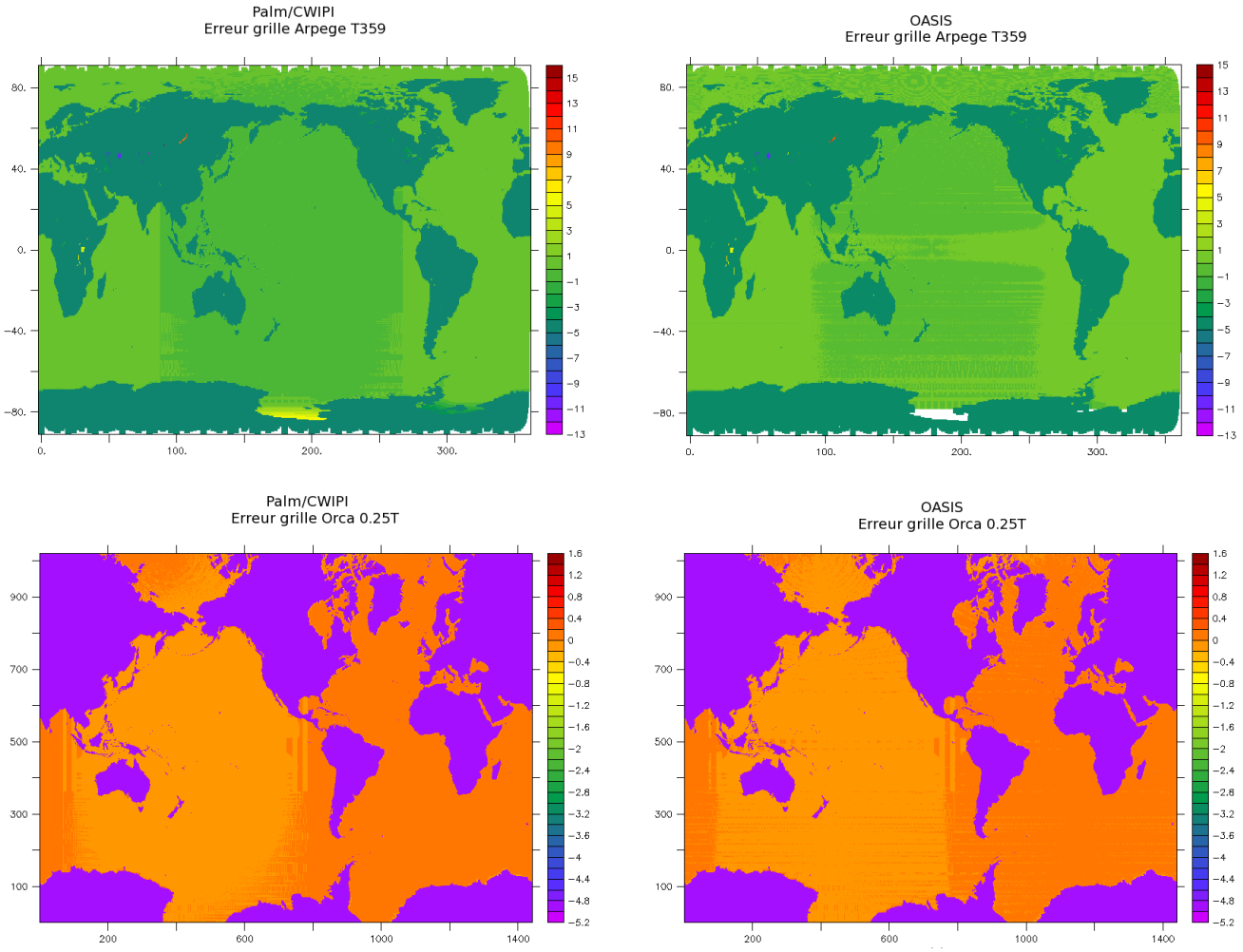


FIG. 4.8 – Erreurs d'interpolation sur les grilles Arpège et Orca avec échelle adaptative, les points masqués ont une valeur de -5.

Cette projection géographique n'est pas la seule possibilité d'analyser les résultats. En effet pour une résolution importante les points isolés à erreurs fortes ne seront pas facilement repérables. C'est pour cette raison que des tracés de l'erreur en fonction de l'indice du point ont été réalisés (voir figure 4.10). Cette méthode à été utilisé pour comparer les différents algorithmes utilisés mais il reste à analyser en détails ces résultats et vérifier que les points à erreur forte sont bien dans des zones loin de tout point de la grille source (comme par exemple les lacs).

