

Couplage de codes avec OASIS

S. Valcke - Cerfacs

30 août 2017

Plan du cours

1. Introduction:

- Besoin et exemples de couplage de codes
- Quelques définitions
- Les contraintes du couplage de code
- Quelques technologies pour échanger des champs de couplage

2. Différents outils et infrastructures de couplage

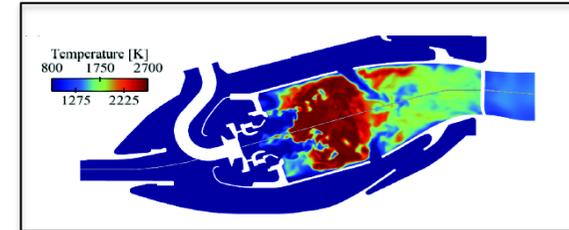
3. Le couplage océan-atmosphère

1. Introduction: Besoin et exemples de couplage de codes

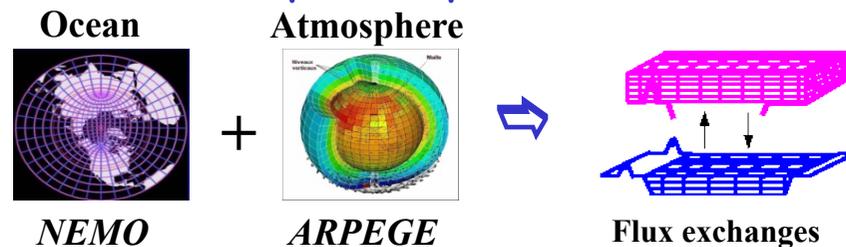
Pourquoi coupler des codes de calcul?

- Pour modéliser un système dans sa globalité en tenant compte des interactions des sous-systèmes

- ❖ Couplage fluide-structure:
la déformation de la structure a une influence directe sur l'écoulement

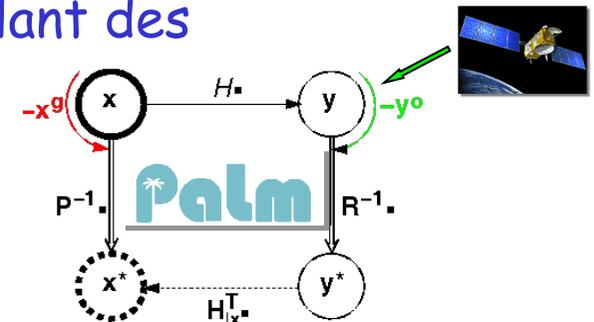


- ❖ Couplage océan-atmosphère pour la modélisation du climat



- Pour créer une application modulaire en assemblant des composants de calcul élémentaires:

- Une chaîne d'assimilation de donnée:
enchaînement d'une unité d'ingestion de données,
du code direct, du code adjoint, d'un minimiseur, etc.



1. Introduction: Quelques définitions

- **« Coupler » des codes:**
Echanger et transformer de l'information à l'interface de ces codes
Gérer l'exécution et la synchronisation des codes
- **Composante du couplage:**
Entité fonctionnelle matérialisée par un code numérique échangeant de l'information avec d'autres composantes
- **Champ de couplage:**
Tableau numérique (réels, entiers, etc.) échangé entre le code source (sortie) et le code cible (entrée)
- **Grilles de couplage:**
Grilles de discrétisation numérique des codes
- **Algorithme de couplage:**
Séquence d'échange entre les composantes
- **Composante dynamique:**
Composante dont les caractéristiques de couplage (champs, grille, etc.) varient en cours de simulation
- **Couplage dynamique:**
Couplage dont les composantes varient en cours de simulation

1. Introduction: Contraintes du couplage de codes

Quelles sont les contraintes du couplage?

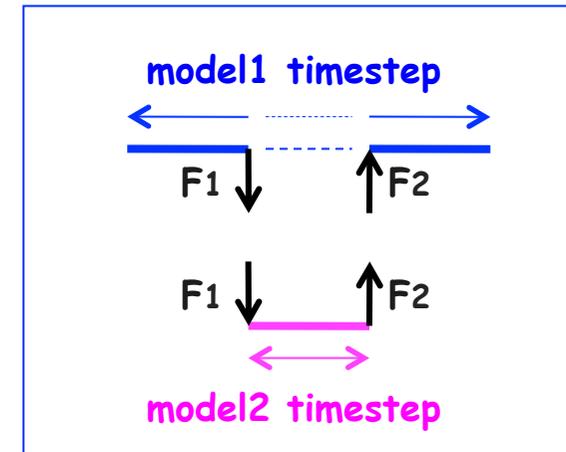
- ✓ Le couplage doit être **facile à implémenter et portable**
- ✓ Le couplage doit être **flexible** (changement des composantes, des champs de couplages, des fréquences d'échange, etc.)
- ✓ L'assemblage se fait à partir de **codes existants** développés de façon indépendante par des groupes différents
- ✓ Les **caractéristiques de la plateforme** de calcul (puissance du processeur, taille mémoire, etc.) et son **système d'administration** doivent être considérées
- ✓ L'**algorithme de couplage** est imposé par la science
- ✓ La **performance globale** du modèle couplé doit être bonne (cout CPU, besoin en mémoire, équilibrage des charges, etc.)

