
ECOLE NATIONALE
DES PONTS ET CHAUSSEES


LODYC


Ocean Modelling System
OPA
LODYC - IPSL

Sommaire

Résumé

Notre stage scientifique s'est déroulé dans un laboratoire d'océanographie (le LODYC) dans lequel les chercheurs étudient la dynamique de la circulation des masses d'eaux océaniques. Ils utilisent un modèle de calcul appelé OPA auquel on doit fournir en plus des conditions initiales et limites caractérisant le fluide une topographie des fonds marins. Cette topographie, appelée bathymétrie, est une modélisation par des éléments finis du relief des fonds océaniques et des limites géographiques des océans en surface. Notre travail a été d'écrire dans le langage de programmation IDL un programme de construction automatique de cette bathymétrie et de réécrire complètement le programme de retouche à la main de cette bathymétrie.
Nous avons cru bon d'exposer en guise d'introduction quelques notions de base concernant la dynamique océanique. Le reste concerne les programmes que nous avons écrits.
Nous mettons aussi à disposition et en complément à ce rapport de stage deux annexes ; la première concernant le code lui-même peut être utile aux initiés du langage IDL, la seconde est plus scientifique et s'adresse aux amateurs d'océanographie.

Remerciements

Nous remercions tous les chercheurs qui nous ont accueilli et plus particulièrement Sébastien Masson et Arnaud Jouzeau pour leur aide en IDL, ainsi que Robinson Hordoir et Edmée Durand, les deux ingénieurs du laboratoire qui nous ont aidé tout au long des trois mois. Merci à Julien Emile-Geay, stagiaire DEA, et à Henri Hay, stagiaire des Ponts et Chaussées, pour nous avoir supportés dans ce laboratoire. Et enfin merci à Gurvan Madec de nous avoir fait confiance et de nous avoir aidé malgré les innombrables sollicitations dont il fait chaque jour l'objet.

Mots clés

LODYC : Laboratoire d'Océanographie Dynamique et de Climatologie
OPA : modèle de calcul d'évolution des masses d'eaux océaniques
BATHYMETRIE : topographie des fonds marins en éléments finis
IDL : langage de programmation
OCEANOGRAPHIE
TOPOGRAPHIE DES FONDS MARINS

1. INTRODUCTION

1.1 Présentation du laboratoire d'accueil :

Le LODYC (Laboratoire d'Océanographie DYnamique et de Climatologie) est une Unité Mixte de Recherche dépendant du CNRS, département des Sciences de l'Univers, de l'Université Pierre et Marie Curie (Paris VI), et de l'I.R.D. (Institut de Recherche pour le Développement). Le LODYC fait partie de l' Institut Pierre-Simon Laplace et est implanté dans les locaux de l'Université Paris VI. Le laboratoire accueille du personnel de l'Université Versailles Saint-Quentin, de l'Université Paris VII et de Météo France. Il regroupe environ 80 personnes, parmi lesquelles 24 chercheurs et enseignants-chercheurs, 18 ingénieurs, techniciens et administratifs, et de nombreux étudiants en thèse et stagiaires, tous localisés à Jussieu.

L'axe principal des recherches du LODYC est l'étude des processus dynamiques gouvernant la circulation océanique et à terme la compréhension des mécanismes gouvernant l'évolution du système climatique terrestre dans lequel l'océan joue un rôle important à différentes échelles climatiques. Les recherches portent également sur l'étude des cycles bio-géochimiques océaniques, en particulier celui du carbone, qui mettent en jeu, entre autres, la biosphère marine.

Les recherches et les simulations sont effectuées grâce à un model d'océan couplé avec des modèles de glaces et d'atmosphère, le tout étant modélisé numériquement grâce au code informatique OPA, un modèle de circulation océanique général développé par le LODYC.

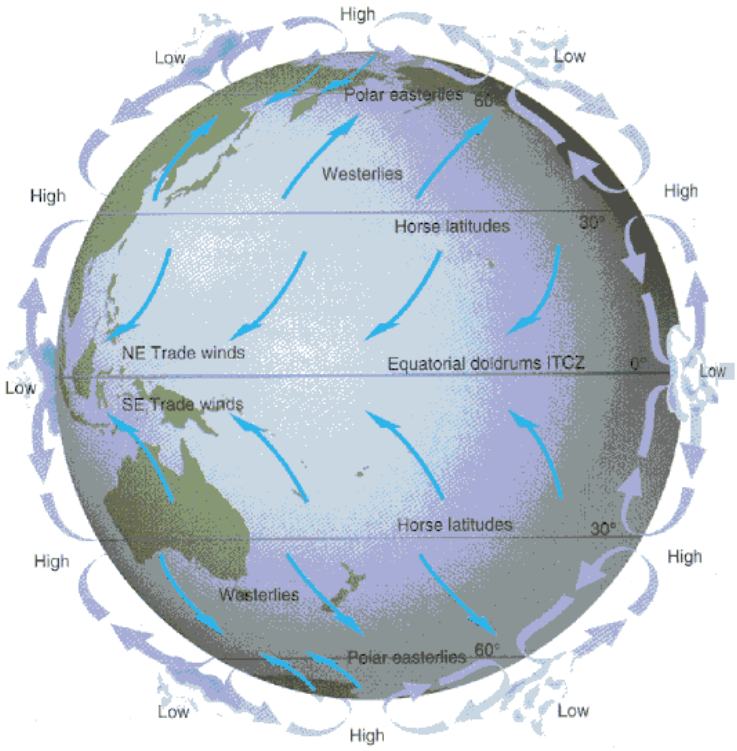
Les principaux projets du laboratoire relèvent :

- d'une activité expérimentale fondée sur l'implication forte du LODYC dans les campagnes en mer ;
- d'une activité de développement instrumental ;
- de l'interprétation conjointe des observations spatiales et in situ ;
- d'une activité de modélisation théorique, conceptuelle et statistique avancée, liée à l'interprétation des données ;
- d'une activité de modélisation numérique, le laboratoire ayant développé un modèle de circulation générale océanique.

Le laboratoire participe également activement à l'enseignement supérieur et l'accueil de thésards dans les domaines de l'océanographie, de la climatologie et de l'étude des cycles bio-géochimiques océaniques.

1.2 Notions de base

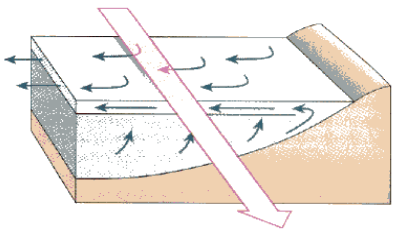
L'évolution du climat terrestre est déterminée par les courants océaniques et atmosphériques. L'océan stocke l'énergie solaire dans les couches superficielles équatoriales et transfert cette énergie à l'atmosphère au niveau des hautes et moyennes latitudes. Grâce à sa grande inertie, l'océan est le principal régulateur du climat terrestre. C'est en espérant comprendre et à plus long terme prévoir l'évolution de notre climat que des chercheurs étudient la circulation océanique, résultat de phénomènes physiques très complexes et couplés entre eux.

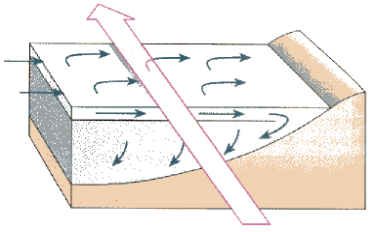


la circulation des vents et la force de Coriolis

La masse océanique subit notamment l'action du vent sur sa partie superficielle et celle de la force de coriolis. De celles-ci résulte un transport horizontal des masses d'eau de la couche supérieure de l'océan (les cents premiers mètres d'eau à compter de la surface). Ce transport – proportionnel à la tension de vent, i.e. l'intensité du frottement du vent sur la surface – a pour direction principale de déplacement la perpendiculaire à celle du vent (le courant est dextrogyre dans l'hémisphère Nord et lévogyre dans l'hémisphère Sud).

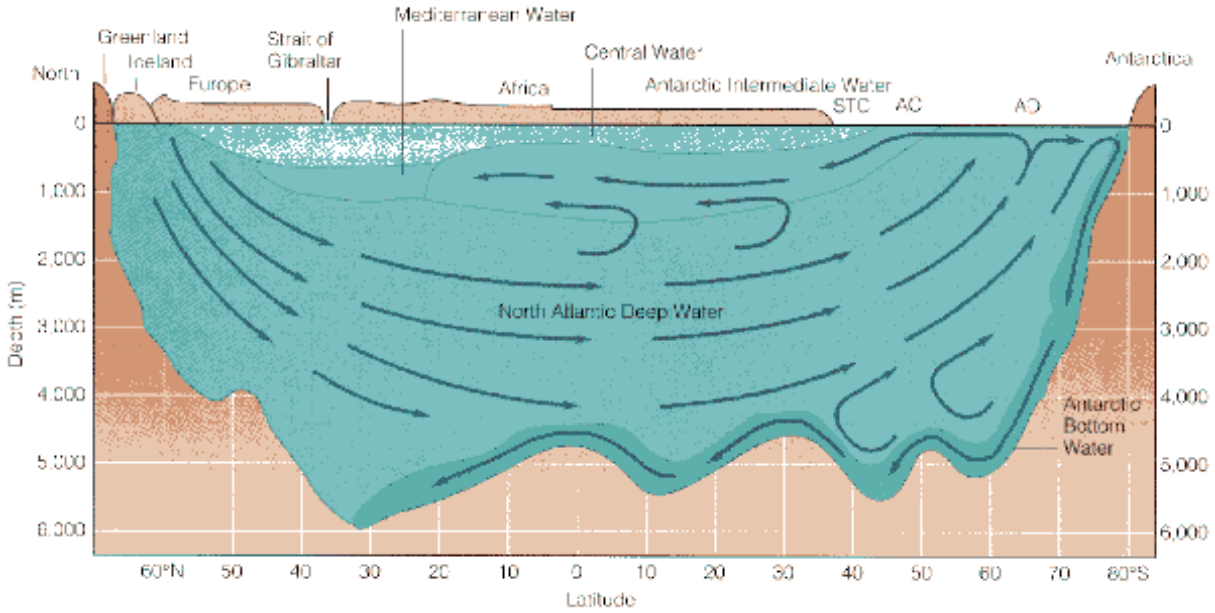
Lorsque le vent vient du nord, l'action de celui-ci et de la force de Coriolis entraîne les masses d'eau vers le large et celles-ci sont remplacées par des masses d'eau plus profondes.





A l'inverse lorsque le vent vient du sud, celui-ci combiné à la force de Coriolis pousse les masses d'eau contre le rivage et celles-ci plongent vers le fond.

La dynamique de ces courants entraîne une convection verticale des masses d'eau (phénomène appelé « pompage d'Ekman », cf. *annexe 2 du rapport de stage*). Les échanges de chaleur (conducto-convection et rayonnement) et d'eau chargée de particules (évaporation et précipitation) entre l'atmosphère et les différents courants marins créent une hétérogénéité spatiale de la température et de la salinité. Cette dernière associée à la variation d'altitude de la surface génère un gradient de pression horizontal variable en direction, sens et intensité dans toute la masse océanique. En causant une variation de la densité, elle génère aussi un gradient de pression horizontal, et la circulation de masses d'eau qui en découle est dite thermohaline.



Circulation atlantique des masses d'eau

Par exemple la boucle la plus lente de la circulation thermohaline, d'une période estimée à un millénaire, découle de la formation des glaces au pôle nord. Les eaux polaires étant ainsi plus denses plongent vers le fond pour former la couche d'eau profonde (voir figure ci-dessus). Elles se répandent vers le sud et remontent progressivement à la surface. Réchauffées aux tropiques et à l'équateur, elles regagnent les hautes latitudes polaires où elles sont refroidis par les vents extrêmement froids.

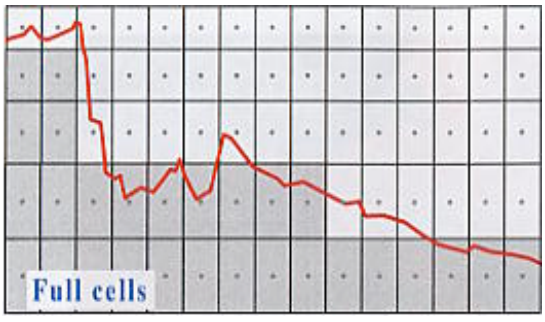
Le vent est l'un des facteurs principaux qui régissent la circulation des masses d'eau dans l'océan, mais en retour les différences de températures des eaux à la surface jouent sur la répartition des zones de haute et basse pression atmosphérique et donc sur le vent. C'est ce qu'on appelle le couplage océan / atmosphère que l'on étudie grâce à un outil informatique qui résout des équations à partir de bilans locaux pour déterminer une valeur approchée de la vitesse, de la salinité et de la température en tout point de l'océan. Le nom de cet outil d'OGCM (Ocean General Circulation Model) est OPA et il est couplé avec un modèle d'atmosphère (AGCM) pour déterminer les prévision climatiques. Il est également couplé avec des modèles de glace (présence des banquises et icebergs dans le cycle) et avec un modèle simulant l'écoulement des fleuves.

Vous pouvez retrouver ces explications en annexe 2 de notre rapport accompagnées de nombreux autres croquis et schémas.