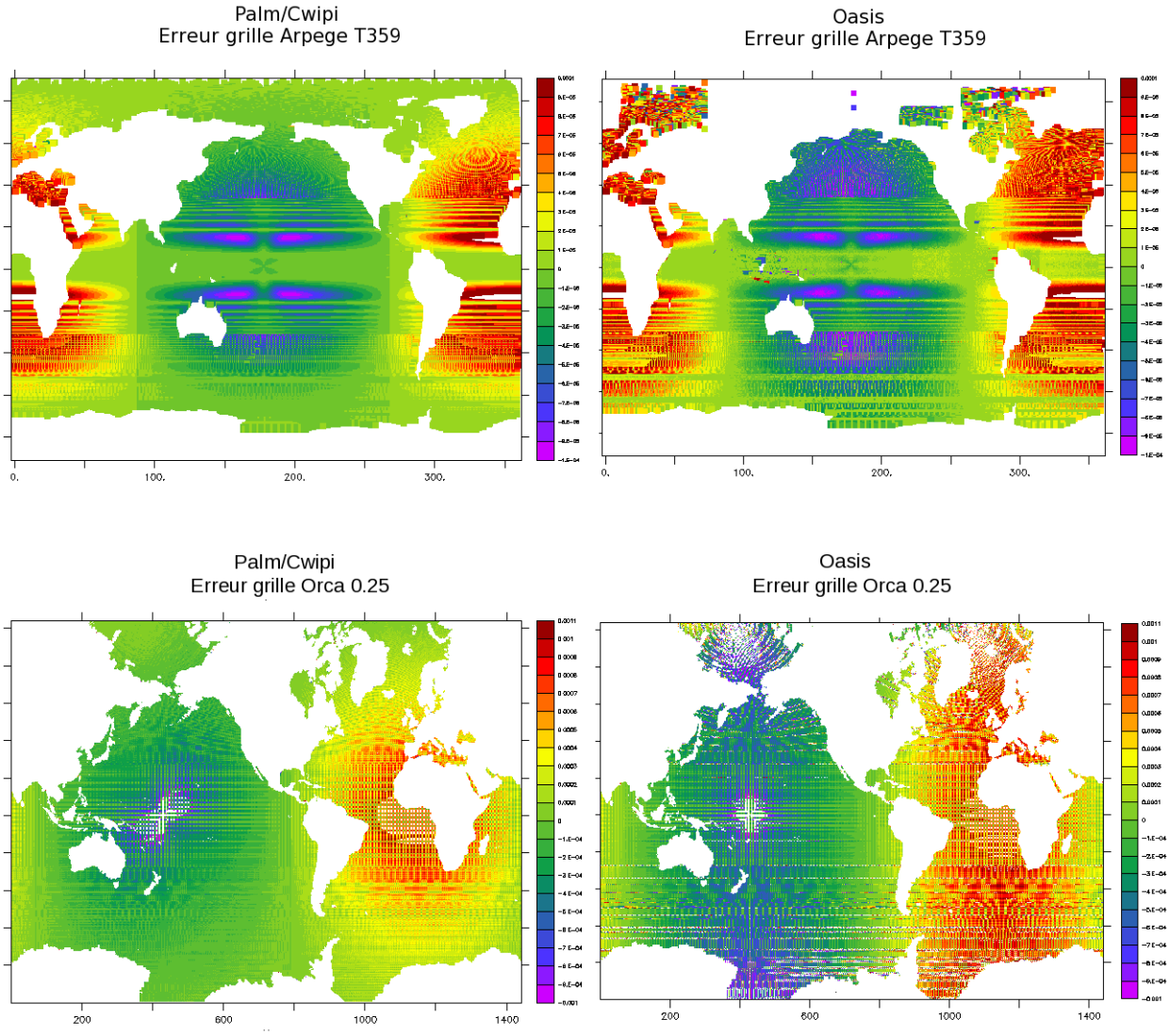


FIG. 4.9 – Erreurs d'interpolation sur les grilles Arpège et Orca à échelle fixée, les points en blanc sont les points masqués ou hors de l'échelle de couleur.