1. Introduction: Contraintes du couplage de codes

✓ L' algorithme de couplage est imposé par la science

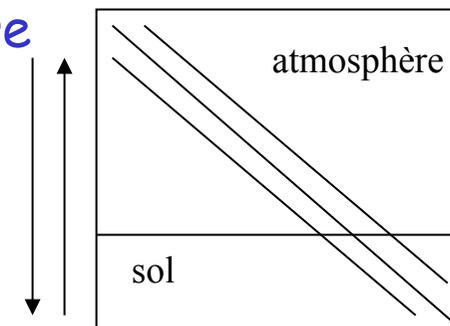
Les composantes peuvent être:

- **Séquentielles par nature:**
une composante est nécessairement inactive quand l'autre tourne et vice-versa



Exemple: résolution implicite de l' équation de diffusion de la chaleur de la base du schéma de sol jusqu' au sommet de l' atmosphère

$$\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial z^2}$$
$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = K \frac{T_{k+1}^{n+1} + T_{k-1}^{n+1} + 2T_k^{n+1}}{\Delta z^2}$$
$$AT^{n+1} = T^n$$

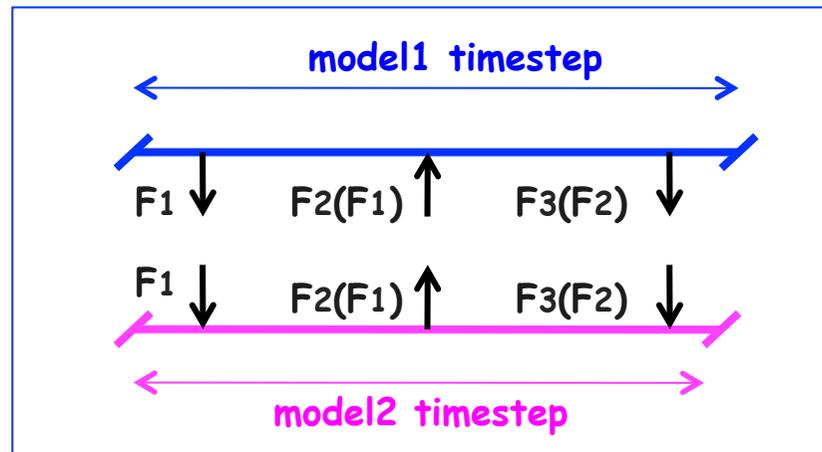


1. Introduction: Contraintes du couplage de codes

✓ L'algorithmme de couplage est imposé par la science

Les composantes peuvent être:

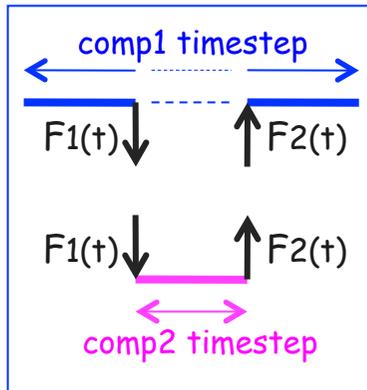
- **Concurrentes par nature :**
un champ de couplage produit par une composante à un pas de temps donné dépend d'un champ de couplage produit par l'autre composante au même pas de temps et vice-versa.



1. Introduction: Contraintes du couplage de codes

- ✓ La performance globale du modèle couplé doit être bonne

Sequential coupling :

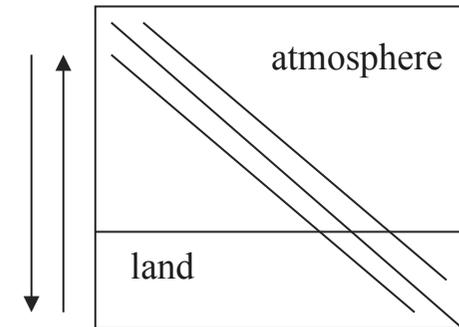


Implicit resolution of heat diffusion equation from the top of the atmosphere to the bottom of the land