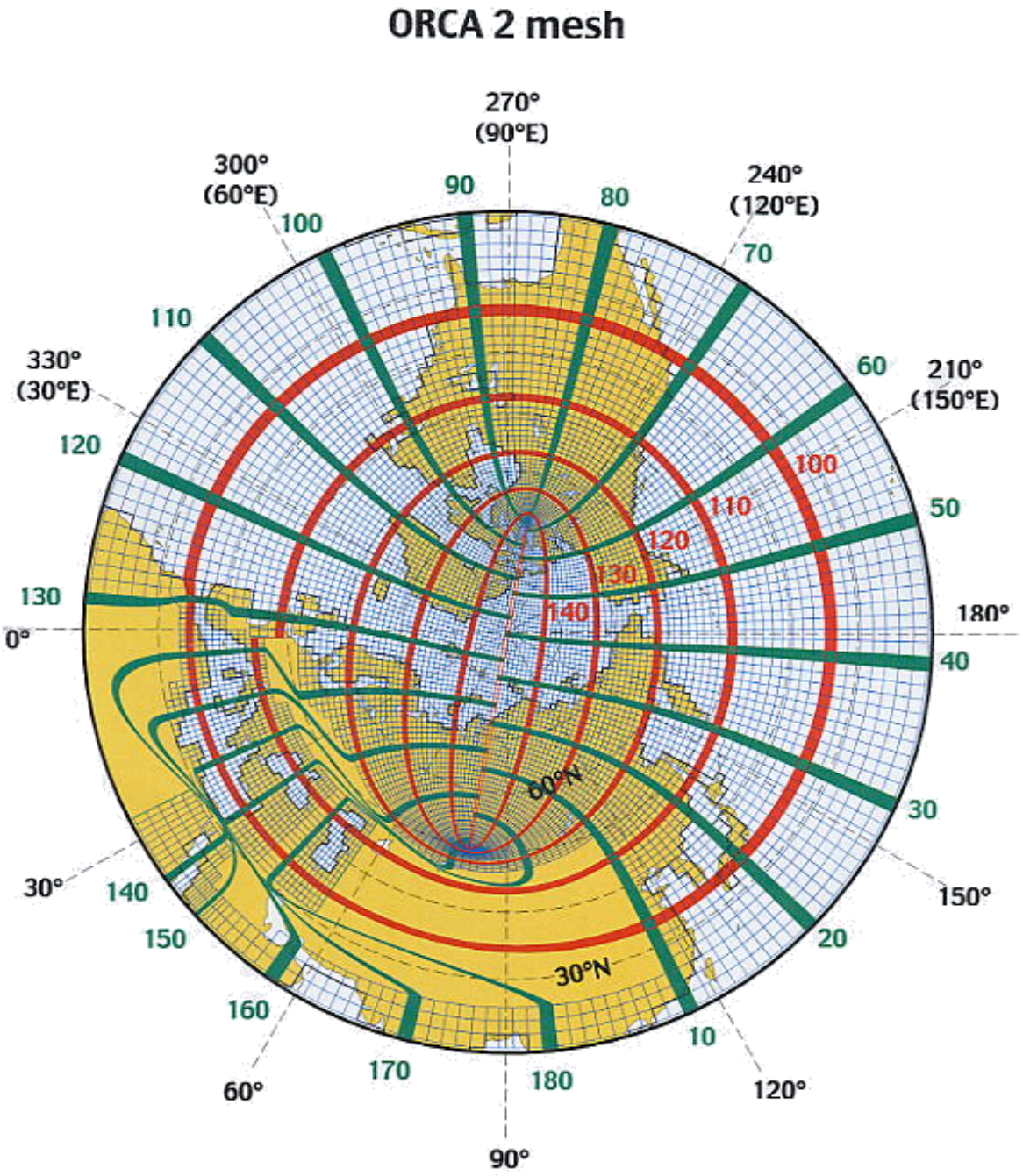
1.3 Le projet scientifique dans lequel notre travail prend place

OPA est un projet visant à modéliser le comportement dynamique de l'océan mondial. Pour cela on utilise une représentation de celui-ci composée d'un nombre fini d'éléments pour lesquels on applique les équations primitives des bilans locaux (matière, énergie…) par la méthode des différences finies. La représentation de son comportement dynamique est donc donnée par un modèle de calcul qui résout des équations différentielles sur une grille tridimensionnelle – un maillage de l'océan, si on utilise le vocabulaire des éléments finis – qui représente la forme physique de l'océan.

Le développement d'une grille dont les propriétés sont satisfaisantes pour la représentation de la réalité (quasi-isotropie sur l'océan mondial) et le bon fonctionnement du modèle (éléments cubiques de taille normalisée avec utilisation de rapport d'échelle lors du calcul) a été réalisé il y a deux ans. En effet le modèle initial de calcul ne pouvait manipuler qu'une grille très régulière et composée d'éléments entiers (cf schéma full cells). L'océan a donc été divisé en couches selon l'axe des profondeurs en quelques dizaines de niveaux d'épaisseur croissante avec la profondeur (quelques dizaines de mètres au voisinage de la surface et 500 mètres pour les derniers niveaux). Ce nombre assez faible de divisions a été dicté par le caractère limité des possibilités des ordinateurs de l'époque et d'aujourd'hui (le calcul de l'évolution pendant une année de l'océan sans couplage pour la résolution la plus grossière de la grille – la maille fait environ 200 km de coté - prend encore 34 minutes sur le super-calculateur du CNRS qui est une des plus puissantes machines de la communauté scientifique européenne). Quant à l'inégalité d'épaisseur des niveaux, elle correspond à la réalité des déplacements de masse d'eau dans l'océan (l'épaisseur des courants est croissante avec la profondeur).



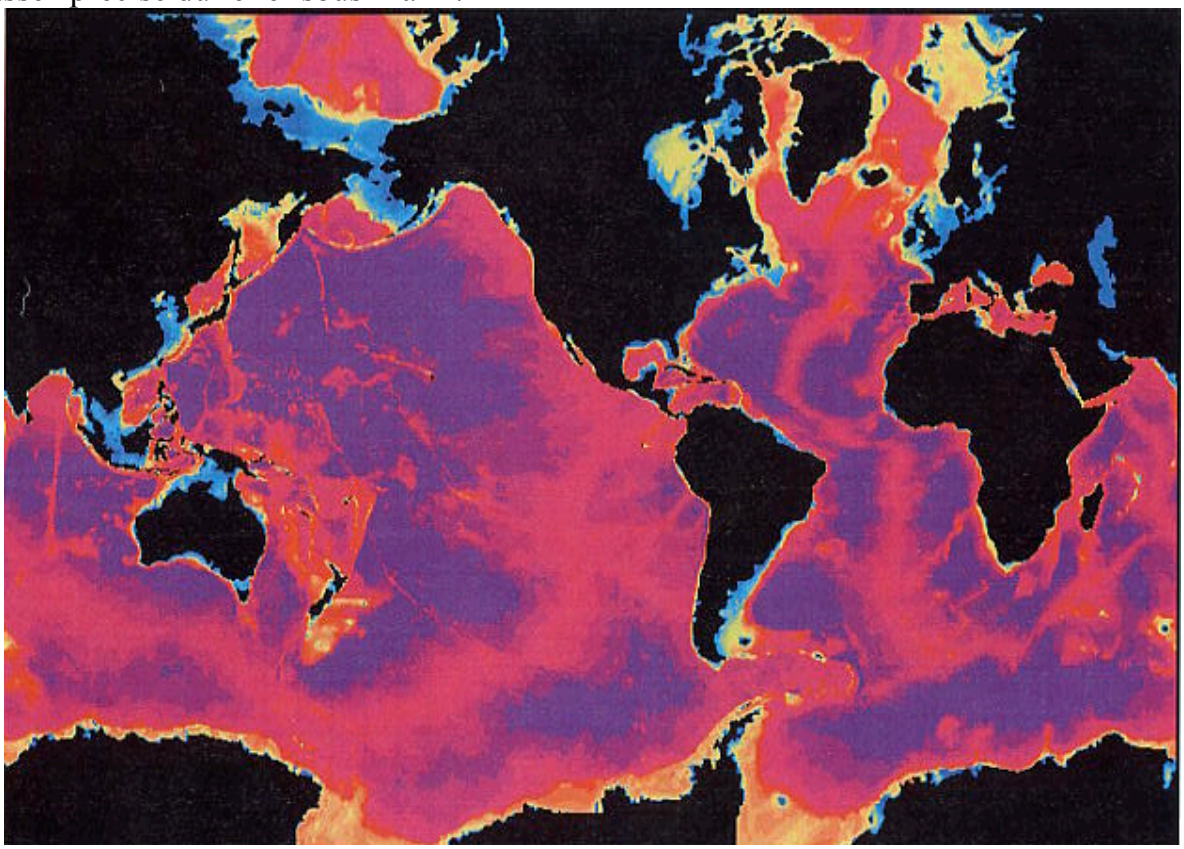
La construction des mailles horizontales, qui est en fait le vrai travail de construction de la grille, et qui était le sujet d'un précédent stage scientifique est explicité dans le rapport de stage de Christophe Bagot. La grille est valable quelque soit la profondeur puisqu'elle est définie en fonction de la latitude et de la longitude. Il faut donc fournir au modèle de calcul cette grille ; on construit un tableau d'entier bidimensionnel, chaque cellule représentant une maille de la grille, et l'entier qu'elle contient le nombre de niveaux d'océan de cette maille. Ce tableau est appelé bathymétrie. Il est essentiel au modèle de calcul mais surtout il a une grande influence sur sa cohérence avec la réalité.



Grille surfacique de l'océan à une précision de 2 degrés :

Cette bathymétrie est construite à partir de fichiers contenant des relevés de profondeur. Au départ, le programme OPABAT qui construisait les fichiers nécessaires à la réalisation de la bathymétrie était un ensemble de programmes FORTRAN pour la construction à partir des données et de programmes IDL pour la retouche à la main. En effet, étant donné la taille des mailles, même une moyenne des profondeurs (sans troncature en niveaux donnés) n'est pas suffisante pour représenter correctement la réalité étant donné le fonctionnement du modèle (par exemple pour calculer des flux latéraux dans un canal, il faut que le canal ait au moins deux mailles de large sinon il n'y a pas de gradient entraînant un flux). De même, sur une maille de 200 kilomètres de large la présence d'une faille océanique de 20 kilomètres de large n'affecte que très peu la profondeur moyenne alors qu'elle permet un important mouvement de masses d'eau profondes froides, ce qui a une grosse incidence sur la circulation océanique à l'échelle globale. Il faut donc retoucher à la main le tableau calculé de manière automatique.

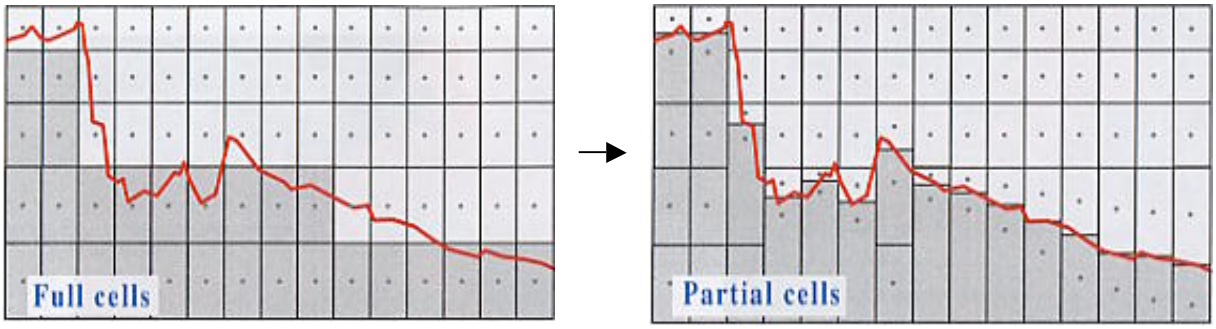
Cette opération est inévitable, fastidieuse et nécessite une connaissance complète des fonds océaniques, du fonctionnement du modèle et de sa sensibilité; cela suppose qu'on ait déjà fait tourner le modèle de nombreuses fois. Mais c'est à travers elle que ce manifeste réellement l'influence du scientifique sur la modélisation. Pour effectuer cette opération il faut avoir une idée assez précise de la forme des fonds marins à cet endroit, d'où la nécessité d'une bathymétrie à plus haute résolution. Dans la pratique lors de la retouche on affiche à l'écran la bathymétrie sur laquelle on travaille et par dessus les isobathes de la bathymétrie haute résolution, ce qui permet d'avoir une idée assez précise du relief sous-marin.



Bathymétrie demi-degré

Notre travail c'est concentré sur l'implémentation d'un nouvel ensemble de programmes pour construire automatiquement les bathymétries haute et basse résolutions de chaque précision (2 degrés puis 0.5 degrés) à partir des données. Il remplace l'ancien code en fortran et est écrit en IDL. Nous avons également réécrit le programme OPABAT de retouche à la main de la bathymétrie afin qu'il soit plus clair, plus fonctionnel, plus ergonomique et surtout plus évolutif et que la bathymétrie soit construite et retouchée en mètres entiers et non plus en niveaux. Le but avoué de notre stage était de passer des profondeurs en niveaux aux profondeurs réelles en mètres ; en d'autres termes il s'agissait de réécrire les programmes utilisant les full cells pour qu'ils mettent en œuvre des bathymétries en partial cells (fig. ci-dessous). En effet une amélioration récente du modèle de calcul lui permet de traiter des bathymétries dont le dernier niveau est de taille variable. On fournit donc au modèle une bathymétrie en mètres entiers et une grille de profondeur donnant les bornes de chaque niveau, celui-ci construit autant de niveaux entiers que possible puis il attribue la profondeur restante au dernier niveau ; par exemple si un point de grille est à la profondeur 35 mètres, sachant que le troisième niveau commence à trente mètres et le quatrième à soixante, le modèle va attribuer à ce point de grille deux premiers niveaux entiers et un troisième partiel de 5 mètres d' « épaisseur » . Celle des derniers niveaux étant de 500 mètres, l'augmentation de la précision est appréciable.

Une fois les calculs effectués pour construire les bathymétries, nous avons pu retoucher avec Gurvan Madec les seuils et détroits importants sur la bathymétrie à 2 degrés de résolution (voir annexe 2).



Passer des full cells aux partial cells comme il nous était demandé permet de mieux épouser la topographie des fonds marins

1.4 Les principales contraintes matérielles rencontrées

La principale difficulté de l'écriture de ce code reste la manipulation de tableaux très volumineux, qui amène rapidement les super calculateurs à leurs limites en place mémoire et en temps CPU (temps de calcul). Le programme initial était en fait uniquement conçu pour la bathymétrie à deux degrés et au demi-degré et la plupart de ces routines étaient inadaptables directement (par exemple la lecture des fichiers de basse résolution comprenait un affichage complet à l'aide d'une palette de couleur, mais dès le travail sur la demi-degré un tel affichage prend un temps dissuasif). De même la solution adoptée pour la sauvegarde était de garder en mémoire le tableau initial complet et le tableau modifié complet, ce qui dépasse les capacités mémoires de Rhodes (c'est la frontale du super calculateur du CNRS sur laquelle nous avons fait tourner nos programmes). Nous exposerons plus loin les problèmes rencontrés et les solutions adoptées pour chaque routine que nous avons modifiée ou complètement réécrite.