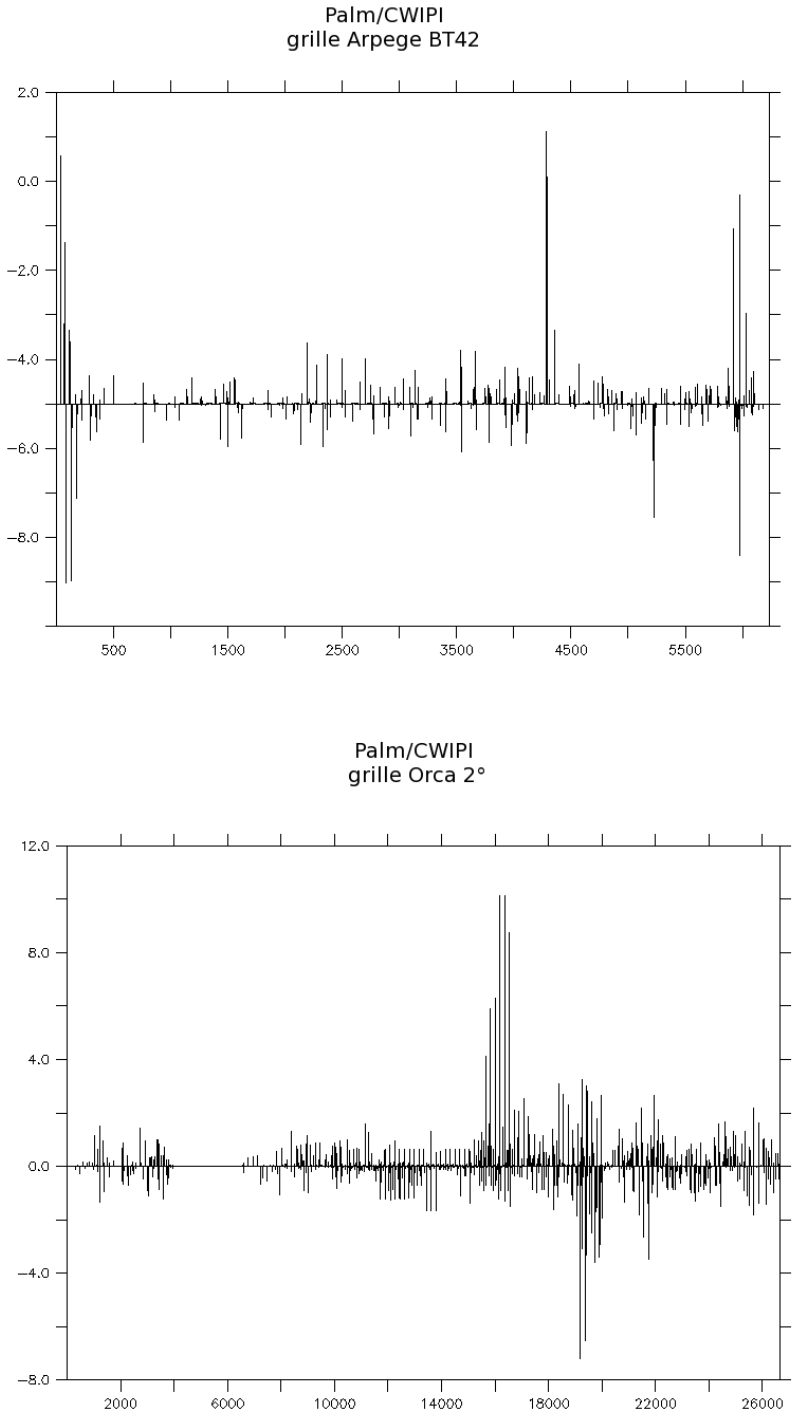


FIG. 4.10 – Erreurs d'interpolation en fonction de l'indice du point.

4.6 Problèmes rencontrés et solutions apportées

4.6.1 Traitement du domaine de complétion

Algorithme

Nous avons vu que CWIPI nécessite que les partitions se chevauchent pour effectuer les interpolations. Or dans le modèle ARPEGE qui utilise la grille gaussienne réduite les partitions ne se chevauchent pas. Les champs des points ajoutés artificiellement aux bordures n'ont donc pas leur valeur calculée sur le processus courant.

Un processus a donc deux domaines, son domaine de calcul et son domaine d'interpolation, le domaine de calcul étant inclus dans le domaine d'interpolation.

Nous appellerons dans la suite domaine de complétion la différence entre le domaine d'interpolation et le domaine de calcul.

La solution proposée est d'aller chercher les valeurs manquantes (c.à.d celles appartenant au domaine de complétion) chez les autres processus à l'aide de l'interface de communication MPI (Message Passing Interface).

L'algorithme se décompose en deux étapes :

- La première étape permet de localiser les points des domaines de complétion des processus. C'est à dire pour chaque point du domaine de complétion un processus demandera à tous les autres processus lequel le possède dans son domaine de calcul. Ainsi il saura quel processus interroger par la suite sans autre requête.
- La deuxième étape implémente l'échange entre processus permettant d'obtenir toutes les valeurs des domaines de complétion à partir des processus qui contiennent ces points dans leur domaine de calcul. Cette étape doit s'effectuer à chaque pas de temps du couplage.