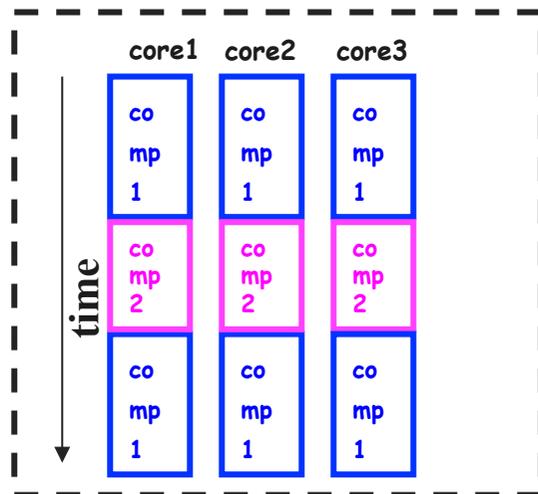
$$\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial z^2}$$

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = K \frac{T_{k+1}^{n+1} + T_{k-1}^{n+1} + 2T_k^{n+1}}{\Delta z^2}$$

$$AT^{n+1} = T^n$$



=> sequential execution on the same set of cores in one executable

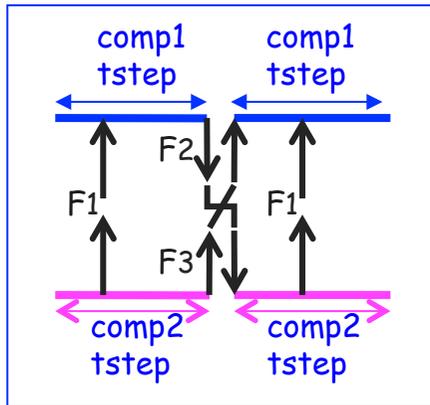


- ☺ Efficient coupling exchanges through the memory
- ☺ Optimal for load balancing if components can run efficiently on same number of cores
- ☹ Possible conflicts as components are merged in one executable (I/O, units, internal comm, etc.)
- ☹ No flexibility in coupling algorithm

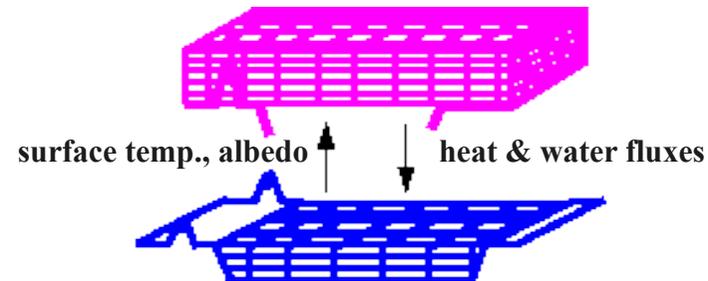
1. Introduction: Contraintes du couplage de codes

- ✓ La performance globale du modèle couplé doit être bonne

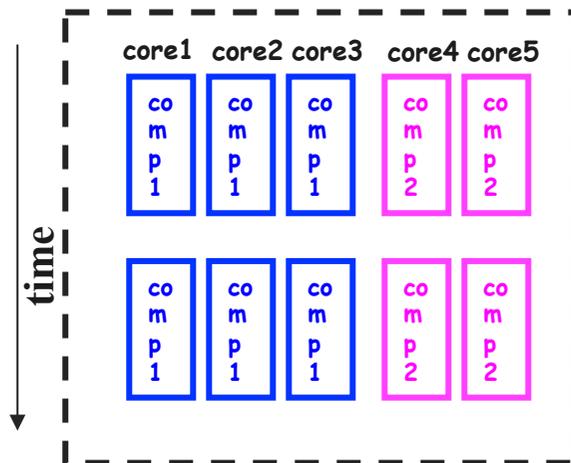
Concurrent coupling:



Traditional asynchronous ocean-atmosphere coupling



=> concurrent execution on different sets of cores within one executable

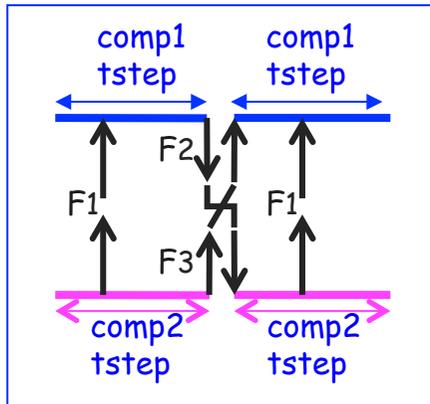


- ☺ Flexible coupling algorithm (exchanges in timestep)
- ☹ Possible conflicts as components are merged in one executable (I/O, units, internal comm, etc.)
- ☹ Less efficient coupling exchanges as components may be on different nodes (no shared memory)
- ☹ Harder load balancing

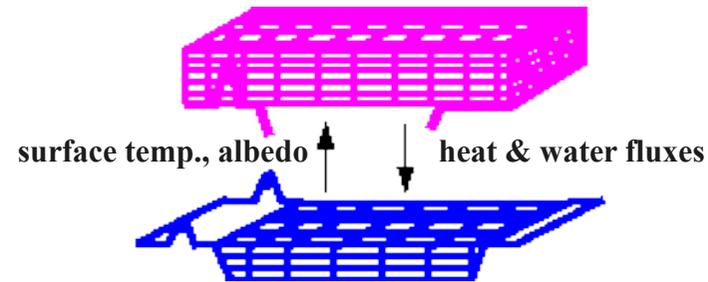
1. Introduction: Contraintes du couplage de codes