De plus nous travaillions directement sur le super calculateur et sur la base de données du CNRS, nous avons donc rencontré d'autres limitations qui proviennent exclusivement du facteur humain. En effet pour permettre au plus grand nombre de chercheurs de profiter de ces ressources, les accès et la soumission de travaux sont réglementés et organisés ; par exemple les travaux sont traités dans l'ordre de soumission. Mais pour pouvoir faire tourner nos programmes avec les ressources mémoires suffisantes, nous utilisons un programme qui nous affecte de manière constante une certaine taille mémoire (la ressource totale était de 20 gigas et nous travaillions généralement avec 4 gigas). Cependant, toujours pour favoriser le plus grand nombre, la durée maximale de cette affectation est une heure. Certains de nos programmes nécessitent une telle mémoire pendant bien plus longtemps d'où la nécessité de prévoir une fonctionnalité qui tienne à jour l'avancée du programme pour reprendre là où il s'est arrêté. D'autre part même une telle ressource mémoire n'est pas toujours suffisante ni très rapide. Nous avons donc dû souvent traiter les fichiers morceaux par morceaux, en répétant pour chaque pavé : lecture des données, mise en mémoire, traitement, sauvegarde. Cependant l'excès opposé qui serait de traiter chaque point de grille à son tour n'est pas efficient non plus. En effet le temps d'accès au disque est cent fois supérieur à celui de la mémoire vive, ce qui rend une telle solution beaucoup trop lente. Il nous faut aussi signaler que les performances d'IDL quant aux opérations vectorisées sont sans commune mesure avec sa difficulté à traiter efficacement les boucles « for ». Cependant IDL est le logiciel le plus efficace actuellement pour ce genre de programme. Nous avons donc utilisé une solution mixte pour tenter d'optimiser le fonctionnement des programmes.

A contrario, la souplesse d'utilisation des fichiers netcdf nous a permis de mettre en œuvre des solutions beaucoup plus efficaces qu'avec n'importe quel autre format. En effet ce genre de fichiers est construit comme un tableau à plusieurs dimensions (par exemple pour la grille d'océan utilisée par le modèle : trois dimensions d'espace et une de temps) comprenant autant de variables que l'on désire. Par contre il faut définir au moment de sa création les dimensions puis déclarer les variables et leurs attributs (unités, valeur limites, valeur attribuée par défaut…) en fonction de ces dimensions, et ces variables sont forcément définies sur la totalité du tableau. On peut par la suite rajouter des variables. En outre cette structure permet de lire et d'écrire des parties du tableau indépendamment les unes des autres en prenant en compte une ou plusieurs dimensions au choix en utilisant les options offset et count. C'est cette facilité qui nous a permis de concevoir des programmes moins gourmands en mémoire et suffisamment rapides.

2. Présentation des programmes

La version initiale

2.1 Son fonctionnement

Les programmes que nous avons retouchés, dont l'ensemble est nommé OPABAT, utilisaient en entrée deux fichiers de bathymétrie :

- un fichier de format binaire (global.bin) qui contient les données de profondeur d'une bathymétrie haute résolution.
- un fichier de format ascii (bathy_orca2_v2.ascii) qui est un tableau de 182 par 149 points contenant les profondeurs de la bathymétrie basse résolution à 2 degrés de précision.

L'utilisation de deux formats différents rendait la lecture de ces fichiers peu aisée (il y avait deux sous routines de lecture **batlec** et **batlec2** bien distinctes). De plus la lecture et l'enregistrement d'un fichier au format ascii est une opération longue et qui nécessite quelques acrobaties algorithmiques car un tableau de 182*149 points - taille du tableau pour la résolution 2 degrés - ne tient pas dans la largeur (les éditeurs de textes comme emacs ou vi n'admettent que 120 caractères par ligne) et donc les tableaux étaient scindés à l'intérieur de ce fichier. Il fallait de plus passer des profondeurs en niveaux (2 caractères, de 0 au niveau 30) à des profondeurs en mètres (4 caractères, de 0 à 9999 mètres) ce qui change la structure du fichier ascii. A l'origine il était écrit en format ascii pour être facilement visualisable et modifiable avec un éditeur de texte (le faible nombre de valeurs possibles rendait « visible » le relief, mais avec des profondeurs en mètres et seulement 20 points de grille par ligne la visualisation n'était plus efficace).

Après initialisation des variables communes (**@batini**), le programme **batexp** lisait les fichiers (en faisant appel aux sous-routines **batlec** et **batlec2**) et proposait le positionnement d'un zoom pour la retouche. L'étape de modification sur ce zoom présélectionné était effectuée par le programme **batmod** : ce programme affiche le zoom en attribuant une couleur à chaque point de grille. Cette couleur dépendait de la valeur dans le tableau de ce point de grille et du niveau de travail choisi avant l'affichage dont les pavés s'affichaient en jaune avec un code de couleur pour les niveaux supérieurs et inférieurs. Ce code était explicité en légende dans une petite fenêtre mais n'était pas évolutif (il était écrit « reference level », « level -1 », « level +2 »).

Enfin la procédure **batisl** enregistrait les modifications et attribuait aux terres émergées (préalablement au niveau 0) une valeur négative distincte pour chacune d'entre elles, opération nécessaire au bon fonctionnement du code OPA. Le problème est que cette dernière opération, effectuée par la routine **batisl**, n'était pas au point (l'algorithme ne remplissait pas entièrement les îles avec les valeurs négatives demandées) et il fallait retoucher le remplissage à la main avec un éditeur de texte (**batmod** n'accepte pas les valeurs négatives).

Les programmes de construction automatique des bathymétries écrits en Fortran que nous avons retrouvés n'étaient pas la version définitive, donc n'étaient pas débugués ; il a fallu les réécrire.

2.2 Ses limites

Les programmes de OPABAT comportaient encore beaucoup de lignes de code inusitées (séquences abandonnées…) et des programmes en double. Par exemple pour zoomer sur une nouvelles partie de la grille, il fallait enregistrer les modifications effectuées sur le précédent zoom et relire les fichiers puisque la lecture de ceux-ci et le positionnement du zoom étaient inséparables dans la procédure **batexp**. Autre difficulté : pour changer de précision (de 2 à 0.5 degrés) il fallait rentrer dans le code source et modifier à l'intérieur de celui-ci les variables de dimension des tableaux ; ceci était fastidieux et souvent sans succès pour une personne ne connaissant pas bien le code et surtout les endroits où apparaissaient les variables concernées. Par exemple lors de notre arrivée nous avons appris la structure du programme ainsi que le langage utilisé, le concepteur du programme a en effet quitté le laboratoire.

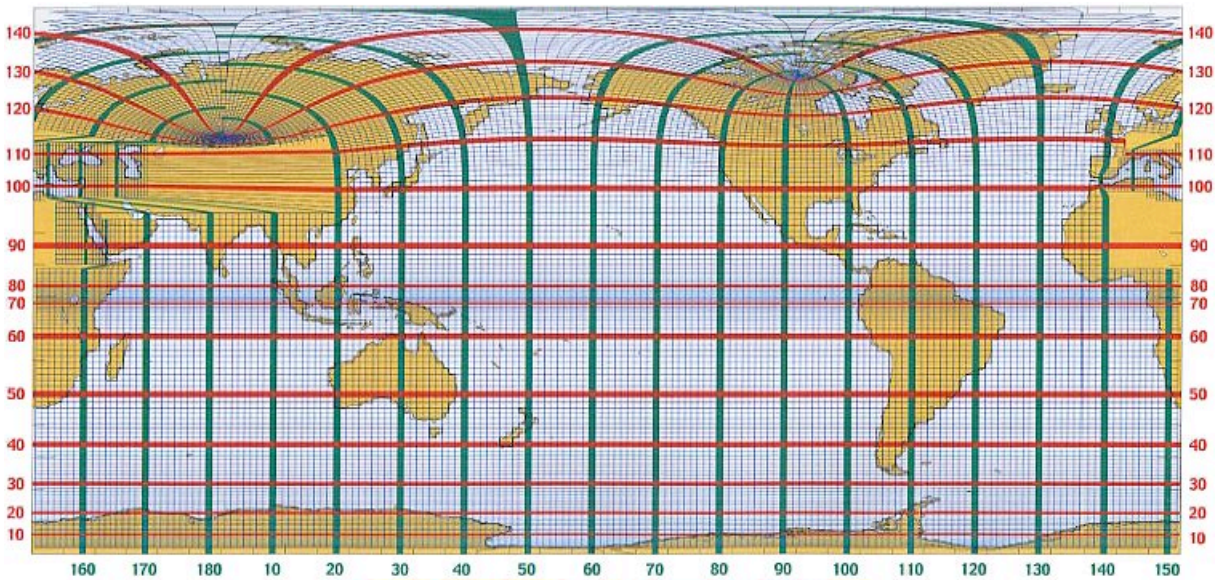
2.3 Les changements

De nombreux changements étaient donc nécessaires ou utiles, comme le fait de pouvoir zoomer à tout moment, de pouvoir choisir au démarrage la résolution du modèle, de bénéficier de plusieurs échelles de zoom et d'une légende évolutive qui affiche le code couleur en fonction des profondeurs en mètres.
Par ailleurs le passage prévu à des résolutions beaucoup plus précises dans un avenir proche – quart de degré dans l'année et dixième de degré dans les cinq ans - nous ont poussé à concevoir et à développer un programme le plus automatisé possible. De plus, son fonctionnement devait être indépendant de la taille du tableau de bathymétrie. Par conséquent, le traitement de bathymétries pour une résolution supérieure ne nécessitera aucune intervention dans le code si ce n'est pour rajouter des choix dans les initialisations proposées ; en fait il s'agit d'éviter la présence de dimensionnement interne au code ou de partition du travail propre à une résolution donnée.

2.4 Son fonctionnement général

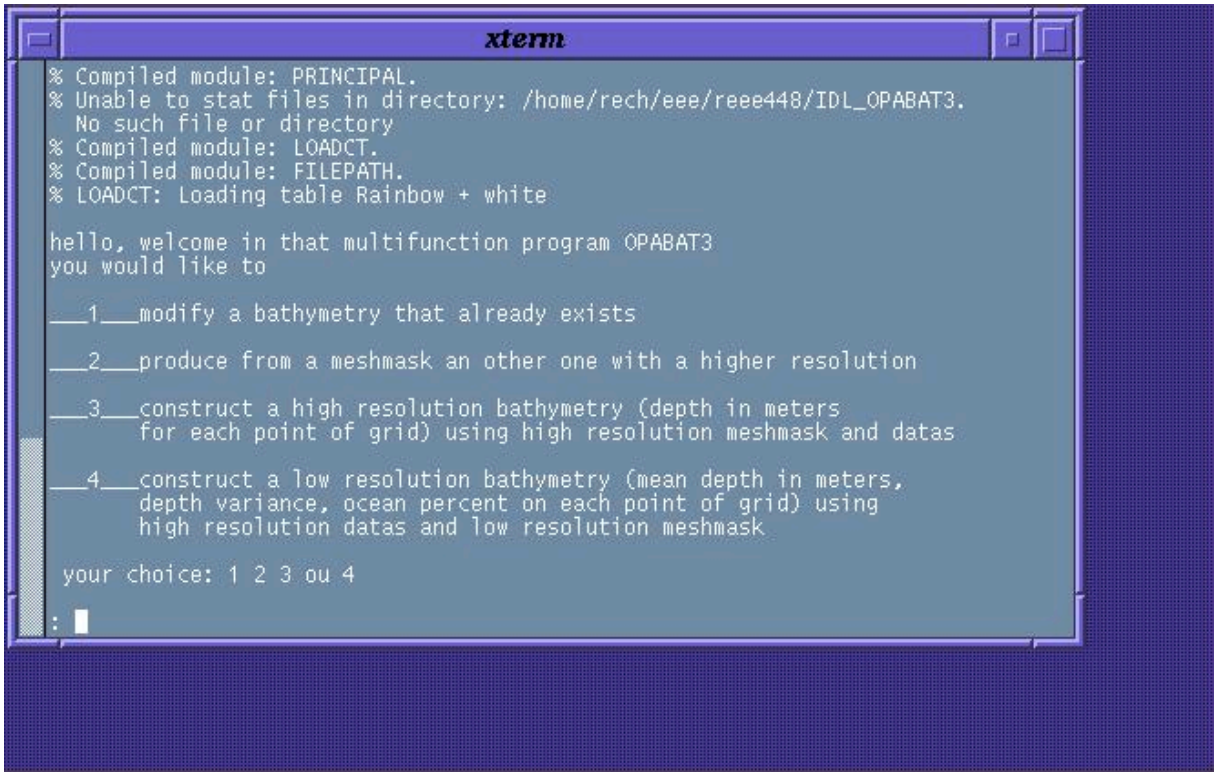
Les données utilisées

Des mesures effectuées par satellite et in situ grâce à des campagnes en mer permettent de recueillir des données de profondeur dans le monde entier. Mais pour les mesures satellitaires comme pour les mesures sur place la zone couverte est limitée. Le fichier le plus précis disponible actuellement est le Smith et Sandwell. Il contient une grille de relevés de profondeur sur toute la surface du globe pour les latitudes comprises entre 70 degrés nord et 70 degrés sud, pour des points situés toutes les trente secondes (une minute est une mesure angulaire qui correspond à un soixantième de degré et une seconde à un soixantième de minute). Pour les zones arctiques et antarctiques on utilise des fichiers de relevés qui ont une précision moindre. On constitue donc un fichier global (pour les latitudes entre 78 degrés sud et 90 degrés nord, les latitudes manquantes correspondent à l'antarctique) en « recollant » les mesures. Il y a donc une irrégularité au niveau du recollement qui est généralement assez visible dès qu'on affiche le fichier. Pour notre part nous avions projeté de constituer ce fichier avec une précision la plus grande possible, nous nous sommes aperçus que le pendant anglais du LODYC avait constitué un fichier de ce type à partir de Smith et Sandwell et de ETOPO5 avec une précision de 2 minutes. A notre demande, ils nous l'ont envoyé. A partir de ce fichier au format netcdf le programme doit construire automatiquement les bathymétries haute et basse résolutions. Cependant les mailles des grilles ne sont pas régulières et ne suivent pas les coordonnées géographiques. C'est pourquoi on doit posséder un fichier contenant la définition des mailles de la grille en fonction de celles-ci. C'est justement la production de ce fichier qui était le sujet du stage de Christophe Bagot, on l'appelle meshmask. Son programme a construit de manière analytique le meshmask des grilles deux degré et demi-degré.