Algorithme :

1ère étape, localisation :

```
Pour chaque processus k
    Si mon rang est k alors

          Envoie des coordonnées des points du domaine
          de complétion du processus courant. (MPI_Bcast)

          Pour chaque processus distant p

              Réception des coordonnées des points du domaine
              de complétion qui font partie du domaine de
              calcul du processus p.(MPI_Recv)

              Ces indices sont stockés dans un vecteur
              attribut de l'objet coupling.

          Fin Pour
    Sinon

        Réception des coordonnées envoyées par le
        processus de rang k. (MPI_Bcast)

        On recherche les indices des points correspondants
        qui appartiennent au domaine de calcul du processus
        courant.

        Ces indices sont stockés dans un vecteur attribut
        de l'objet coupling.

        Envoie des coordonnées correspondantes au processus
        de rang k. (MPI_Send)

    Fin Si
Fin Pour
```

2ème étape, échange :

```
Pour chaque processus k
    Si mon rang est k alors
          Pour chaque processus distant p

                Réception des valeurs manquantes que possède
                le processus p (MPI_Recv) (les indices des
                points du domaine de complétion présents dans
                le domaine de calcul du processus p ont été
                identifiés lors de la première étape).

          Fin Pour
    Sinon

       Envoie des valeurs manquantes au processus k (MPI_Send).
       (les indices qui lui manquent ont été identifiés lors
       de la première étape) (MPI_Recv)

    Fin Si
Fin Pour
```

Remarques :

À la première étape et à la seconde étape, on effectue O(nb_proc) communications pour chaque processus. Les recherches à la première étape sont de complexité $O((n/nb_{proc}) * (nb_{miss}))$ avec n le nombre de points total de la grille et nb_miss le nombre de valeurs manquantes. Or il y a nb_proc recherches d'où une complexité en $O(n * nb_{miss})$.

En pratique, les mesures de performance montrent que ces opérations sont peu coûteuses par rapport aux autres opérations nécessaires à l'échange et l'interpolation des champs entre les codes couplés.

Implémentation

Un prototype simplifié a tout d'abord été implémenté en fortran90 afin de vérifier la faisabilité de l'implémentation de cette algorithme ainsi que de s'assurer que le temps de calcul n'était pas rédhibitoire. Une fois cette étape achevée, il a été décidé d'intégrer cette fonctionnalité à la bibliothèque d'interpolation CWIPI.

CWIPI est implémentée en C++ et a donc une architecture orientée objet. Lors de la création d'un couplage entre deux codes, un objet « coupling » est créé. Les maillages de chacun des deux codes sont définis grâce à la méthode defineMesh de cet objet « coupling ».

Chaque processus appelle la méthode defineMesh de son objet coupling. C'est à ce moment qu'il a été choisi de transmettre l'information signifiant que certains points du maillage ont la valeur de leurs champs présents sur des processus distants.

Cette information est transmise comme un vecteur d'entiers validVertices qui permettra de « taguer » les points tel que :

- validVertices(p)=1 si les champs de p sont calculés sur le processus courant.
- validVertices(p)=0 si les champs de p sont calculés sur un processus distant.

Deux méthodes simples existent pour effectuer cet ajout d'information :

- Surcharger la méthode defineMesh en ajoutant en argument les tags de chacun des points. La méthode permettra d'initialiser un attribut validVertices contenant ces tags et appellera la méthode defineMesh originale pour construire effectivement le maillage.
- Ajouter une méthode defineHollowedMesh fonctionnant de la même façon.

La deuxième solution a été choisie pour que l'utilisateur différencie bien l'utilisation d'un maillage dont certains points ne sont pas calculés localement sur la partition où ils sont définis de l'utilisation d'un maillage pour lequel tous les points d'une partition donnée sont calculés localement. De plus le fait de surcharger la méthode originale aurait pu engendrer des confusions pour l'utilisateur non familier avec la surcharge.

Une méthode privée completeField implémentant l'algorithme décrit précédemment a du aussi être ajoutée. Celle-ci est appelée par la méthode per-

mettant l'échange des champs.

Deux vecteurs attributs de l'objet coupling ont aussi été ajoutés pour permettre le stockage des indices des points dont les champs sont présents sur des processus distants.

Les constructeurs et destructeurs de l'objet coupling ont dû êtres modifiés en conséquence.

4.6.2 Besoin d'une définition d'un halo sur 2 points pour ARPEGE

Une des contraintes importante du calcul parallèle est d'obtenir des codes qui donnent exactement les mêmes résultats en mono-processus et en multi-processus. Grâce à ferret,des différences aux bords des partitions ont pu être localisées (figure 4.11). Celles-ci étaient dues au fait que pour la grille ARPEGE un halo de deux points en latitude était nécessaire du fait des mailles triangulaires (figure 4.12). Le domaine de complétion est donc étendu et la technique de complétion décrite en 4.6.1 est appliquée aux 2 points.