- ✓ La performance globale du modèle couplé doit être bonne

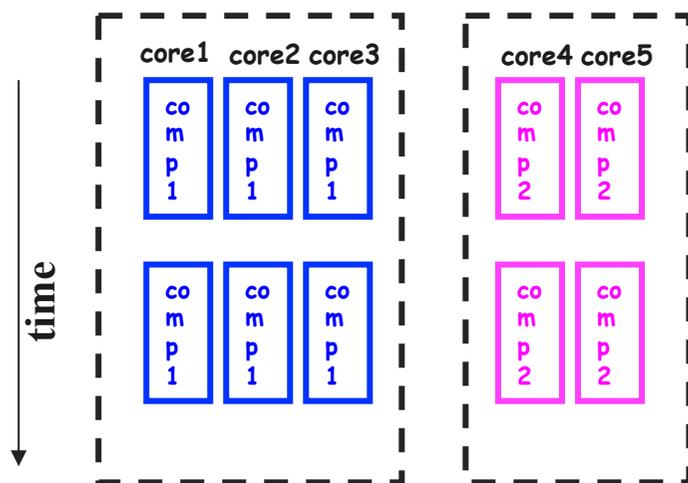
Concurrent coupling:



Traditional asynchronous ocean-atmosphere coupling



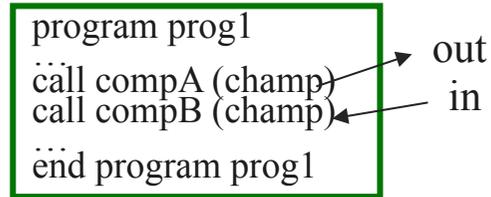
=> concurrent execution on different sets of cores within separate executables



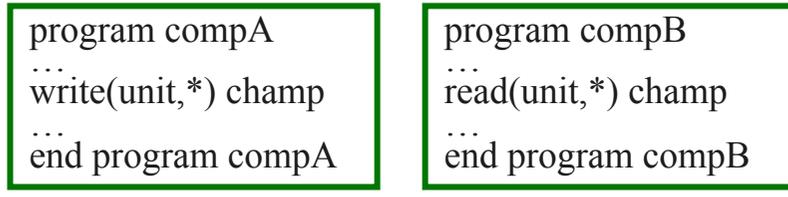
- ☺ Flexible coupling algorithm (exchanges in timestep)
- ☺ No conflicts as components remain separate executables (I/O, units, internal comm, etc.)
- ☺ Optimal use of memory
- ☹ Harder load balancing
- ☹ Less efficient coupling exchanges as components may be on different nodes (no shared memory)

1. Introduction: Quelques technologies pour échanger des champs de couplage

- **Partage de mémoire**



- **Ecriture/lecture de fichiers**



- **MPI (Message Passing Interface):**

norme définissant des fonctions (C et Fortran), devenue un standard de communication pour des programmes parallèles sur des systèmes à mémoire distribuée.

- **PVM (Parallel Virtual Machine):**

bibliothèque de communication (C et Fortran) pour machines parallèles et réseau d'ordinateurs (locaux ou distants, éventuellement hétérogènes).

- **TCPIP (Transmission Control Protocol/Internet Protocol):**

règles de communication sur internet; notion adressage IP pour acheminer des paquets de données.

- **SVIPC (System V Inter Process Communication):**

mécanismes de communication/ synchronisation Unix/Linux; partager de zones mémoire par des exécutables distincts; synchronisation lors de l'accès grâce déjà des sémaphores.

