

Parallélisation de l'étape de préparation des champs physiographiques de SURFEX (PGD).

Concepts généraux de PGD et parallélisation pour une grille en projection conforme

Rédacteur : V. Masson

1. Principes généraux de PGD

PGD (pour PhysioGraphic Data) sert à initialiser tous les champs décrivant la surface et qui n'évoluent pas (ou lentement pour ce qui concerne la végétation). Ces champs comprennent typiquement :

- le type d'occupation du sol, dont peuvent être déduites les fractions de chaque type de « tile » (mer, eaux intérieures, ville, surfaces naturelles ou agricoles). Ces données peuvent provenir d'ecoclimap (1 ou 2) ou être fournies à partir de données propres à l'utilisateur s'il possède des données plus fines par exemple.
- Le relief et tous les paramètres liés au relief sous-maille (écart-type, paramètres de rugosité, divers reliefs dérivés du relief moyen et de paramètres sous-maille, etc...).
- Les champs d'argile et de sable pour la description du sol d'ISBA. Ceux-ci existent maintenant à une résolution de 30'' sur le globe.
- Des champs liés à des fonctionnalités spécifiques d'ISBA (plaines d'inondation par exemple).
- Les paramètres décrivant les lacs.
- De nombreux paramètres peuvent soit être dérivés de l'occupation du sol, soit fournis par l'utilisateur par ailleurs s'il possède des informations plus précises (ces paramètres peuvent être ceux décrivant la végétation comme l'indice foliaire ou la fraction de végétation, ou ceux décrivant la ville).

PGD est ainsi principalement un opérateur géographique, utilisant des fichiers de données à une certaine résolution pour les agréger sur la grille du modèle. Si à l'issue de cette phase de lecture/agrégation certains points de grille du modèle n'ont pas été remplis, ceux-ci sont interpolés à partir des points de grille qui ont été remplis.

La parallélisation revient principalement à découper la grille du modèle en plusieurs sous-grilles que l'on va ensuite concaténer.

2. Méthode scientifique pour paralléliser les calculs de PGD

2.1 découpage en sous-domaines indépendants avec halos

La parallélisation a été construite de telle façon que les calculs scientifiques sur chaque sous-grille sont indépendants de ceux sur les autres sous-grilles. Ceci permet de ne pas avoir d'échange de champs entre les processeurs. Seuls quelques entiers sont parfois échangés (par exemple pour connaître le nombre total de points de mer, ...). Ceci est possible car :

- Chaque sous-grille va lire les données qui concernent son domaine géographique propre (pour optimiser le temps calcul et surtout d'I/O, chaque sous-domaine n'ira lire qu'une partie des fichiers de données comme *ecoclimap* ou *gotpo30* pour le relief).
- Chaque sous-grille va effectuer chaque point de donnée (en fonction de sa position) au point de grille correspondant. Cette mécanique est identique à celle en monoprocesseur, et est le cœur scientifique de PGD.

Cependant :

- Pour certains champs d'orographie sous-maille lorsque l'on est à haute résolution (ce n'est pas important pour ARPEGE par exemple), chaque sous-grille nécessite de connaître les points immédiatement voisins.
- Pour les interpolations, chaque sous-grille peut avoir besoin d'aller chercher des points voisins, relativement loin.

Ainsi, pour satisfaire ces deux dernières conditions sans avoir à intégrer au sein de PGD des routines de parallélisation pouvant être propres à chaque système, ont été ajoutés un certain nombre de points, dits de « halo », à chaque sous-grille. Les calculs se font ainsi sur l'ensemble de ces points, permettant ainsi des calculs propres et reproductibles (au sens multiprocesseur du terme), sur la partie interne (hors halo) de la sous-grille. Le nombre de points de halo peut être spécifié par l'utilisateur s'il le souhaite.

Ce n'est que lors de l'écriture des champs que chaque système d'I/O et de parallélisation va supprimer les points de halo (ce qui est nouveau), puis regrouper les champs et les écrire (ce qui se faisait déjà pour SURFEX, mais pas pour l'étape PGD). Ainsi, hors l'étape de suppression des halos, l'écriture peut se reposer sur des éléments déjà existants (c'est ce qui a été fait pour les I/O dans le système MesoNH).