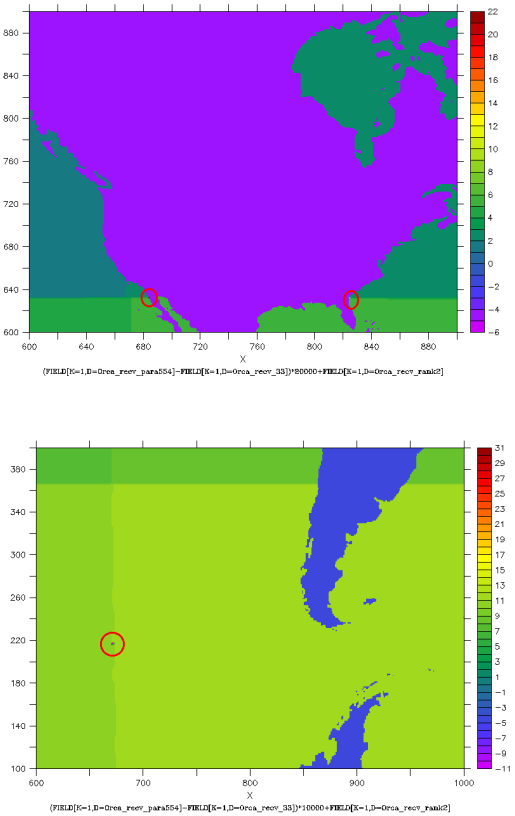


FIG. 4.11 – Localisation des différences entre les résultats mono-processus et parallèles.

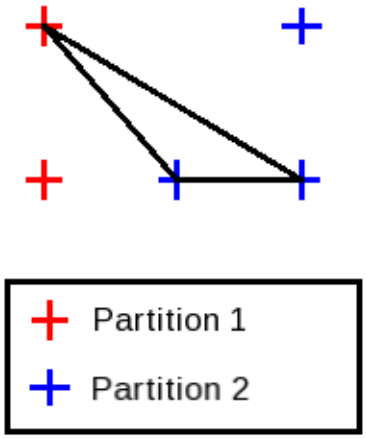


FIG. 4.12 – Un halo de deux points en latitude est nécessaire pour la construction de cette maille.

4.6.3 Points isolés

Certains points se retrouvent isolés notamment au niveau des lacs ou des golfes. Ceux-ci ne peuvent donc pas former de mailles triangles ou quadrangles. La résolution de ce problème a pris du temps, notamment dans le choix de la méthode à utiliser. Afin de résoudre ce problème les points seuls sont repérés et les points masqués les entourant sont considérés comme non masqués afin de former des mailles (voir figure 4.13). Les champs en ces points prennent la valeur du point non masqué.

Afin de permettre cela les tags des points ajoutés précédemment pour leur permettre d'aller chercher leur valeur chez un processus distant ont été étendus.

- validVertices(p)=1 si les champs de p sont calculés sur le processus courant.
- validVertices(p)=0 si les champs de p sont calculés sur un processus distant.
- validVertices(p)=n+1 si les champs de p doivent prendre la valeur des champs au point d'indice n.

Avant de faire l'interpolation la méthode completeField ajoutée à CWIPI(voir la section précédente) donnera la valeur qui correspond à chaque point selon son tag.

On obtient alors des meilleurs résultats au niveau des régions où se trouvent des points isolés (voir figure 4.14).

Néanmoins comme le montre la figure 4.15, même si globalement les résultats sont améliorés cette méthode détériore l'interpolation en certains points. Ceci peut être expliqué par la simplicité de l'extrapolation effectué qui peut apporter en certains points une dégradation. Mais l'amélioration obtenue par cette méthode l'emporte largement.

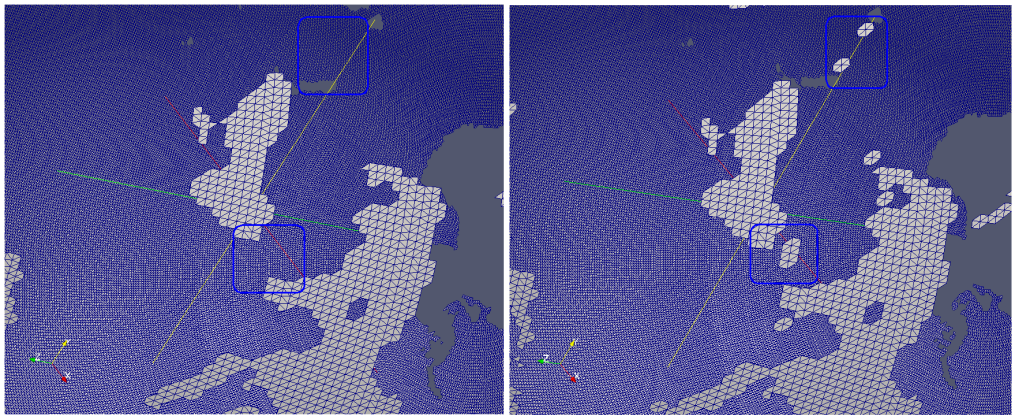


FIG. 4.13 – Différences de maillage avant et après traitement des points isolés(toutes les différences ne sont pas encadrées).