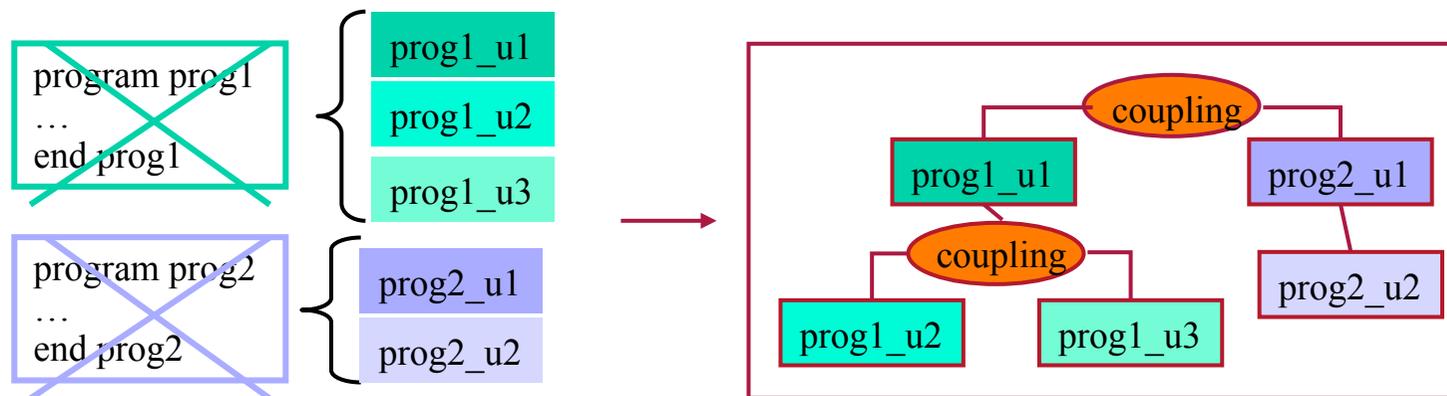
- **CORBA (Common Object Request Broker Architecture):**

architecture logicielle pour l'assemblage de composantes; exécutées dans des processus séparés; langages de programmation distincts; déployées sur des machines distinctes.

2. Différents outils et infrastructures de couplage

Approche 1: environnement logiciel avec bibliothèque de fonctions

- Diviser le code original en unités élémentaires
- Adapter les structures de données et l'interface d'appel de ces unités
- Ecrire ou utiliser des unités de couplage (transformation, interpolation)
- Utiliser l'environnement logiciel pour bâtir un **code unique intégrant les composantes**



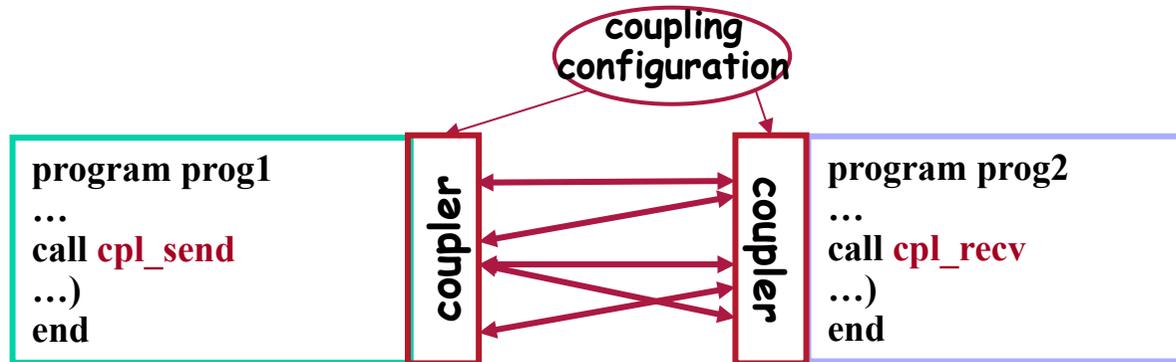
- ☺ Flexible, portable
- ☺ **Efficace**
- ☺ Utilisation d'utilitaires génériques de l'infrastructure logicielle (parallélisation, etc.)
- ☺ Composantes séquentielles et concurrentes

- ☹ codes existants
- ☹ facile

ESMF **FMS** **CESM**
(GFDL) (NCAR)

2. Différents outils et infrastructures de couplage

Approche 2: un coupleur et/ou une librairie de communication



- ☺ Codes existants
- ☺ Flexible, portable
- ☺ Utilisation d'utilitaires génériques du coupleur (interpolation, etc.)
- ☺ Composantes concurrentes

- ☹ Composantes séquentielles
- ☹ (efficace)



O-Palm



- **L'outil de couplage idéal unique n'existe pas**; l'outil adapté dépend de:
- possibilité/volonté de changer les codes originaux
 - l'environnement de calcul
 - efficacité/performance recherchée
 - fonctions annexes recherchées (transformations, interpolation, etc.), ...

2. Différents outils et infrastructures de couplage

• Approche 1:

- **ESMF**: Earth System Modeling Framework (US)
- **FMS**: Flexible Modelling System (GFDL) modèle couplé modulaire; échange facile des composantes; utilitaires (parallélisation, I/O, échanges).
- **CPL7 / CESM**: Community Earth System Model (NCAR)