2.2 lecture de la sous-partie correspondante des fichiers de données

Chaque grille lisait auparavant, dans les fichiers de données à accès direct (comme *ecoclimap*, *gtopo30* sable et argile) que les zones géographiques qui la concernaient. En pratique, on détectait tous les carrés de $0.5^\circ \times 0.5^\circ$ qui intersectaient la grille du modèle visée. Et on ne lisait, grâce à l'accès direct, que les points dans ces quarts de degrés carré (qu'on savait utiles). Pour une grille en projection conforme comme pour ALADIN, AROME ou MesoNH, ceci permettait de limiter le nombre de points lus dans le fichier (limitant les accès disques) et limitant aussi les calculs liés à ces points. Evidemment, pour une grille de gauss pour ARPEGE, on lisait tous les points, puisque la grille est globale.

A présent, dans la version parallèle, on travaille sur des sous-grilles. Les processus de détermination des zones géographiques à lire reste donc valable. Il faut juste adapter les routines permettant de détecter si un quart de degré carré est ou non concerné par la sous-grille (avec son halo). C'est trivial pour une sous-grille en projection conforme (car on s'arrange pour qu'elle soit elle-même une grille rectangulaire de type projection conforme) ou une grille rectangulaire régulière en latitude-longitude ou grille irrégulière à qui les points sont fournis (mêmes raisons). La seule grille actuelle dont les sous-grilles ne sont pas de type de la grille complète est la grille de gauss. La grille de gauss est globale, alors que les sous-grilles concernent des sous-domaines géographiques.

2.3 enchaînement des tâches, appels à des fonctions d'échanges

La routine principale PGD_SURF_ATM, qui effectuait l'ensemble des calculs scientifiques de PGD, a été coupée en 3. 2 parties relativement petites, et une partie scientifique principale.

- une partie (nouvelle routine PGD_GRID_SURF_ATM) sert uniquement à initialiser la grille globale (soit from scratch à partir d'une namelist, soit comme un sous-domaine d'un autre fichier, ce qui est utilisé pour le gridnesting de MesoNH par exemple). A l'issue de cette partie, on connaît la grille complète du modèle. A la fin de cette partie, on dispose donc des informations nécessaires pour initialiser les sous-domaines ou simplement diverses variables d'I/O vis à vis du système utilisé (appel à la routine PGD_GRID_IO_INIT, qui était déjà appelée lorsqu'on était dans le monde MesoNH pour initialiser la surcouche parallèle et les I/O, ... même si l'on tournait forcément à 1 processeur). Les données sont stockées dans le module MODD_SURF_ATM_GRID_n, variables CGRID, NGRID_PAR, XGRID_PAR).

- La deuxième partie (nouvelle routine SPLIT_GRID) découpe la grille totale du modèle en sous-grilles, en fonction du système utilisé (qui peut avoir ses préférences en terme de formes de sous-domaines, même si cela a peu d'intérêt scientifiquement parlant pour PGD). Chaque système rajoute aussi les points de halo à la sous-grille. Les paramètres descriptifs de la sous-grille remplacent ceux de la grille globale dans MODD_SURF_ATM_GRID_n.

- La troisième partie (toute la suite de l'ancienne routine PGD_SURF_ATM) effectue les calculs scientifiques sur sa sous-grille.

2.4 suppression des interpolations globales

Une contrainte liée à la parallélisation est de ne pas avoir de calcul scientifique lourd qui fait intervenir l'ensemble des points du domaine total du modèle. Dans PGD, les calculs étaient principalement locaux (lecture dans des zones géographiques spécifiques et agrégation sur chaque point de grille indépendamment du voisin). Mais, au cas où certains points ne sont pas remplis à l'issue de l'étape d'agrégation, il faut les interpoler, ce qui signifie aller chercher des points à côté. Deux interpolateurs étaient disponibles :

- une routine utilisant des splines. Elle a l'avantage de fournir des champs lisses. Mais a les inconvénients suivants : d'une part, elle est très chère à calculer (variant en n^3 où n est le nombre de points pour lesquels on dispose d'une

valeur à partir de laquelle on interpole). Par conséquent on ne cherchait à l'utiliser que lorsque l'on dispose de données initiales que l'on sait lâches par rapport à la grille du modèle (par exemple l'argile et le sable qui étaient avant à 10km de résolution quand on voulait faire un domaine AROME à 2.5km). Et d'autre part, les calculs sont globaux, faisant intervenir l'ensemble des points de la grille.