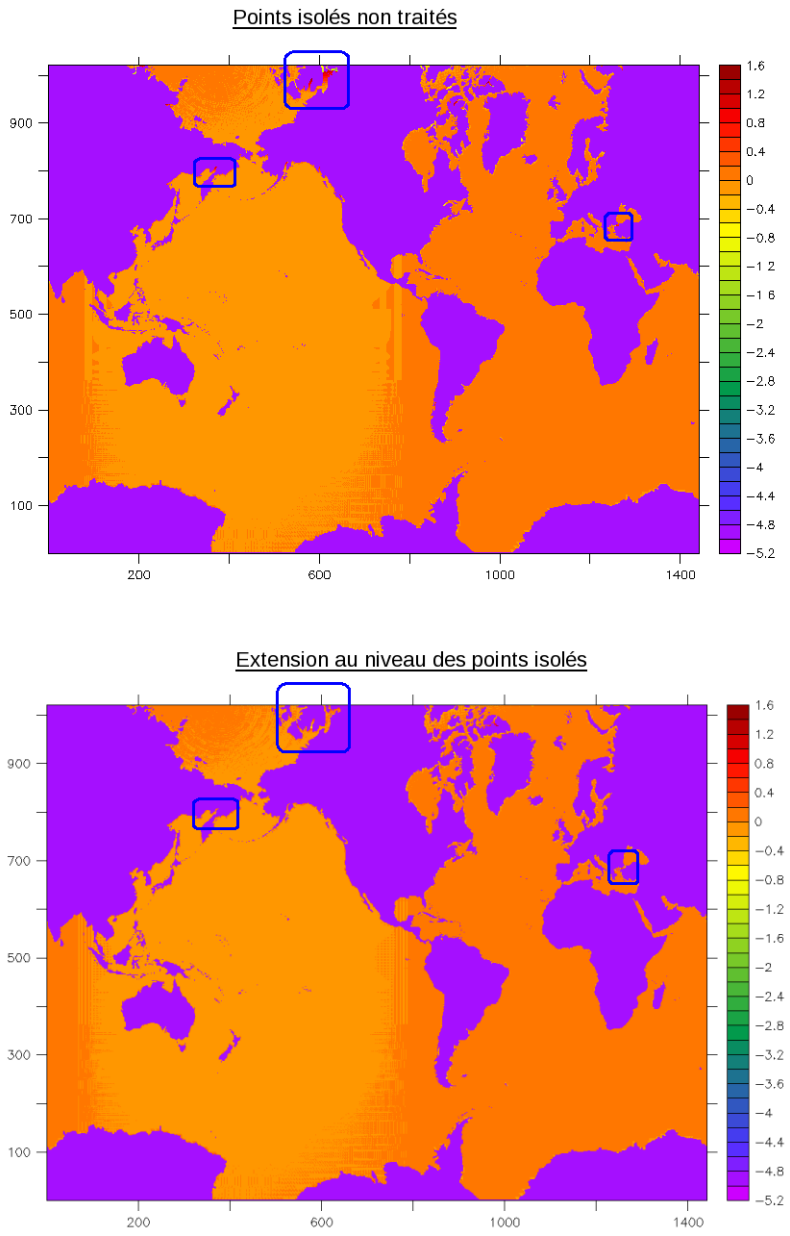


FIG. 4.14 – Cartes des erreurs d'interpolation avant et après traitement des points isolés.

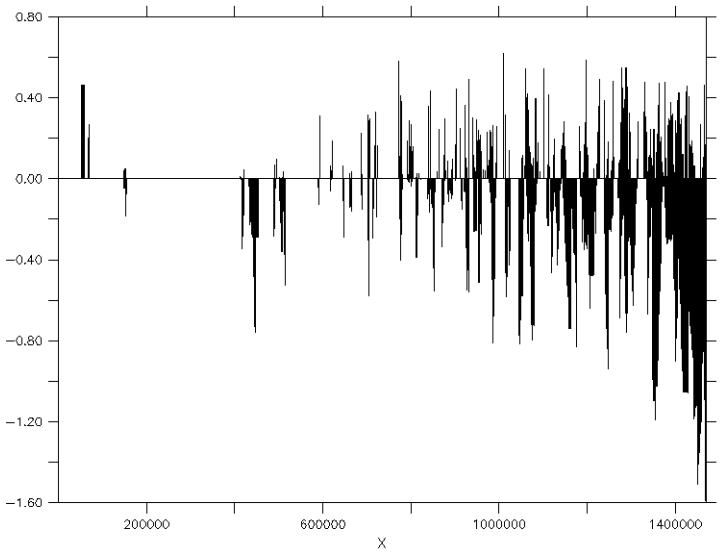


FIG. 4.15 – Différences de la valeur absolue de l'erreur avant et après traitement des points isolés (les améliorations correspondent aux valeurs négatives).