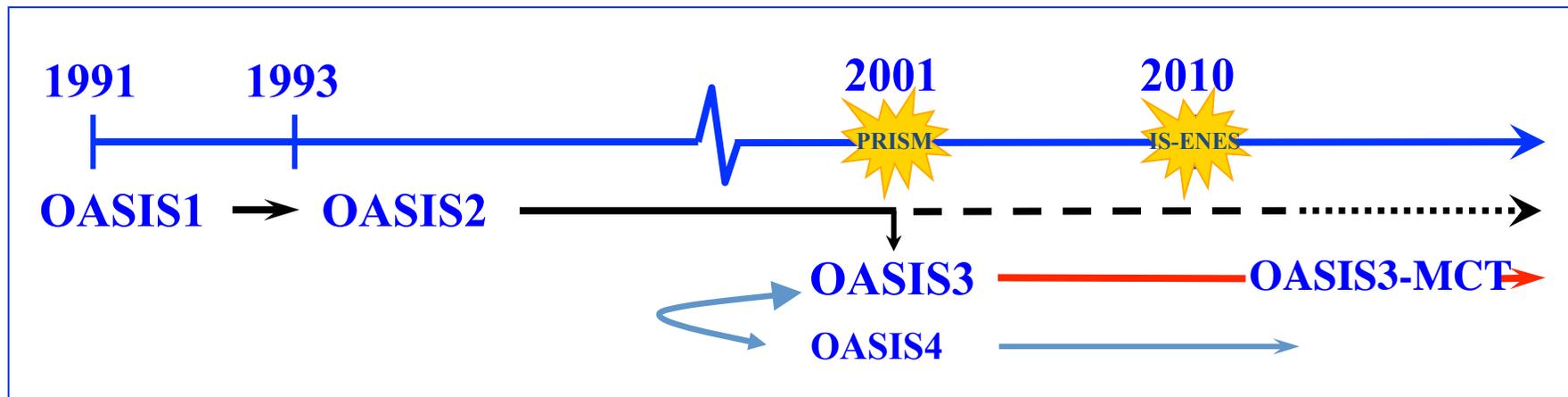
• Approche 2

- **TDT** (Type Data Transfer, Potsdam Institute for Climate) : transfert de données entre programmes (langages différents, plateformes différentes, divers protocoles : Unix Socket, I/O fichiers, MPI)
- **MpCCI** (Fraunhofer SCAI): Interface de couplage pour coupler des codes indépendants (commercial, pas « open source »).
- **Open-PALM, OASIS** (CERFACS)
- **YAC** (Yet Another Coupler) : DKRZ (Germany)

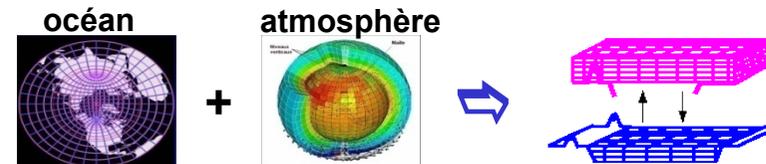


2. Différents outils et infrastructures de couplage - OASIS

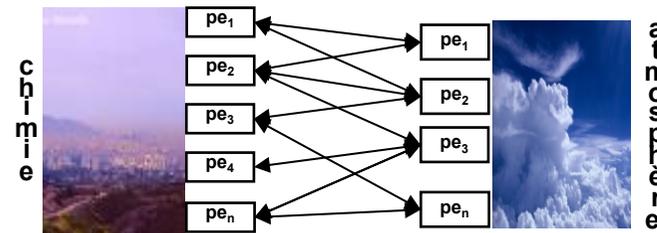
Le coupleur OASIS



- OASIS1 -> OASIS2 -> OASIS3:
Couplage océan-atmosphère 2D
à basse fréquence et basse résolution:
→ **Flexibilité, interpolations 2D**



- OASIS4 / OASIS3-MCT:
Couplage climat 2D/3D, composantes et plateformes
→ **Parallélisme, performance**



➤ F90 & C, licence LGPL, librairies domaine publique (MPI, NetCDF, libXML, mpp_io, SCRIP)



2. Différents outils et infrastructures de couplage - OASIS

Plus de 40 groupes en France, Europe et au-delà pour la modélisation du climat et la prévision saisonnière, e.g.:

- France: CERFACS, CNRM, LOCEAN, LMD, LSCE, LA, LEGOS, LGGE, IFREMER, ENSTA
 - Europe: ECMWF + EC-Earth community
 - Allemagne: MPI-M, IFM-GEOMAR, HZG, U. Frankfurt, BTU-Cottbus
 - UK: MetOffice, NCAS/U. Reading, ICL,
 - Danemark: DMI
 - Norvège: U. Bergen
 - Suède: SMHI, U. Lund
 - Irlande: ICHEC, NUI Galway
 - Pays-Bas: KNMI
 - Belgique: KU Leuven
 - Suisse: ETH Zurich
 - Italie: INGV, ENEA, CASPUR
 - République tchèque :CHMI
 - Espagne: IC3, BSC, U. Castilla
 - Tunisie: Inst. Nat. Met
 - Arabie Saoudite: CECCR
 - Japon: U. Tokyo, JMA, JAMSTEC
 - Chine: IAP-CAS, Met. Nat. Centre, SCSIO
 - Corée: KMA
 - Australie: CSIRO, BoM, ACT, NCI
 - Nouvelle-Zélande: NIWA, NCWAR
 - Canada: Environnement Canada, UQAM
 - USA: Oregon St. U., Hawaii U., JPL, MIT
 - Pérou: IGP
- + téléchargements du Nigeria, Colombie, Singapour, Russie, Thaïlande, ...