Grille de l'océan constituant le meshmask

Toutes les procédures utilisables sont appelées par **principal**. **Principal** est donc le programme qu'il faut lancer en premier, c'est lui qui initialise (**@init**) les chemins d'accès aux routines IDL de Sébastien Masson (dont on s'est servi dans nos programmes) et c'est lui qui met les variables communes en mémoire (**@common** et **@commonbath**).



menu de démarrage de principal,
premier choix de l'utilisateur

A tout moment l'utilisateur peut lancer principal pour avoir à nouveau accès aux quatre choix :

- modifier une bathymétrie existante
- construire un nouveau meshmask plus fin
- construire une bathymétrie haute résolution
- construire une bathymétrie basse résolution

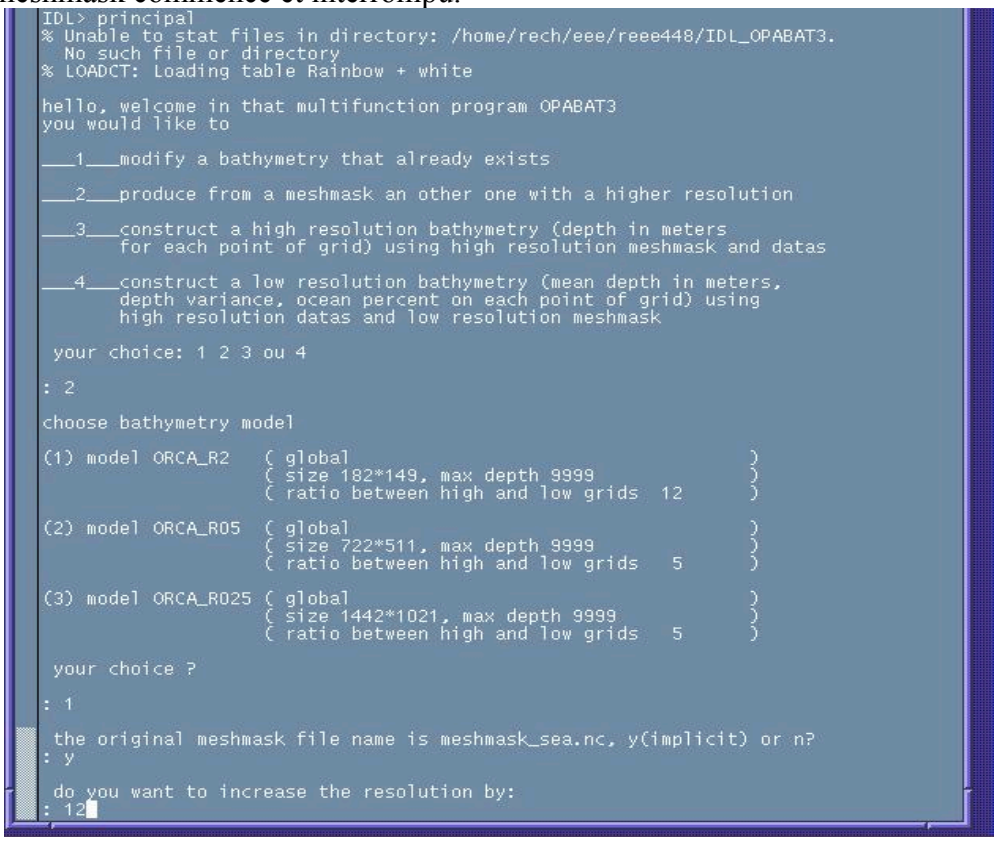
A partir de principal, on peut donc :

- construire la grille (meshmask , haute résolution puis basse résolution)
- retoucher cette grille 'à la main'

La construction de la grille

On utilise le masque (meshmask) pour interpoler ou faire une moyenne sur les valeurs du fichier de données. On ne possède cependant pas de meshmask pour les hautes résolutions de chaque grille, en effet leur meshmask correspond à la définition de la grille au sixième de degré et au dixième de degré. Mais la production analytique d'une grille est coûteuse et nécessite de nombreux ajustements manuels. De plus, étant donné que la seule utilité des bathymétries haute résolution est le tracé des isobathes lors de la retouche manuelle, c'est à dire de donner une idée de relief sous-marin, on considère qu'il n'est pas nécessaire qu'elle soit définie aussi exactement que celle de la basse résolution, on produit donc un meshmask directement à partir de celui de la basse résolution. Il faut donc commencer par fabriquer le meshmask pour la bathymétrie haute résolution :

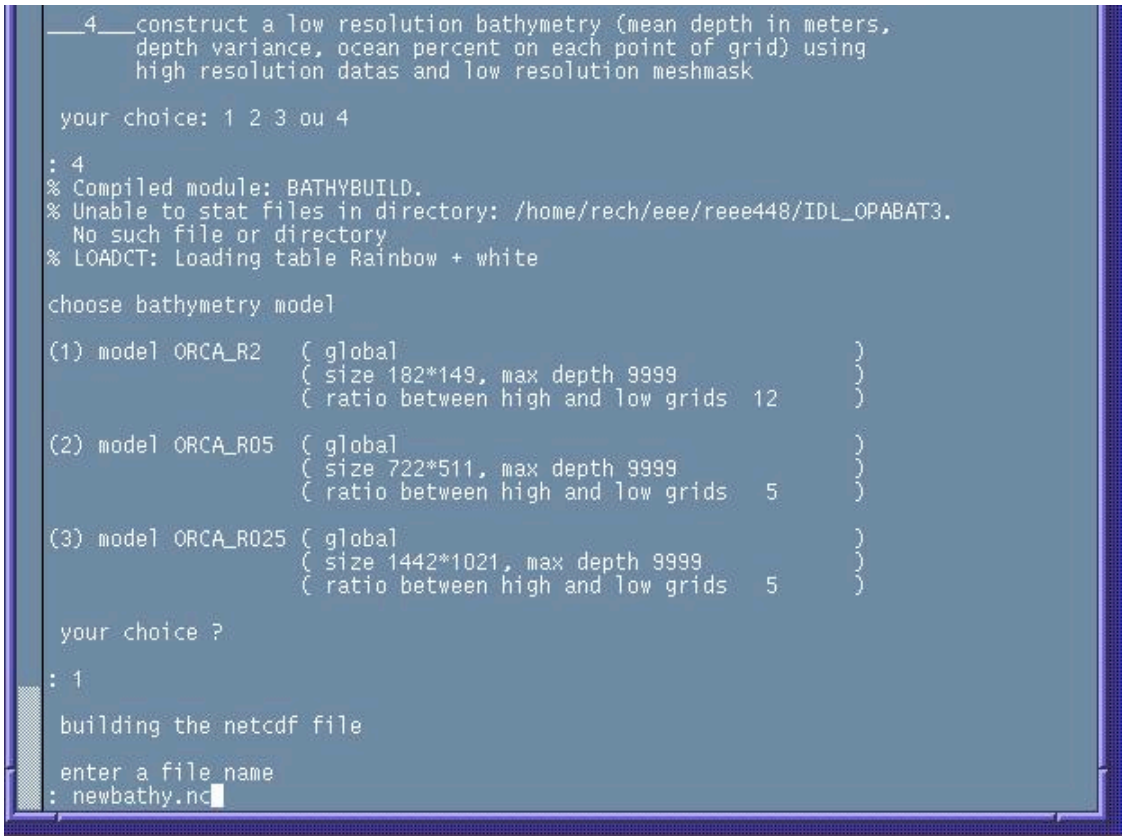
- Les programmes **mesh**, **meshdemi** et **mesh2** appelés par **meshmain** sont les programmes qui produisent ce meshmask haute résolution. On fabrique ces meshmasks par une simple interpolation linéaire sur les longitudes et les latitudes. Ce qui revient à considérer que la grille est suffisamment régulière. Ainsi la résolution du meshmask est augmentée linéairement par 5 pour le demi degré et par 12 pour le 2 degrés. Le calcul du meshmask haute résolution 2 degrés se fait directement, celui du meshmask haute résolution demi degré pose des problèmes de mémoire. C'est pourquoi une première procédure (**meshdemi**) construit un fichier .netcdf vide et lance une sub-routine contenant une boucle qui lit, interpole et enregistre les données pavés par pavés. Pour les problèmes de limitation temporelle sur rhodes expliqués précédemment, la procédure possèdent donc une fonctionnalité permettant de reprendre un calcul de meshmask commencé et interrompu.



création d'un meshmask 12 fois plus fin

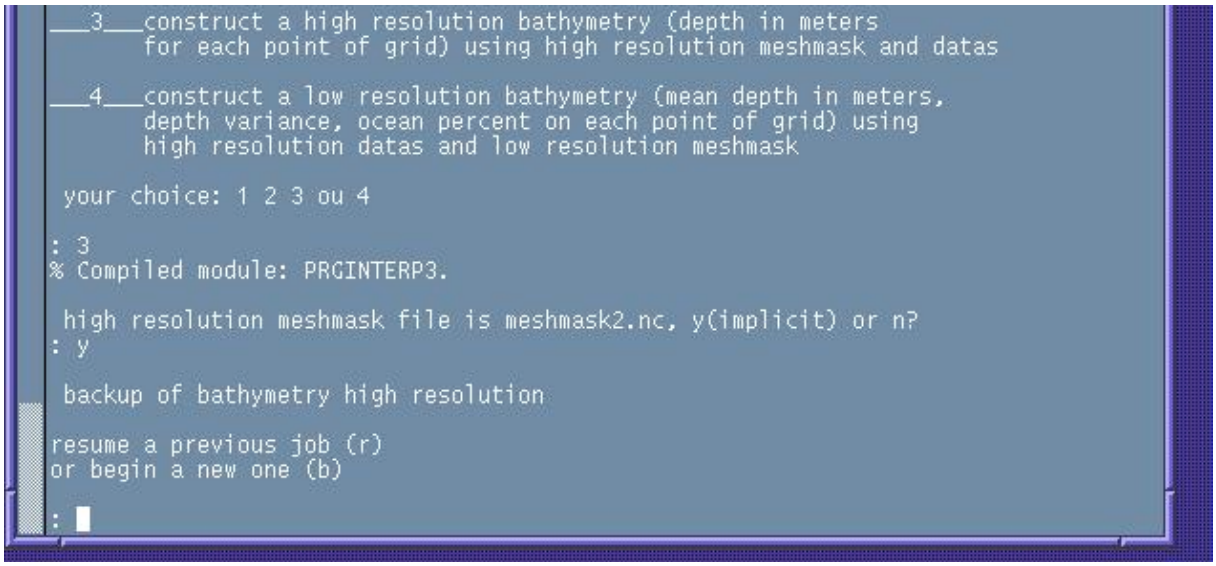
Une fois tous les meshmasks en main, les programmes **bathybuild** et **prginterp3** permettent de construire respectivement les fichiers de bathymétrie basse et haute résolution :

- **Bathybuild** crée un fichier netcdf et lance une double boucle (une sur les lignes puis une sur les colonnes) sur les points de la grille en appelant pour chacun d'entre eux la fonction **bathymean**. Cette procédure repère les limites géographiques de la maille auquel le point de grille correspond puis elle calcule différentes valeurs sur l'ensemble des points du fichier de données qui sont dans ces limites : la moyenne des profondeurs, sa variance et le calcul du pourcentage d'océan. En sortie de ces procédures nous obtenons la bathymétrie basse résolution (à 2 degrés ou au demi degré de résolution) au format netcdf contenant 4 variables : la profondeur de la basse résolution (depth) – la procédure attribue a cette variable la valeur moyenne - , la profondeur moyenne (depmean), la variance (depvar) et le pourcentage terre-mer (watperc).



Construire une bathymétrie basse résolution avec bathybuild

- **Prginterp3** est le principal programme responsable de la construction de la bathymétrie haute résolution. Il appelle **prginterp2** qui lui-même appelle **prginterp1**. Cet emboîtement des taches sert à optimiser le fonctionnement. Cependant le programme est encore trop long pour être exécuté en une seule fois, il y a donc un programme (**inclec**) qui suit l'avancement du programme. On peut alors relancer l'opération là où elle s'est arrêtée, cela se fait grâce à un choix au début du programme, le programme reprend alors là où il s'était arrêté. L'opération étant très gourmande en mémoire, le calcul s'effectue sur un pavé de mailles (dont le nombre optimal est calculé à l'intérieur de ces programmes) et à l'intérieur d'une double boucle sur les lignes et les colonnes. **Prginterp3** charge en mémoire les latitudes et les longitudes du meshmask haute résolution. Puis il appelle successivement sur chaque pavé **prginterp2**. Celui-ci applique sur chaque maille du pavé la fonction **prginterp1** puis il enregistre pour ce pavé uniquement les valeurs dans le fichier netcdf, il n'y a donc en mémoire que les valeurs du pavé. La fonction **prginterp1** repère les quatre points les plus proches dans le fichier de données et fait l'interpolation. Cependant les boucles coûtent en temps CPU, c'est pourquoi le calcul s'arrête plusieurs fois avant la fin (2 ou 3 fois). Lorsque l'incrément arrive au terme des lignes et colonnes calculées, la bathymétrie haute résolution est prête et on peut utiliser le programme OPABAT pour modifier à la main certain seuils et détroits.