- Une méthode où l'on interpole à partir des 3 points (ou du point) le(s) plus proche(s). Cette méthode recherchait d'abord si l'on disposait d'assez de points proches (dans une zone de 5x5 points de grille), et sinon explorait l'ensemble du domaine (mais cela prenait en général peu de temps car peu de points étaient concernés).

Le passage en parallèle a obligé à faire les choix suivants :

- 1) Abandon des interpolations en splines, du fait de leur caractère global. Cependant, des nouveaux champs d'argiles et de sable à 1km de résolution sont maintenant disponible, ce qui rend moins pertinent la nécessité d'interpoler avec les splines.
- 2) Les routines d'interpolation à partir des points les plus proches ne vont plus chercher sur l'ensemble du domaine, mais seulement sur l'ensemble du halo. Si aucun point de grille valide pour interpolé n'est trouvé, alors une valeur par défaut est affectée si elle existe, sinon le programme s'arrête encourageant l'utilisateur à soit augmenter le halo soit fournir un champ plus précis.

A noter que certaines données peuvent toujours être à faible résolution (par exemple des émissions chimiques à 1 degré de résolution, mais il existe mieux maintenant). Pour de telles données, les interpolations à 3 points telles que réalisées dans PGD fournissent des champs bien moins lisses et réguliers que les interpolations par splines. Il pourrait être intéressant d'améliorer ces routines d'interpolation.

3. Parallélisation pour une grille en projection conforme

Le travail de parallélisation effectué jusqu'à présent concerne uniquement la grille de type projection conforme pour les aspects scientifiques. Elle est réalisée complètement au point de vue informatique dans le cadre MesoNH, et une maquette est disponible dans le cadre offline. A priori, il n'est pas nécessaire de construire un environnement spécifique ARPEGE/ALADIN pour la partie purement PGD, les fichiers pouvant être réalisés en offline au format FA puis utilisés au sein des systèmes ARPEGE/ALADIN. Mais c'est aux groupes concernés de trancher s'ils veulent développer un système PGD parallèle spécifique.

3.1 Sous-grilles en projection conforme

Les grilles en projection conforme sont des rectangles en coordonnées conformes sur une surface de projection (cylindre, cône ou plan) sur lequel est projetée la sphère terrestre (permettant d'associer à chaque point du rectangle ses coordonnées géographiques latitude et longitude). Chaque sous-grille est définie par la même surface de projection. En s'arrangeant pour que chaque sous-grille soit aussi un rectangle (ce qui est naturel), on obtient une sous-grille du même type.

Les points de halo sont ajoutés tout autour du rectangle (par défaut on utilise une bande de halo de 5 points sur chaque bord). Ceci permet donc d'obtenir à nouveau une grille de type projection conforme.

3.2 Maquette pour PGD offline

Une maquette a été réalisée en PGD offline. Cette maquette ne fait pas une parallélisation au sens informatique du terme, mais simule les calculs scientifiques en appelant successivement les sous-grilles (au lieu de les appeler parallèlement et simultanément). Chaque sous-grille (et les champs physiographiques associés) sont affectés à un pointeur de surface différent (via les GOTO_SURFEX), mais cela est juste un artefact de la maquette pour être sûr d'avoir bien séparé toutes les infos de chaque processeur.

La maquette regroupe aussi les champs à la fin (formats ASCII ou LFI) afin de produire un unique fichier, afin de comparer les runs pour différents nombres de sous-grilles simulées. Ceci est réalisé très simplement (et très inefficacement) : chaque sous-grille écrit ses données dans un fichier ascii, et la dernière sous-grille relit, lors de l'écriture d'un champ, le champ tous les fichiers et écrit le champ total au format désiré. Ceci n'a été évidemment développé qu'à des fins de tests préliminaires.

Cette maquette n'interagit pas avec le type de grille utilisé vis à vis des calculs scientifiques (lecture, agrégation interpolation), et donc ce qui a été développé pour la grille conforme le sera encore pour d'autres grille. Par contre, elle interagit vis à vis du découpage en sous-grille, des halos à rajouter en fonction de ces découpages, et de la récupération des champs internes pour l'écriture.