OASIS3 utilisé dans les 2 ESMs français et 5/7 ESMs Européens participant à CMIP6



2. Différents outils et infrastructures de couplage - OASIS

Comment utiliser OASIS

- Identifier les champs de couplage à échanger entre les composantes
- Interfacer les codes avec la librairie de communication d' OASIS
 - Initialisation: `call oasis_init_comp (...)`
 - Définition des grilles (longitudes, latitudes) : `call oasis_write_grid (...)`
 - Définition de la partition locale: `call oasis_def_partition(...)`
 - Déclaration des champs de couplage `call oasis_def_var(...)`
 - Envoi et reception des champs de couplage (dans le pas de temps des composantes)
`Call oasis_put (var_id, date, var_array, info, ierr)`
`Call oasis_get (var_id, date, var_array, info, ierr)`
Ce qui est défini par l'utilisateur dans le fichier de configuration:
 - La cible/source d'un prism_put/prism_get (communication «end-point »)
 - Les temps auxquels s'effectuent automatiquement les envois et réceptions
 - Moyennes, accumulations, interpolations entre la source et la cible, etc.
- Choisir les paramètres du couplage et écrire le fichier de configuration
- Compiler OASIS et les composantes (lien avec la librairie de comm), les démarrer et laisser OASIS et la librairie gérer les échanges tels que configurés.



2. Différents outils et infrastructures de couplage - OASIS

La configuration d'un modèle couplé avec OASIS

• Pour OASIS3 / OASIS3-MCT: dans un fichier texte *namcouple*:
** un interface graphique basé sur OPENTEA du Cerfacs **

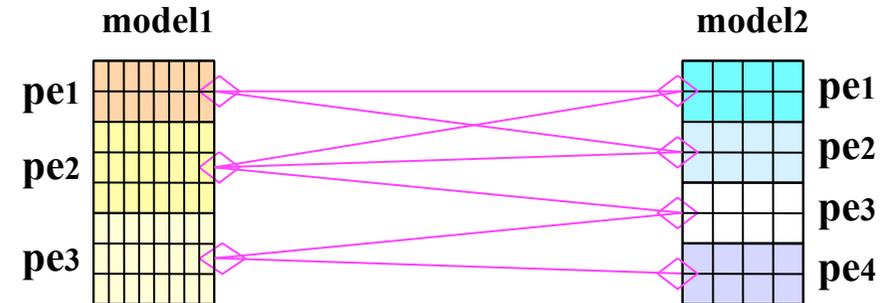
- temps total de la simulation
- composantes modèles à coupler
- pour chaque champ de couplage à échanger:
 - Nom symbolique dans la composante source et nom symbolique dans la composante cible
 - Période de couplage
 - Transformations/interpolations à faire par OASIS



2. Différents outils et infrastructures de couplage - OASIS

La communication dans OASIS3-MCT (MPI1)

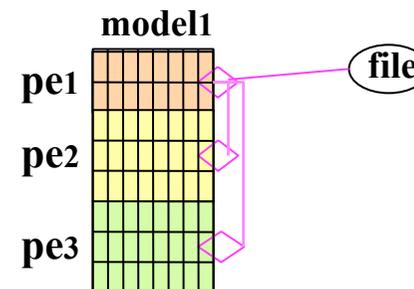
• Communication parallèle entre les modèles parallèles (grilles et partitions différentes) :



Si besoin, les poids & adresses requis pour les interpolations sont calculés par un processus.

L'interpolation en tant que telle est faite de la grille source vers la grille cible sur les processus sources ou cibles.

• Entrées/sorties (I/O): passage du mode couplé au mode forcé

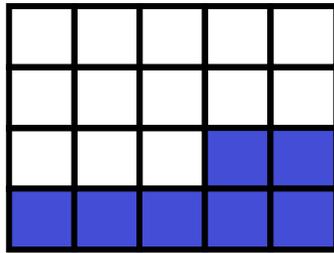




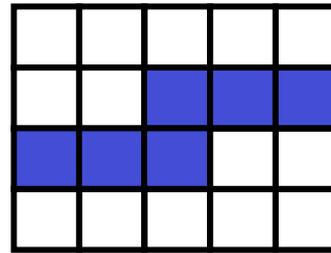
2. Différents outils et infrastructures de couplage - OASIS