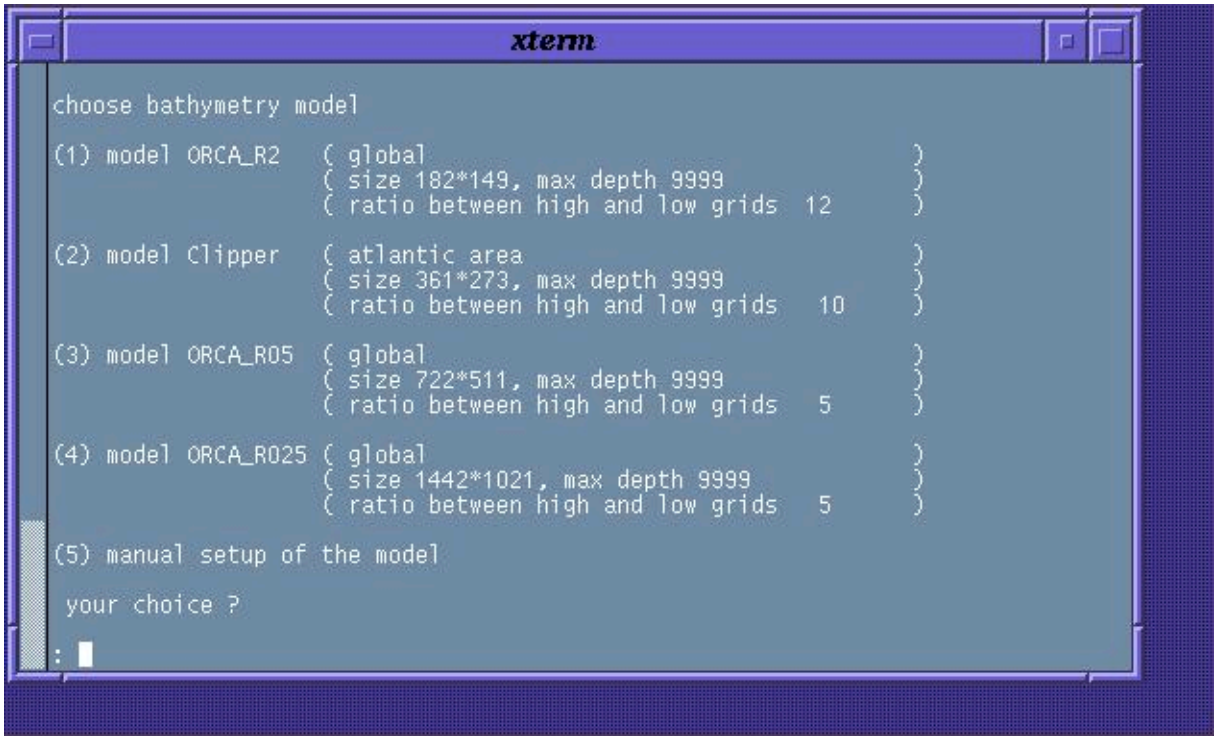


grâce à prginterp3 on peut lancer un nouveau calcul
de bathymétrie haute résolution ou en reprendre un autre

La retouche de la bathymétrie :

L'idée pour retoucher une bathymétrie est de zoomer sur une partie du globe afin de voir à une échelle correcte les mailles de la grille. On peut alors travailler à différentes profondeurs sur cette zone à l'aide d'un code de couleurs qui met en évidence une profondeur demandée. Sur ce zoom, OPABAT affiche les pavés de couleur de la basse résolution ainsi que les isobathes calculées sur les données de la haute résolution. Ces isobathes permettent de détecter les seuils et les failles invisibles sur la basse résolution alors que ceux-ci sont primordiaux pour les échanges d'eaux profondes entre les bassins. **Batini** est la procédure qui se lance quand on choisit la retouche dans le programme principal.

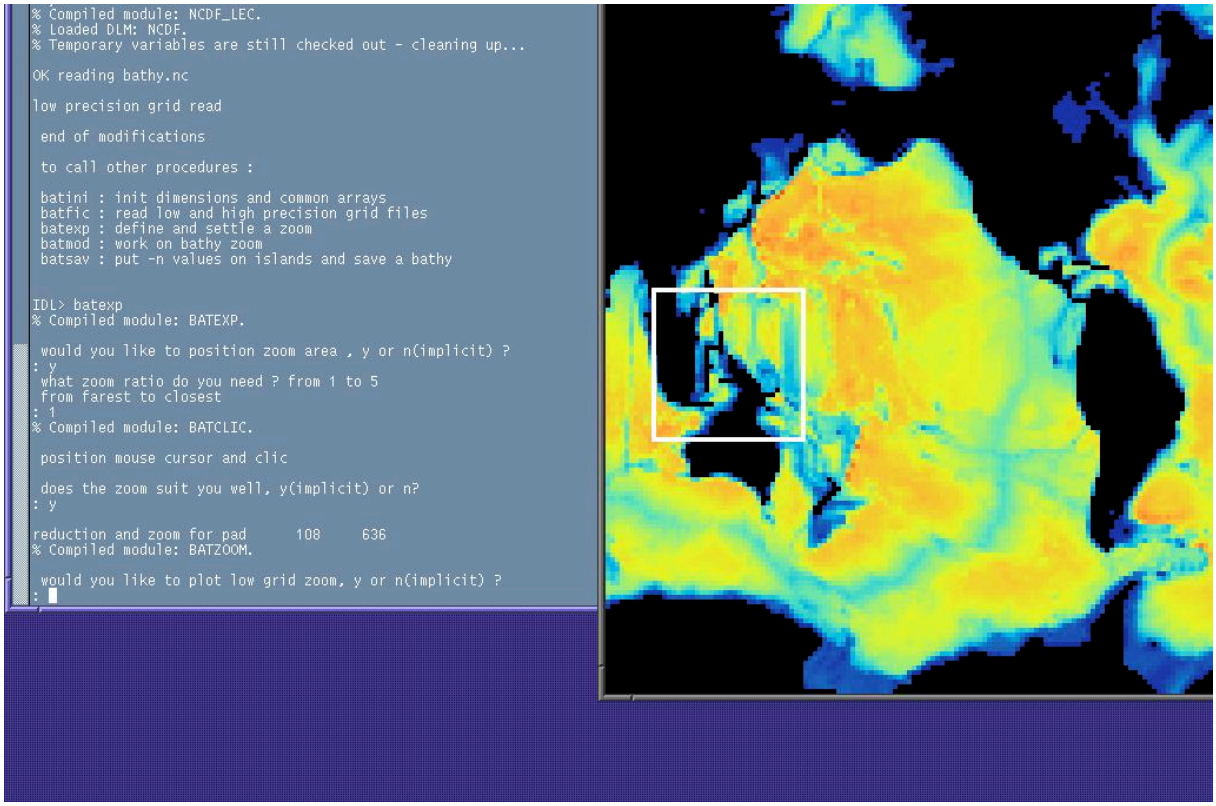
- **Batini** sert uniquement à initialiser les grandeurs propres à chaque bathymétrie et à créer les tableaux qui contiendront les parties des bathymétries que l'on va modifier.



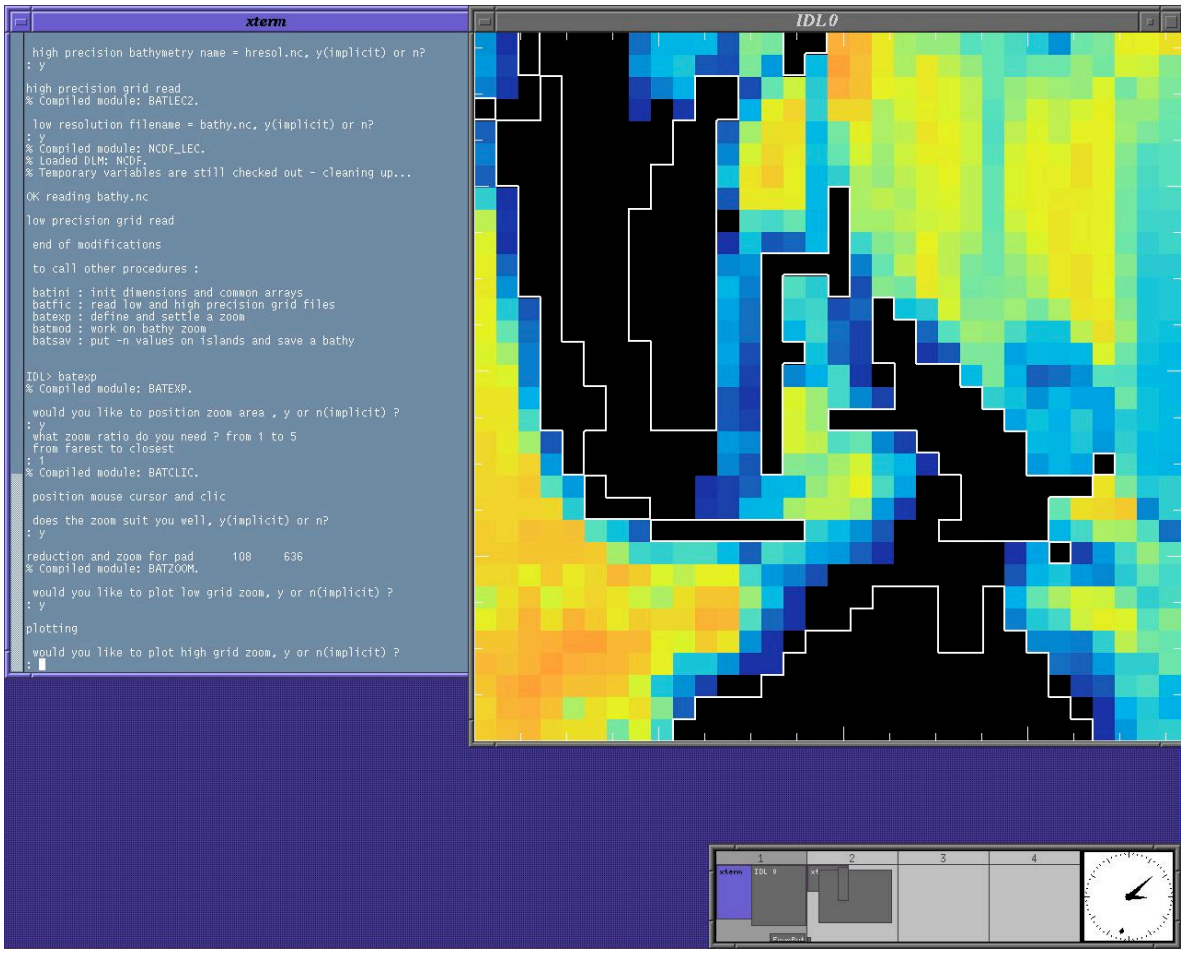
le choix du modèle se fait dans batini

Puis il permet de choisir une des sous-routines. Chaque sous-routine correspond à une fonctionnalité du programme. Le premier **batfic** permet de choisir les fichiers de bathymétries à lire et garde leurs chemins d'accès en mémoire. Pour cela il fait appel à deux sous-routines **batlec** et **batlec2** qui permettent de référencer respectivement la haute et la basse résolution. Dans le cas de la basse résolution, la routine **batlec2** contient en plus un affichage en couleur du fichier choisi.

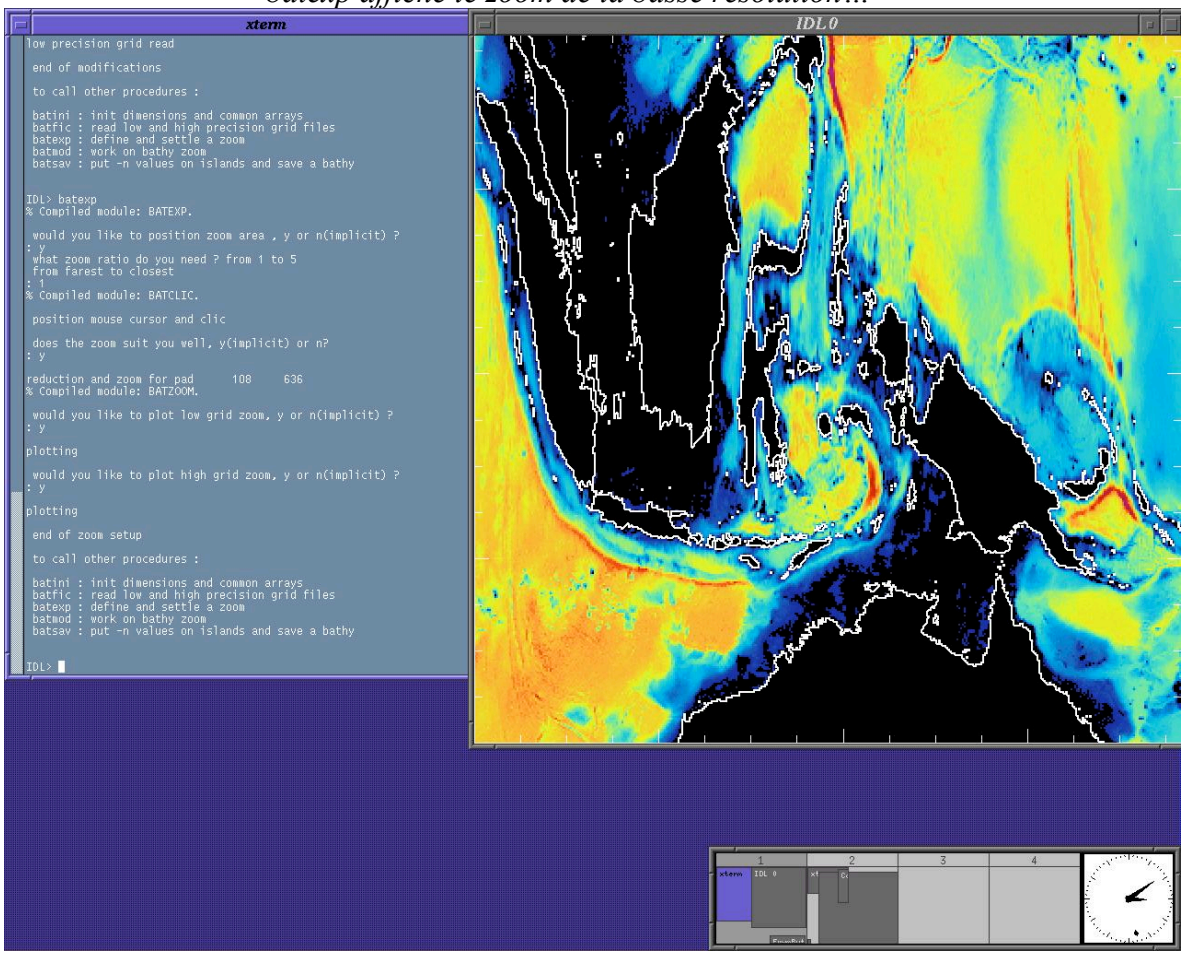
La routine **batexp** permet ensuite de positionner le zoom et de régler sa précision, c'est à dire de sélectionner la partie de la bathymétrie sur laquelle on va travailler, le réglage de la précision étant là pour adapter la taille de la zone à la taille des mailles, en effet il faut que l'affichage de la zone sélectionnée permette de distinguer les mailles individuellement sans effort mais tout en gardant à l'image un nombre suffisant de mailles, cela pour deux raisons. La première est la nécessité de garder un certain recul par rapport à la région autour de la maille que l'on modifie (le dessin de la bathymétrie doit être aborder du point de vue de la circulation océanique) mais aussi pour pouvoir modifier un nombre significatif de mailles sans changer tout de suite de zoom. Cette routine charge alors en mémoire dans quatre tableaux différents les valeurs correspondant à la zone sélectionnée : les bathymétries haute et basse résolutions, la valeur moyenne et le pourcentage d'océan.