4. Ce qu'il reste à faire pour les autres grilles

4.1 Paramètres définissant une sous-grille

Une première tâche concerne à pouvoir remplir dans les pointeurs de type XGRID_PAR toutes les informations relatives à une sous-grille. Ceci revient à modifier les structures de données qui y sont stockées. Comme mentionné plus haut, ceci n'est nécessaire que pour un seul type de grille : la grille de gauss.

Il faudra par exemple ajouter une information dans XGRID_PAR pour conserver quelles sont les bandes de latitude appartenant à la sous-grille (si on choisit de découper la grille de gauss en bandes de latitude).

A noter que l'on peut aussi au passage simplifier les informations contenues dans XGRID_PAR. Par exemple, dans le cas des grilles conformes, les valeurs X et Y de chaque point étaient stockées, alors que, si on considère l'ensemble des types de surface, la grille est régulière. Une modification a donc été réalisée afin de ne stocker que 2 vecteurs listant les valeurs possibles de X et de Y. Si on se situe sur un tile (exemple la ville), les points ne sont plus agencés régulièrement et on garde l'ancienne structure. Pour une grille de gauss, on pourrait effectuer la même simplification.

Enfin, point important qui simplifiera le travail, pour une grille de gauss, a priori, on pourrait se permettre d'avoir un halo de zéro, en tout cas pour des configurations de SURFEX classiques. En effet, les données d'occupation du sol, de relief d'argile et de sable sont à meilleure résolution que la grille de gauss, ce qui évite a priori des interpolations. Toutefois, c'est à vérifier au cas où il y ait des inconsistances entre fichier d'occupation du sol (et donc l'indice terre-mer) et le fichier de relief.

4.2 Détermination de la zone géographique à lire dans les fichiers à accès direct.

Toujours pour la grille de gauss, puisqu'une sous-grille est différente (en structure) de la grille totale telle que définie actuellement, il faudra adapter les calculs permettant de déterminer les quarts de degrés carrés à lire dans les fichiers de données.

5 Quelques remarques techniques pour aider à la compréhension du code

a. NDIM et NSIZE

La variable `NDIM_FULL` est le nombre de points de surface dans la grille totale. De même, la variable `NDIM_SEA` est le nombre total de points de mer dans la grille totale. (pareil avec `TOWN`, `NATURE`, `WATER`)

La variable `NSIZE_FULL` quant à elle contient le nombre de points de la sous-grille avec son halo, donc celle sur le processeur en cours.

De même, la variable `NSIZE_SEA` contient le nombre de points contenant de la mer de la sous-grille avec son halo. (pareil avec `TOWN`, `NATURE`, `WATER`)

b. Les fonctions d'échange en cours de calcul

En cours de calcul, les seuls échanges de données entre processeurs concernent les échanges de quelques entiers (pas de tableaux d'entiers). Ils servent à compter le nombre de points du domaine de la grille totale pour lesquels une conditions logique est vérifiée. Ceci sert à :

- calculer le nombre de points de chaque tile (`SEA`, `WATER`, `TOWN`, `NATURE`)
- déterminer si il y a des types d'occupation du sol propres à `ecoclimap 2` (ce qui alors induit l'initialisation des tables des paramètres dérivés pour `ecoclimap2`)
- calculer le nombre de points qui ont été interpolés ou remplis à la valeur par défaut.

c. L'ajout ou la suppression de halos contrôlé par le programme maître

Les routines d'I/O sont incluses dans les `READ_SURF_type_de_système` et `WRITE_SURF_type_de_système`. Ce sont ces routines qui déterminent comment les champs sont découpés et échangés suite à une lecture ou avant une écriture. Ceci est évidemment propre à chaque système de modélisation.

En ce qui concerne les halos, il faut juste ajouter dans les routines d'écriture (les `WRITE_SURF`) la suppression des points de halo. Ces points de halo sont déterminés par le système de modélisation (`offline`, `MesoNH`, éventuellement `ARPEGE/ALADIN`) au moment du découpage en sous-grille (routine `SPLIT_GRID_PARAMETERS_type_de_système`).

A noter qu'après l'étape `PGD`, il n'est plus –actuellement- nécessaire d'avoir ces points de halo. Les calculs scientifiques sont (pour le moment) locaux. Le cas échéant, on pourra les rajouter au moment des lectures de champs, mais c'est plus compliqué : dans `PGD`, on a juste à déterminer où sont ces points (et ensuite `PGD` les rempli), alors que si on les relit dans le fichier, il faut aussi lire leur valeur.