4.7 Performance du couplage

4.7.1 Méthode

Afin d'obtenir des mesures de temps précises des différentes sections de code la fonction MPI_Wtime() a été utilisée. Et afin d'éviter tout délai dû à des problèmes de synchronisation un appel à MPI_Barrier sur le communicateur commun aux deux codes de couplage est appelé avant chaque section dont on mesure la performance.

Les sections mesurées sont la section initialisation et les sections d' échanges. Le temps d'initialisation comprend la construction du maillage ainsi que l'allocation des poids. La durée d'un échange se calcule de l'envoi (PCW_Send) à la réception (PCW_Recv) d'un échange ping-pong d'un champ dans un même modèle et comprend donc à la fois le temps d'interpolation mais aussi celui des communications MPI. Afin d'obtenir des mesures de temps significatives, on effectue une centaine d'échanges dont on moyenne la durée.

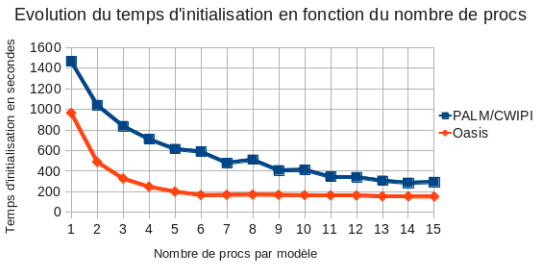
4.7.2 Résultats obtenus sur Octopus

Afin de limiter les cas, ce sont des partitions en bandes horizontales qui ont été utilisées pour ces tests.

On remarque que le temps d'initialisation est plus long avec OpenPALM/CWIPI mais que les échanges sont beaucoup plus rapides (voir figure 4.16). Or ce sont le temps des échanges qui sont les plus importants car l'initialisation n'a lieu qu'une fois pour des milliers d'échanges. Le coupleur OpenPALM interfacé avec CWIPI est donc plus performant pour le type de couplage que l'on souhaite réaliser(du moins sur ce type de machine). Ce sont des premiers tests qui seraient à étendre sur un nombre plus élevé de processeurs/cœurs.

Evolution du temps d'initialisation en fonction du nombre de processus par modèle

nombre de proc par modèle	PALM/CWIPI	Oasis
1	1466	965
2	1042	489
3	838	328
4	711	249
5	615	201
6	590	168
7	480	170
8	512	172
9	408	170
10	414	167
11	345	165
12	342	164
13	307	156
14	287	155
15	295	153



Evolution du temps d'échange en fonction du nombre de processus par modèle

nombre de proc par modèle	Palm/CWIPI	Oasis
1	0.2	0.94
2	0.17	0.65
3	0.17	0.5
4	0.15	0.46
5	0.1	0.42
6	0.1	0.4
7	0.13	0.38
8	0.08	0.35
9	0.1	0.34
10	0.11	0.33
11	0.066	0.31
12	0.074	0.3
13	0.09	0.3
14	0.1	0.29
15	0.068	0.3

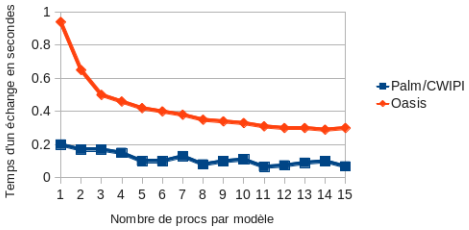


FIG. 4.16 – Performances observées sur Octopus.

4.8 Grilles icosaèdres

Les grilles icosaèdres[5] sont des grilles non structurées qui ne peuvent pas êtres traitées par le coupleur OASIS4. Lors de ce stage j'ai vérifié qu'il était bien possible de définir les maillages associés à ces grilles à la librairie d'interpolation CWIPI.Les mailles obtenues sont visualisées à l'aide du logiciel Paraview (logiciel open-source multiplateforme de visualisation et d'analyse de données) (figure 4.17 et 4.18).

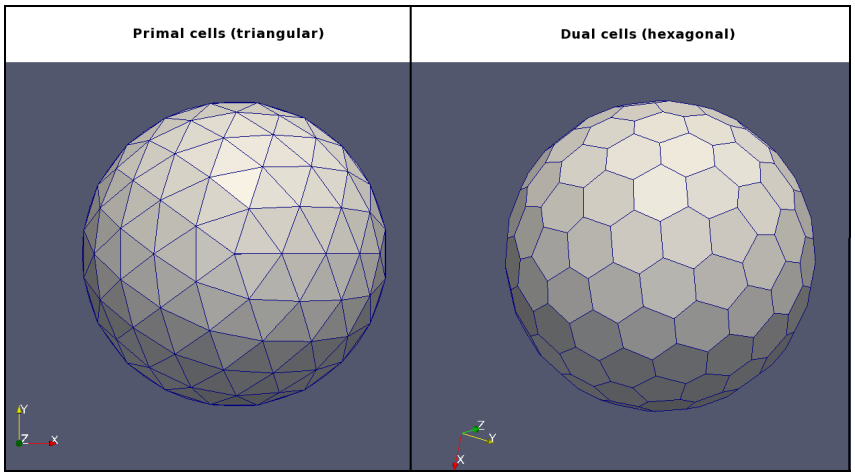


FIG. 4.17 – grilles icosaèdres.