positionnement du zoom dans batexp
choix de la taille de ce zoom

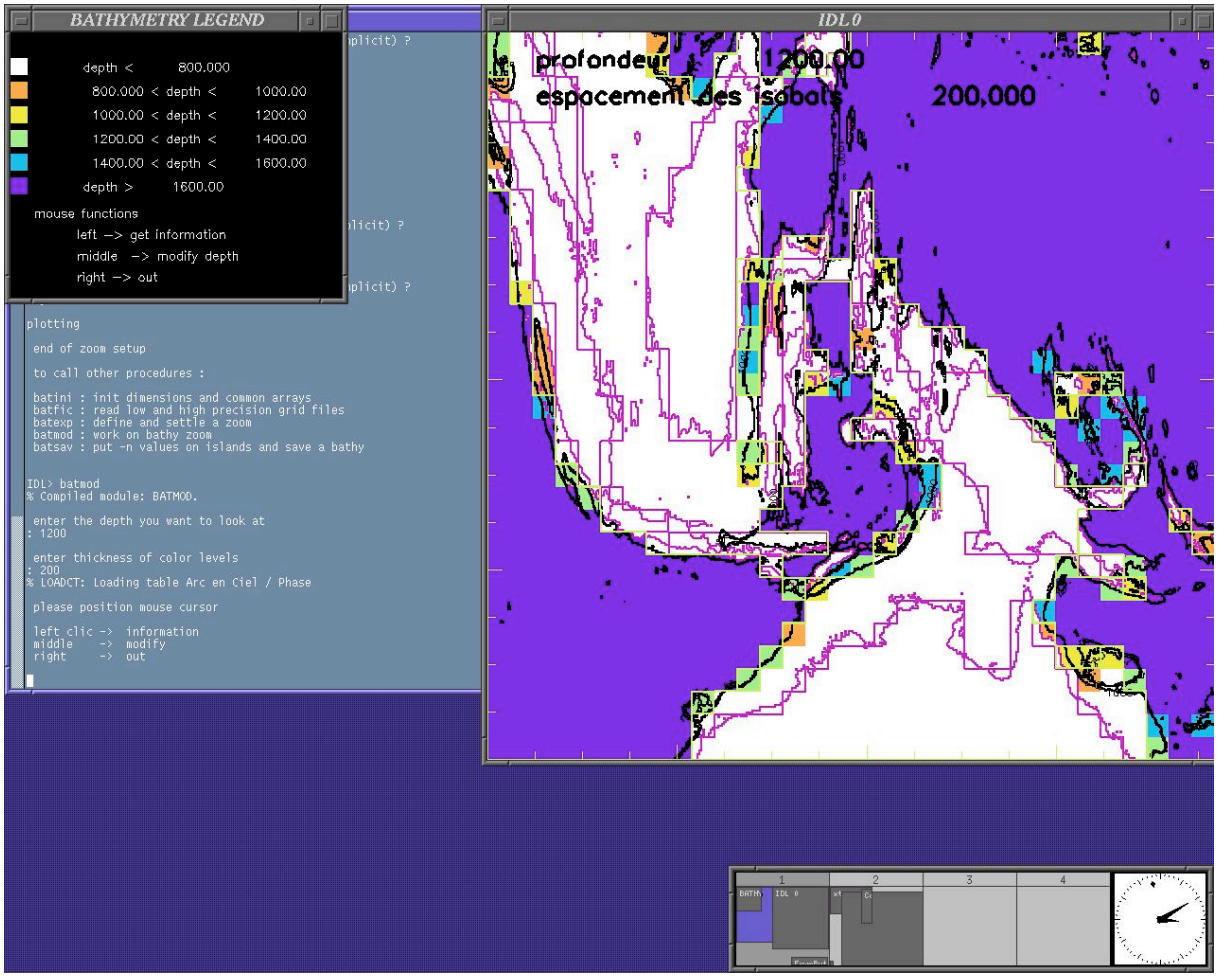


batexp affiche le zoom de la basse résolution…



...puis le zoom de la haute résolution de la zone sélectionnée

La routine **batmod** est la routine qui permet de modifier la valeur de la profondeur des mailles. Elle affiche deux fenêtres : une pour la légende et une pour afficher la bathymétrie grâce à une palette de couleur. Le programme demande à l'utilisateur de choisir une profondeur et une épaisseur en mètres, la profondeur représente la valeur centrale du niveau que l'on va regarder et l'épaisseur l'intervalle de chaque code couleur (il y en a six): par exemple si on choisit 1000 et 300, les six couleurs correspondront aux six intervalles de profondeurs suivants, prof < 400, 400 < prof < 700, 700 < prof < 1000…, prof > 1600. De plus elle trace les isobathes 1000 et 700 en gras et les 400 et 1300 en clair à partir des chiffres de la haute résolution. En cliquant avec la souris on peut obtenir des informations (valeur de la bathymétrie pour la maille, valeur moyenne, valeurs extrêmes de profondeur de la haute résolution pour la zone correspondant à la maille, pourcentage d'océan) ou modifier la valeur de la maille en rentrant un chiffre au clavier. Enfin la procédure **bats** permet de sauvegarder les modifications effectuées dans le fichier netcdf.



une vue de batmod qui permet de repérer aisément les profondeurs voisines de celle demandée grâce à la légende interactive en haut à gauche

2.5 Architecture du programme

PRINCIPAL

- **BATINI**
  - BATFIC
    - BATLEC
    - BATLEC2
  - BATEXP
    - BATCLIC
    - BATZOOM
  - BATMOD
  - BATSAV
    - SCANISL
      - LOOKSOUTH
      - LOOKNORTH
    - LOOKSTART
- **MESHMAIN**
  - MESH
  - MESHDEMI
    - MESH2
- **PRGINTERP3**
  - INCLEC
  - PRGINTERP2
    - PRGINTERP1 (fonction)
- **BATHYBUILD**
  - BATHYMEAN (fonction)

3. Descriptif du code

3.1 La construction du masque haute résolution

Architecture et fonctionnalités respectives

MESHMAIN
    MESH
    MESHDEMI
        MESH2

Meshmain appelle **mesh** pour fabriquer le **meshmask** de la bathymétrie deux degrés et **meshdemi** pour celui de la demi-degré. Celui-ci appelle la sous-routine **mesh2** pour effectuer l'opération sur des pavés et une variable après l'autre.

Problèmes rencontrés et choix des solutions

L'existence de deux sous-routines différentes est nécessitée par la différence dans la taille des fichiers ncdf utilisés qui engendre un manque de mémoire pour le calcul du meshmask de la grille au demi degré. Le calcul pour le deux degré peut se faire d'un trait. Seul l'autre nécessite de parcelliser le travail sur des pavés à l'intérieur de boucles.

Le principe pour affiner un meshmask est de l'interpoler linéairement selon les longitudes et latitudes. L'interpolation sous IDL s'effectue de cette manière :

On a besoin du tableau de départ à interpoler :

$$A = \begin{bmatrix} 10 & 20 & 30 & 40 \\ 50 & 60 & 70 & 80 \\ 90 & 100 & 110 & 120 \end{bmatrix}$$

Suivant l'endroit où l'on veut obtenir les valeurs interpolées, on constitue une matrice contenant les « coordonnées » des points que l'on veut. Par exemple nous allons affiner le tableau ci-dessus de la même manière que les meshmaks, avec un coefficient de précision supplémentaire de 2. La matrice des coordonnées est donc :

suivant l'axe des x $B = [\ 0\ 0.5\ 1\ 1.5\ 2\ 2.5\ 3\ ]$
idem suivant l'axe des y $C = [\ 0\ 0.5\ 1\ 1.5\ 2\ ]$

Voici alors le code IDL de l'interpolation de A suivant B et C :

```
Idl> Result = INTERPOLATE(A, B, C, /GRID)
Idl> Print, Result
```

$$Result = \begin{bmatrix} 10 & 15 & 20 & 25 & 30 & 35 & 40 \\ 30 & 35 & 40 & 45 & 50 & 55 & 60 \\ 50 & 55 & 60 & 65 & 70 & 75 & 80 \\ 70 & 75 & 80 & 85 & 90 & 95 & 100 \\ 90 & 95 & 100 & 105 & 110 & 115 & 120 \end{bmatrix}$$

Un problème se pose cependant lors de l'interpolation des longitudes au niveau du changement de date puisque les longitudes passent de –180 à +180, et donc à ce niveau

l'interpolation ne vaut rien. Il faut repérer à quelle colonne se situe la ligne de changement de date et ajouter 360 à toutes les longitudes. Ainsi il y a continuité et on peut interpoler. Enfin on retire 180 degré à toutes les longitudes supérieures à 360.

```
glamt = NCDF_VARID(id, 'glamt')  ;; on charge en mémoire les longitudes
NCDF_VARGET, id, glamt, vglamt
;; on fait alors une boucle pour rechercher sur chaque ligne à quel endroit
;; se trouve la colonne de changement de date
FOR j = 0, jpj2-1 DO BEGIN
FOR i = 0, jpi2-2 DO BEGIN
e = (vglamt(i, j) – vglamt(i+1, j))
IF (e GT 30 end incr eq 0) THEN BEGIN
num = i+1
incr = 1
ENDIF
ENDFOR
;; on rajoute 360 aux longitudes à partir de cet endroit
vglamt(num : jpi2-1, j)=vglamt(num: jpi2-1, j) + 360
ENDFOR
```

Pour calculer le meshmask du demi degré, un problème supplémentaire se pose, à savoir le manque de mémoire (les tableaux à gérer sont trop gros). **Meshdemi** construit le fichier ncdf de sortie vide et lance **mesh2** qui va s'occuper uniquement du calcul et redonnera la main à **meshdemi** pour qu'il enregistre le résultat. **Mesh2** travaille sur des pavés prédéfinis, ainsi, grâce à la fonctionnalité des fichiers ncdf, seuls ces pavés sont ouverts et gardés en mémoire. Pour le demi degré, la grille du meshmask d'origine fait une taille de 722 par 511 points. On découpe cette grille en des pavés de 200 par 200 points (dont la mise en mémoire est supportée par IDL) puis des pavés de tailles variables pour contenir tout le reste du fichier. On appelle l et k les coordonnées du premier point du pavé à aller chercher dans le fichier total et klen et llen les tailles du pavé. Ainsi on utilise la fonction suivante de IDL pour mettre les valeurs du pavé considéré de glamt dans un tableau nommé vglamt :

```
NCDF_VARGET, id, glamt, vglamt, count=[llen,klen,1,1], offset=[l,k,0,0]
```

Puis on interpole sur ces valeurs le meshmask plus fin.
Mais cette opération ne peut être réitérée plus de 2 ou 3 fois. C'est pourquoi au lancement de la boucle sur les pavés, il faut donner les indices du pavé de départ et de celui d'arrivée :

```
FOR ii = iidebut, iifin DO BEGIN
FOR jj = jjdebut, jjfin DO BEGIN
;; procédure d'interpolation sur ces pavés...cf ci-dessus
```

Ces coordonnées iidebut, iifin et jjdebut, jjfin sont demandées lors du lancement de meshdemi dans le cas où l'utilisateur veut continuer un calcul de meshmask débuté.

Adaptations possibles

Pour les bathymétries suivantes il suffira de changer la taille des pavés voire de l'automatiser par la même méthode que celle écrite dans bathybuild. Si la mémoire le permet, on peut évidemment envisager d'effectuer les calculs en une seule fois.

3.2 La construction de la bathymétrie haute résolution

Architecture et fonctionnalités respectives

PRGINTERP3
    PRGINTERP2
        PRGINTERP1 (fonction)
INCLEC

Pour construire la bathymétrie haute résolution, il faut être en possession du meshmask fin (cf. ci-dessus). **Prginterp3** appelle successivement **prginterp2** puis **prginterp1** pour reporter les calculs sur des petits pavés (dans le même ordre d'idée que pour **mesh2**). Cet ensemble de procédures est long et ne peut fonctionner en une fois par manque de temps CPU. C'est pourquoi **prginterp3** enregistre un incrément qui correspond à la dernière ligne calculée. Cet incrément peut être lu par **inclec** pour reprendre un calcul inachevé.

Problèmes rencontrés et choix des solutions

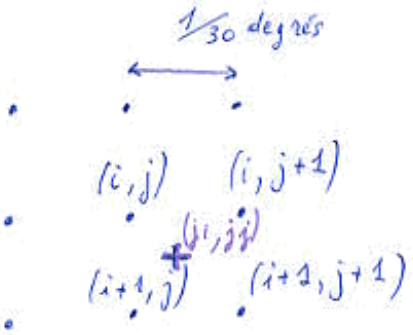
Il faut arriver à garder le moins de variables en mémoire, et donc ouvrir partiellement les fichiers de données. L'idée est de reconnaître l'emplacement d'un point de la grille ORCA sur le fichier de données (car les mailles ne sont pas régulières et ne correspondent donc pas) grâce aux longitudes et latitudes. **Prginterp3** ouvre donc les variables de longitude et de latitude du meshmask fin, puis crée un fichier qui contiendra la future bathymétrie haute résolution. **Prginterp3** calcule ensuite la taille des pavés sur lesquels **prginterp2** et **1** vont travailler (taille allant jusqu'à 30 points par 30). Une double boucle imbriquée sur les lignes puis les colonnes lance **prginterp2** et augmente l'incrément. **Prginterp2** définit quant à lui les pavés et lance **prginterp1** dans une boucle sur chacun des points du pavé. Après chaque pavé interpolé par **prginterp1**, **prginterp2** enregistre les résultats au bon endroit, k et l sont les coordonnées du premier point du pavé considéré (soient les coordonnées de la boucle de **prginterp3** multipliées par la taille du pavé) et u et v représentent la taille du pavé:

NCDF_VARPUT, id, 'hdepth', bthr, count = [u, v], offset = [k, l]

Cette fonction qui permet un gain de mémoire obligatoire est une nouvelle fois possible grâce à l'utilisation de fichiers ncdf. Voici le processus d'interpolation de **prginterp1** : pour trouver les coordonnées des points avoisinant le point du meshmask que l'on considère on multiplie par 30 les longitudes et latitudes ce qui nous donne les indices de ce point dans le fichier de données (car celui-ci est au trentième de degré)

```
lat = glat(jj,ii)
lon = glon(jj,ji)
j=fix(rapo*lon)          ;; rapo = 30
i=fix(rapo*(lat+90))  ;; on ajoute 90 aux latitudes qui vont de –78 à +90
```

Nous obtenons donc i et j les coordonnées du premier point proche du point de la grille du meshmask fin. Les autre points sont donnés par leurs indices : (i, j+1) ; (i+1, j) et (i+1, j+1).