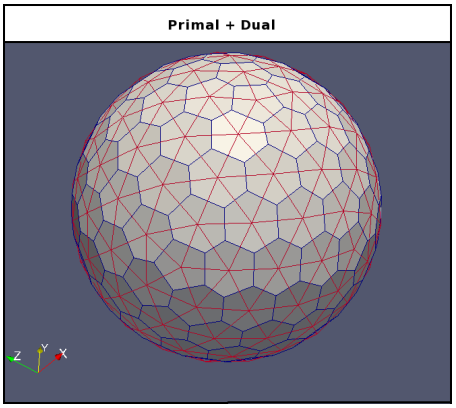


FIG. 4.18 – grilles icosaèdres superposées.

Chapitre 5

Conlusion

Ces 7 mois de stage m'ont apportés beaucoup de connaissances sur le problème du couplage et m'ont permis de mettre en pratique des concepts étudiés lors de ma troisième année à l'ENSEIRB en filière PRCD. L'encadrement par à la fois la partie OASIS et la partie PALM de l'équipe global change du CERFACS m'ont apportés différents points de vue sur le problème du couplage et du calcul parallèle. Les objectifs du stage ont été remplis, il a été néanmoins choisi de ne pas passer tout de suite au couplage du modèle réel. Ceci afin de se concentrer sur le développement des fonctionnalités qui manquaient au coupleur PALM/CWIPI pour la réalisation d'un couplage équivalent à celui obtenu avec OASIS. Différents algorithmes ont notamment été développés, pour le partitionnement, la gestion des bordures de partitions grâce à MPI (Message Passing Interface), la constructions des maillages et enfin la gestion des points isolés. De plus des tests de performance et de qualité d'interpolation ont été réalisés pour valider ces développement. Il restera néanmoins à étendre les tests sur des nombres de processeurs/cœurs plus importants et à analyser plus en détails la cohérence des résultats et en particulier vérifier que les erreurs importantes en certains points de grille sont bien justifiés par leur localisation et que l'on obtient bien les mêmes résultats pour n'importe quel nombre de processus utilisés.

Le travail de ce stage a montré à la communauté qu'un passage du coupleur OASIS au coupleur PALM/CWIPI serait possible. Cette transition éviterait au CERFACS d'avoir à maintenir et développer deux coupleurs. Le stage s'étant bien déroulé il débouche sur un CDD de 9 mois pour continuer les développements et coupler le modèle réel ARPEGE-NEMO.

Travailler au CERFACS m'a aussi indirectement fait découvrir la recherche en climatologie et notamment les outils informatiques et/ou mathématiques

qui sont utilisés dans ce domaine. L'émulation et la satisfaction de travailler dans une équipe de haut niveau scientifique m'ont conforté dans l'idée de m'orienter vers le métier d 'ingénieur de recherche. Idée qui avait émergée lors de mon stage de deuxième année au CEA.

Glossaire

CERFACS : Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique.

OASIS : Ocean-Atmosphere-Sea Ice-Soil.

PALM : Projet d'Assimilation par Logiciel Multi méthode.

CWIPI : Coupling With Interpolation Parallel Interface.

ARPEGE : Action de Recherche Petite Echelle Grande Echelle.

NEMO : Nucleus for European Modelling of the Ocean.

ORCA : Grille du modèle NEMO. Pour les modèles jouets le nom de la grille remplace le nom du modèle NEMO. On parle du modèle jouet ORCA.

Bibliographie

[1] *Coupleur O-PALM version 4.00 Manuel utilisateur et de formation*, CERFACS/Global Change and Climate Modelling Team, septembre 2011, T. Morel, A. Thévenin et A. Piacentini.

[2] *OASIS4 USER GUIDE*, octobre 2010, S. Valcke(CERFACS), R. Redler(MPI-M).

[3] *PART VI : TECHNICAL AND COMPUTATIONAL PROCEDURES*, janvier 2010, IFS documentation Cy36r1, ECMWF.

[4] *Coupling technologies for Earth System Modelling : Today and Tomorrow*, December 2011, Sophie Valcke/CERFACS, Rocky Dunlap/Georgia Tech.

[5] *ICON Grid Documentation* , Léonidas Linardakis, Daniel Reinert, and Almut Gassmann, Max-Planck-Institut für Meteorologie, Deutscher Wetterdienst.