Ces quatre points sont les données les plus proches du point en lequel on veut la valeur interpolée de la profondeur. On lit alors les 4 profondeurs en ces points et l'on interpole le tout :

```
prof = NCDF_LEC('fichier_de_données_au_trentième_de_degré.nc',
                                    var='TOPO ', count = [ 2, 2 ] ,offset = [i, j] )
result = fix( INTERPOLATE(prof, [0.5] , [0.5] ,  /GRID ) )
```

Lorsque l'on arrive au bout du fichier, il faut aller chercher au début les valeurs manquantes (par périodicité) : il suffit de faire une boucle conditionnelle et si i+1 dépasse la taille du fichier, alors on prend les valeurs de la première colonne.
La valeur renvoyée par **prginterp1** est enregistrée dans **prginterp2**.

3.3 La construction du fichier basse résolution

Architecture et fonctionnalités respectives

BATHYBUILD
    BATHYMEAN (fonction)

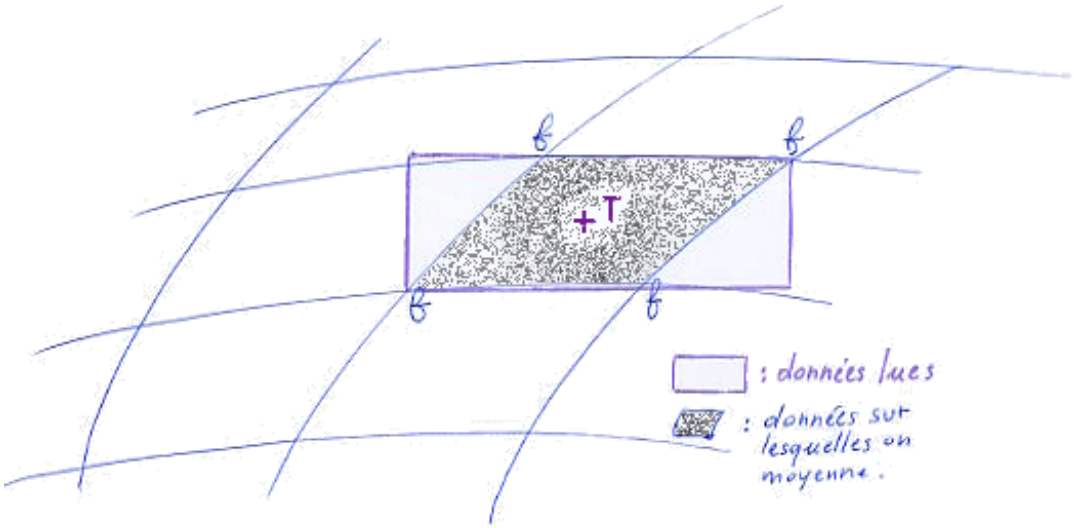
Bathybuild fabrique la bathymétrie basse résolution en appelant **bathymean**. Ces procédures se servent du fichier de données et du meshmask original.

Problèmes rencontrés et choix des solutions

Ces deux procédures ont été écrites par J Emile-Geay, stagiaire DEA au LODYC. Nous les avons réécrites afin d'en faire un programme exécutable, en automatisant les choix pour les différentes précisions de grille. Nous avons également écrit les lignes permettant le calcul des pourcentages terre-océan et de la variance, et c'est dans **bathybuild** que ces variables sont enregistrées dans le fichier de sortie.
Comme dans la plupart des procédures, un choix interactif avec l'utilisateur lui permet de choisir la précision de la grille, le nom du meshmask à utiliser et le nom de la bathymétrie à construire. Le principal problème revient comme toujours à réduire le temps de calcul sans pour autant dépasser la mémoire qui nous est allouée. Le principe réside donc une fois de plus à effectuer une boucle sur les points de la grille, de lancer à l'intérieur de cette boucle **bathymean** et d'enregistrer les résultats ligne par ligne dans le fichier à la fin de chaque boucle pour libérer de la mémoire.

```
FOR jj = 1, jpj – 1 DO BEGIN
print, 'jj = ', jj
        FOR ji = 1, jpi – 2 DO BEGIN
        bathymean, ji, jj, zdep, zvar, wper  ;; les arguments avec lesquels est exécuté
                ;;  bathymean sont les coordonnées du point dont on veut les valeurs,
                ;;  profondeur moyenne, la variance puis le pourcentage
                ;;  En les mettant ainsi en argument, on les nomme afin de pouvoir
                ;;  utiliser comme suit, sans pour autant les garder toujours en
        mémoire :
        zdept(ji – 1) = zdep
        depv(ji – 1) = zvar
        wperc(ji – 1) = wper
        ENDFOR
NCDF_VARPUT, id, zvarid, zdept, count = [jpi-2, 1, 1], offset = [1, jj, 0]
                ;;  on enregistre à l'aide de varput la ligne de résultats zdept
                ;;  dans une variable zvarid du fichier de sortie
        ...     ;;  on fait la même chose avec les trois autres variables
ENDFOR
```

Bathymean commence par définir une 'boite' qui contient la maille ORCA concernée sur laquelle sera faite la moyenne des profondeurs. Seules les données intérieures à cette boite sont lues pour gagner de la mémoire, toujours grâce au format netcdf.



la 'boite' entourant la maille

Un autre problème apparaît lorsque l'on arrive à l'extrémité du fichier : il faut une nouvelle fois utiliser la périodicité et 'recoller' les données pour lire les profondeurs sur toute la zone nécessaire. Les mailles ORCA n'étant pas régulières, il faut calculer les équations des droites qui forment les côtés de la maille contenue dans la 'boite'. Celles-ci sont calculées grâce aux coordonnées des quatre extrémités (points nommés f) de la maille concernant le point ji,jj en lequel on veut la profondeur (point nommé T). **Bathymean** calcule enfin les valeurs désirées sur ce domaine (la maille T). On applique de cette manière au point T la moyenne des profondeurs de toutes les données intérieures à la maille concernée.
Le fichier de sortie contient donc quatre variables : la moyenne des profondeurs des données, la profondeur de la basse résolution (qui avant retouche est la même que la profondeur moyenne), la variance des profondeurs et le pourcentage terre-océan.

3.4 La retouche de la bathymétrie basse résolution

Architecture et fonctionnalités respectives

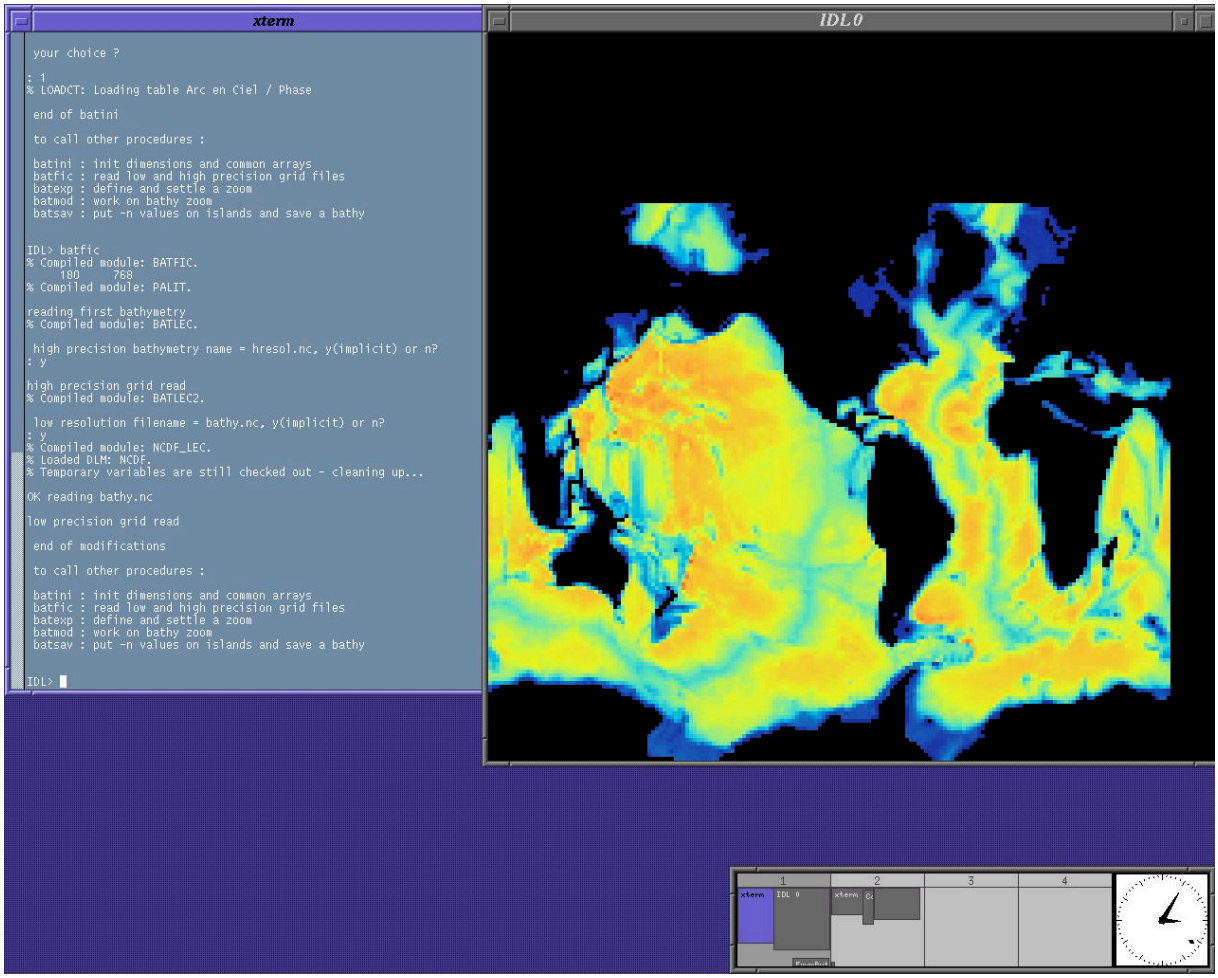
BATINI
- BATFIC
  - BATLEC
  - BATLEC2
- BATEXP
  - BATCLIC
  - BATZOOM
- BATMOD
- BATSAV
  - SCANISL
    - LOOKSOUTH
    - LOOKNORTH
  - LOOKSTART

Batini initialise les variables et les tailles des tableaux suivant le modèle que choisit l'utilisateur. Ensuite **batfic** lit les bathymétries demandées en appelant **batlec** (lecture de la haute résolution) et **batlec2** (lecture et affichage de la basse résolution). **Batexp** permet de choisir une zone à agrandir (**batclic**) avec différentes tailles de zoom, et affiche ce zoom (**batzoom**). **Batmod** sert à modifier point par point les profondeurs du zoom grâce à un code de couleur, et enfin **bats** arrange la bathymétrie pour qu'elle soit exploitable par OPA (mise des îles à –n par **scanisl**, périodicité et repliement) et enregistre les modifications faites dans un nouveau fichier netcdf.

Problèmes rencontrés et choix des solutions

La lecture de fichiers, surtout lorsqu'ils sont volumineux comme ceux que nous utilisons, prend un temps non négligeable et beaucoup de place en mémoire. L'ancienne version du logiciel regroupait en une seule procédure la lecture des fichiers et l'affichage du zoom. Nous avons séparé les deux étapes pour pouvoir zoomer à tout moment sans avoir à relire les fichiers (**batfic** puis **batexp**). Lire en entier les deux résolutions des bathymétries reste une opération trop lourde. Donc **batfic**, qui appelle successivement **batlec** et **batlec2**, ne retient que le nom des fichiers à utiliser (*inohr* pour la haute résolution et *inobr* pour la basse). **Batlec2** affiche également une version 'dégradée' de la bathymétrie haute résolution, c'est à dire qu'il ne lit pas toutes les lignes ni toutes les colonnes. La résolution est donc divisée par *xresol* horizontalement et par *yresol* verticalement. La différence n'est pas visible, mais le gain en mémoire est important. C'est la fonction *stride* qui permet de lire un point tous les *xresol* et *yresol* points dans un fichier ncdf :

```
;; définition de la résolution de lecture
xresol = jpi2 / 182
yresol = jpj2 / 149
zbat = NCDF_LEC(iodir+iname, var = 'depth', STRIDE = [xresol, yresol, 1])
```



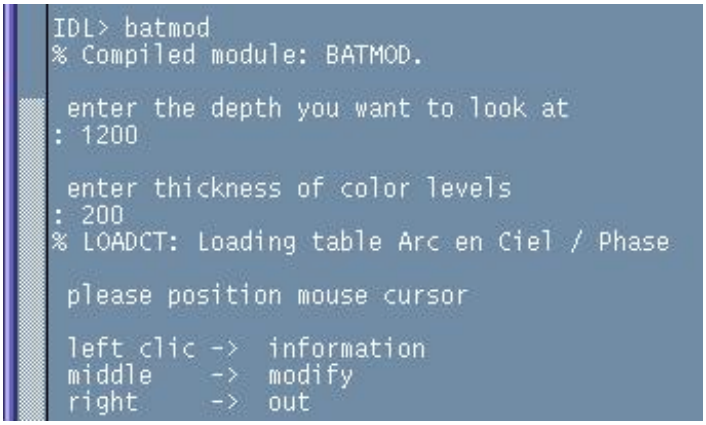
affichage d'une version 'dégradée' de la basse résolution

Opabat fonctionne avec toutes les précisions de bathymétrie (2 degrés, demi-degré…) mais le zoom initialement n'était pas réglable. Or pour retoucher les bathymétries plus fines que celle à 2 degrés le zoom par défaut ne suffit pas. L'utilisateur choisit entre 5 zooms différents (*izoo*) et le rapport de zoom devient : *alph = 2^izoo*xresol.*
C'est uniquement au moment de l'affichage du zoom que la lecture des fichiers s'effectue. Ainsi ne sont lues que les profondeurs de cette zone. Il faut cependant faire une petite gymnastique pour retrouver les coordonnées véritables du zoom, ainsi que la taille du sous tableau lu dans le fichier : en effet la position reconnue par le curseur de la souris ne correspond pas à cause de la précision de l'affichage (*xresol* et *yresol*) :

```
izz = jpiwdo / alph
jzz = jpjwdo / alph
nizoom2 = fix(xresol#nizoom#(jpi2-l)/(jpi-1))      ;; coordonnées du zoom ...
njzoom2 = fix(yresol#njzoom#(jpj2-l)/(jpj-l))      ;; ... pour la basse résolution
nizoom3 = nizoom2#jpres-jpres/2                    ;; ... pour la haute résolution
njzoom3 = njzoom2#jpres-jpres/2
izz2 = float(round(xresol#izz#(jpi2-l)/(jpi-l)))   ;; taille du zoom ...
jzz2 = float(round(yresol#jzz#(jpj2-l)/(jpj-1)))   ;; ... pour la basse resolution
izz3 = izz2#jpres                                  ;; ... pour la haute résolution
jzz3 = jzz2#jpres
;;
```

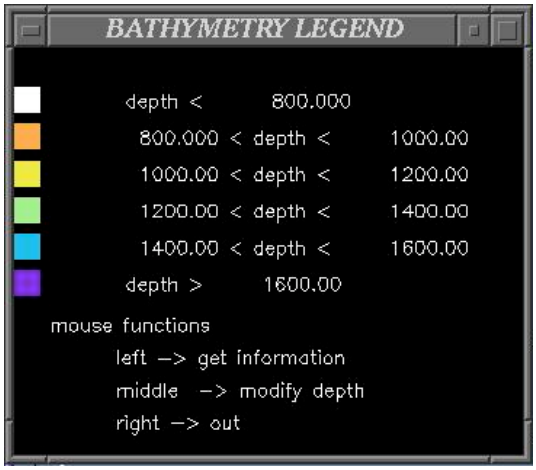
```
;; reading only the sub-arrays to be plot
br2=ncdf_lec(iodir+'temp.nc', var='depth', count=[izz2,jzz2,1],
                                    offset=[nizoom2,njzoom2,0])
hr2=ncdf_lec(inohr, var='hdepth', count=[izz3,jzz3], offset=[nizoom3,njzoom3])
brmoy=ncdf_lec(iodir+'temp.nc', var='depmean', count=[izz2,jzz2,1],
                                    offset=[nizoom2,njzoom2,0])
brwper=ncdf_lec(iodir+'temp.nc', var = 'watperc', count=[izz2,jzz2,1],
                                    offset=[nizoom2,njzoom2,0])
```

On ne peut pas travailler sur un éventail de plus de 5000 mètres de profondeur. Batmod permet de travailler sur une profondeur donnée à plus ou moins quelques dizaines ou centaines de mètres près. C'est un pas de profondeur (correspondant aux couleurs) qui nous renseigne à ce sujet.



au lancement de batmod, on demande la profondeur et le pas des mailles sur lesquelles on veut travailler

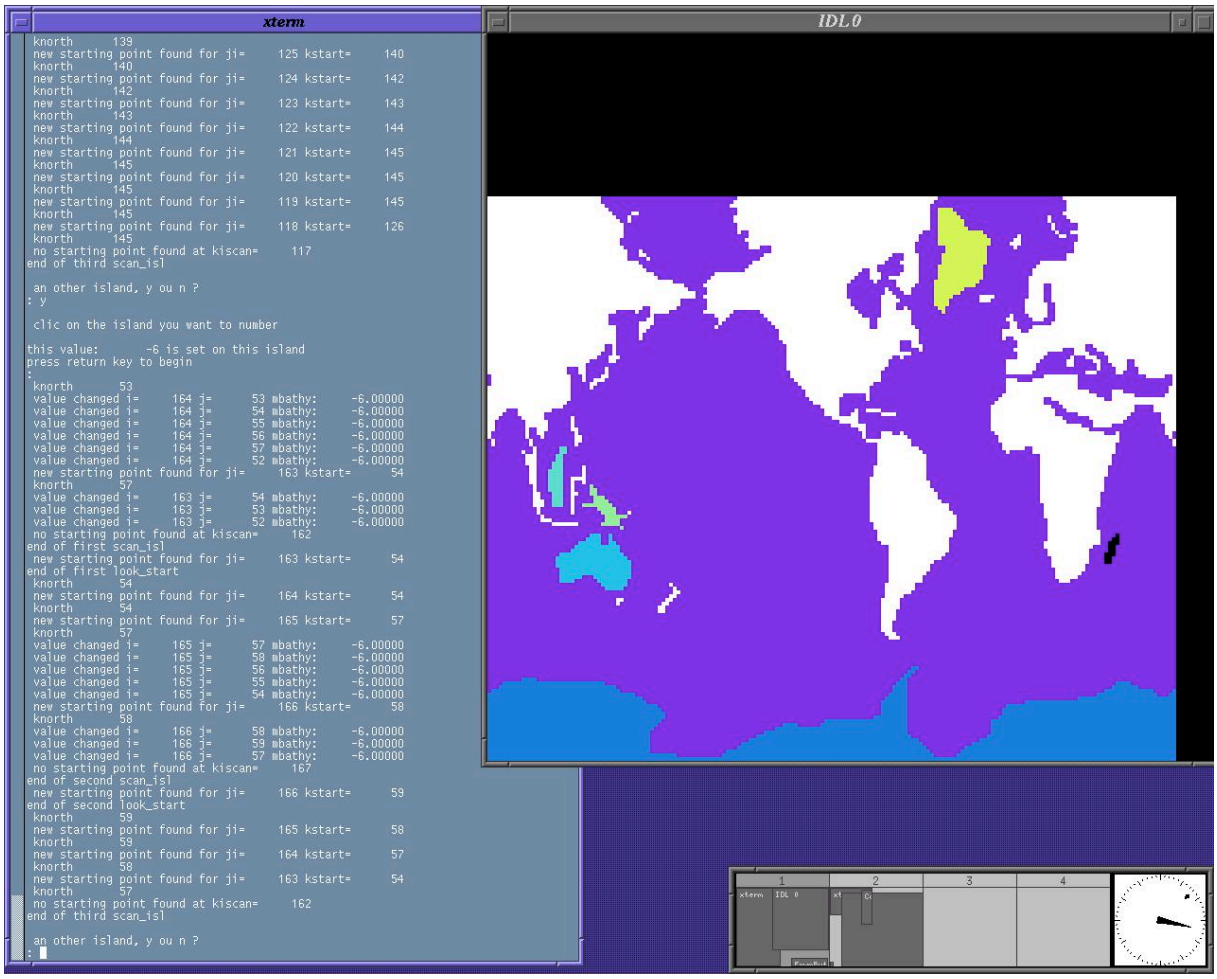
Pour savoir à tout moment à quelle profondeur nous travaillons dans **batmod**, nous avons automatisé l'affichage de la légende qui affiche les codes de couleurs correspondant aux profondeurs souhaitées :



L'espacement des niveaux de couleurs est maintenant réglable (dans l'ancienne version les couleurs correspondaient aux niveaux de profondeur, maintenant changer de couleur tous les mètres n'a pas de sens) et cet espacement correspond aussi aux tracés des isobathes qui sont essentielles lors de la retouche de la bathymétrie.

La structure de **batmod** pour ce qui est du changement de profondeur de certains pavé est basée sur de nombreuses boucles imbriquées les unes dans les autres (boucles *while, if…*) pour effectuer l'opération que l'on veut (*infos, modifs, quitter…*) au moment où on veut. Comme on peut travailler successivement sur plusieurs zooms, il ne faut pas oublier d'enregistrer les modifications. Avant les modifications du zoom étaient reportées directement dans le tableau de bathymétrie qui restait tout le temps en mémoire. Maintenant que ce tableau n'est plus lu entièrement, il faut reporter les modifications dans un fichier ncdf temporaire (*temp.nc*) qui est créé au départ par simple copie du fichier ncdf de la bathymétrie basse résolution choisie. Lors de **bats** et de l'enregistrement de la bathymétrie, il suffit alors de copie *temp.nc* vers le fichier ncdf choisit et de supprimer *temp.nc*.

Pour que la bathymétrie soit valable, il faut mettre des valeurs négatives sur les continents et îles. C'est le rôle de **batisl**, mais l'algorithme pour couvrir une île en entier est très complexe (il y a des points isolés). Celui qui est en fait utilisé est rapide mais ne couvre pas toutes les îles entièrement (il reste des zones à 0). Il a donc fallu écrire un complément qui repère les zones à 0 des îles numérotées (*rattrapage des points isolés*).



les îles déjà numérotées apparaissent en couleur ;
il faut faire un rattrapage lorsqu'une île n'est pas entièrement colorée

4. Conclusion

Les limites du programme développé et les suggestions

Le programme développé est complètement fonctionnel pour les bathymétries deux degrés et demi-degré et pour le quart de degré seule la fabrication du mehmask haute résolution nécessite une légère intervention dans le code : il faut calculer combien de pavés de 200 sur 200 le tableau contient, la taille des pavés restants et attribuer aux variables correspondantes ces valeurs. On peut aussi, de la même manière que dans prginterp3, mettre en place une affectation automatique de ces variables par choix de l'utilisateur très facilement. Cette seconde solution permet de ne pas faire à chaque fois une intervention dans le code et évite un dimensionnement interne à celui-ci. Pour des résolutions plus fines, il suffit de rajouter dans les codes d'initialisations les valeurs caractéristiques de ces résolutions (c'est déjà fait pour la quart de degré), avec - dans le cas de la production du meshmask haute résolution - une retouche un peu plus longue mais selon la même démarche que celle explicitée pour la quart de degré.
Cependant il faudra veiller à ne pas augmenter la précision des hautes résolutions au delà du trentième de degré, qui est la précision des données. Pour ce qui est du changement de fichier de données, il faudra alors retrouver et changer tous les adressages relatifs à la lecture des données principalement dans les fichiers d'initialisations. Si le nouveau fichier de données est d'une résolution supérieure à celui utilisé actuellement il faudra modifier prginterp1 pour ce qui est du repérage des quatre points les plus proches. Pour les autres il n'y a pas de modification nécessaire.
On peut remarquer que ces programmes utilisent un certain nombre des routines de Sébastien Masson, donc toute modification de la place de ces routines rendrait le programme inopérant. De même le programme est conçu pour utiliser des meshmasks construits comme des fichiers netcdf à quatre dimensions (trois d'espaces puis une de temps) et dont les variables sont toujours appelées de la même façon (ex : glamt, gphif …) et serait incapable de fonctionner avec des fichiers construits différemment sans une retouche du nom des variables, ou - si le nombre de dimensions est différent – une modification de tous les count et les offset des programmes qui lisent ces meshmasks. Il faudrait prendre les mêmes précautions pour retoucher une bathymétrie qui n'aurait pas été créée par ce programme.
Enfin ce programme a été automatisé le plus possible pour la plupart des réglages nécessaires à son fonctionnement mais il ne possède aucune sécurité quant à des erreurs qui viendraient de l'utilisateur. Par exemple si vous lui fournissez une bathymétrie deux degrés et que vous choisissez demi-degré dans l'initialisation, il n'y aura pas de message d'erreur immédiat et peut-être même jamais de blocage fonctionnel. La mise en place de tels contrôles est tout à fait possible mais elle alourdirait significativement le code et de plus, le choix des fichiers se faisant après les initialisations, elle nécessiterait la possibilité de modifier les initialisations à chaque fois qu'une erreur est rencontrée.
Enfin les questions et les choix sont en anglais puisque ce logiciel est destiné à être utilisé par la communauté scientifique mondiale (certains des stagiaires du LODYC ne parlent d'ailleurs pas le français), mais il serait très facile d'en faire une version

bilingue voire multilingue . Il suffit de copier les routines une par une en rajoutant au nom de la copie une lettre distinctive (par exemple, principal.pro deviendrait pour la version française fprincipal.pro) puis de traduire tous les textes qui sont affichés et les appels de sous routines (par exemple fbatfic appelle fbatlec et fbatlec2) ; il faut ensuite écrire une routine demandant de faire un choix et qui, en fonction de ce choix, appelle principal, fprincipal ou dprincipal ….
Cependant le principal défaut de fonctionnement de ce programme est le fait que certaines de ces taches sont interrompues avant leur fin par le dépassement du temps CPU sous Rhodes. Mais il n'y a que deux solutions à ce problème, la première étant de supprimer les limitations de temps CPU de Rhodes ou alors l'utilisation d'un calculateur plus puissant.

Bilan personnel de ce stage scientifique

L'apport principal de ce stage a été la découverte du mode de fonctionnement d'un laboratoire, du travail de chercheur et des réalités matérielles de ce travail. Le projet global est un travail typique de recherche, on cherche à établir un modèle fonctionnel de l'océan, et surtout à rendre ce modèle le plus réaliste possible. Le modèle OPA utilise les équations primitives et les différences finies, qui sont des outils qui nous ont été présentés en cours, et voir leur mise en œuvre concrète dans un projet aussi vaste que celui-ci a été très rassérénant. Et nous avons pu constater la réalité du choix des constantes après des tâtonnements successifs.

5. Bibliographie

OPA version 8.1 Ocean General Circulation Model reference manual. LODYC, France, Internal Report (1993) ; Delecluse P, Madec G, Imbard M, Lévy C.
Cet ouvrage très technique est la documentation du code OPA.

Manuel de l'utilitaire OPABAT. Disponible en ligne www.lodyc.jussieu.fr/opa/

Ocean Circulation, prepared by an open university course team, Library of Congress, British Library, Pergamon Press, England (1989, 1991).
Les chapitres 1 et 2 nous ont permis d'obtenir quelques bases en océanographie.

Le maillage ORCA : un maillage quasi isotrope de l'Océan Mondial. Rapport de stage de Christophe Bagot (ENPC 1998-1999).