Partitions supportées

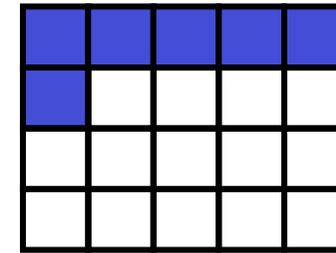
Apple



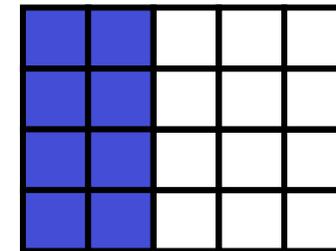
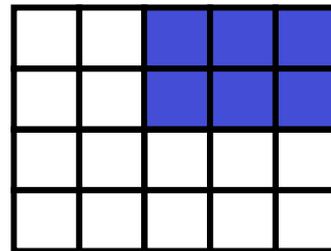
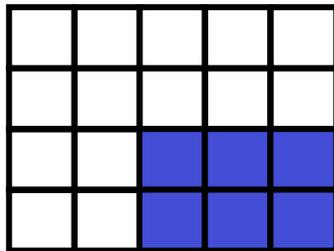
part2



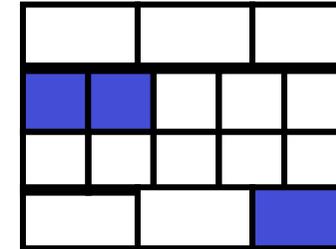
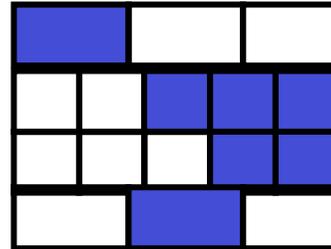
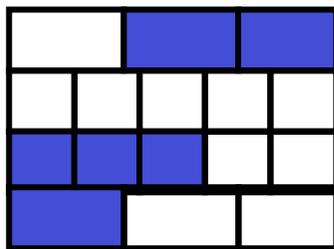
part3



Box



Orange



Apple et orange applicables aux grilles non structurées



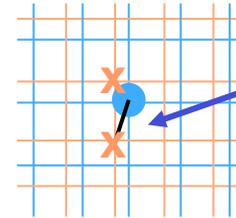
2. Différents outils et infrastructures de couplage - OASIS

Les algorithmes d'interpolation dans OASIS

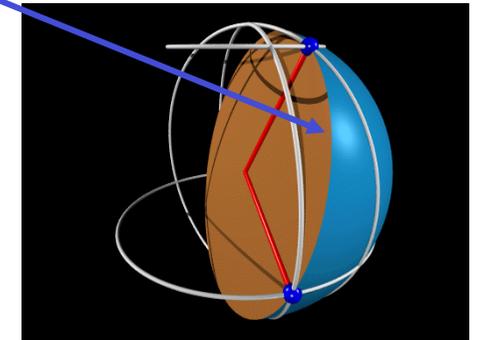
- n-plus-proches voisins (2D): poids(x) $\propto 1/d$

d: distance du grand cercle sur la sphère:

$$d = \arccos[\sin(\text{lat1}) * \sin(\text{lat2}) + \cos(\text{lat1}) * \cos(\text{lat2}) * \cos(\text{lon1} - \text{lon2})]$$



x: point source
● point cible

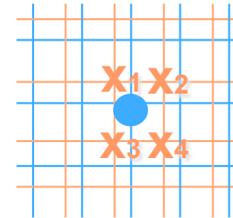


- n-plus-proches voisins pondérés : poids(x) $\propto \exp(-1/2 d^2/\sigma^2)$

- interpolation bilinéaire

➤ interpolation bilinéaire générale dans un système de coordonnées locales en utilisant

f(x) at x_1, x_2, x_3, x_4



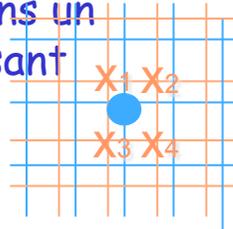
- interpolation bicubique: conserve les propriétés de 2^e ordre comme le rotationnel du vent

➤ interpolation bicubique générale dans un système de coordonnées locales utilisant

f(x), $\delta f(x)/\delta i$, $\delta f(x)/\delta j$, $\delta^2 f/\delta i \delta j$ en

x_1, x_2, x_3, x_4

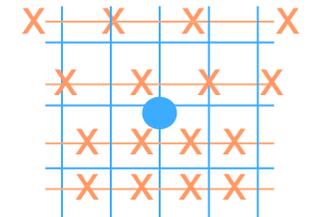
grilles logiquement-rectangles (i,j)



➤ algorithme bicubique standard:

16 points voisins

Grilles Gaussiennes Réduites



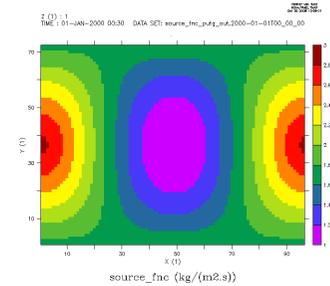
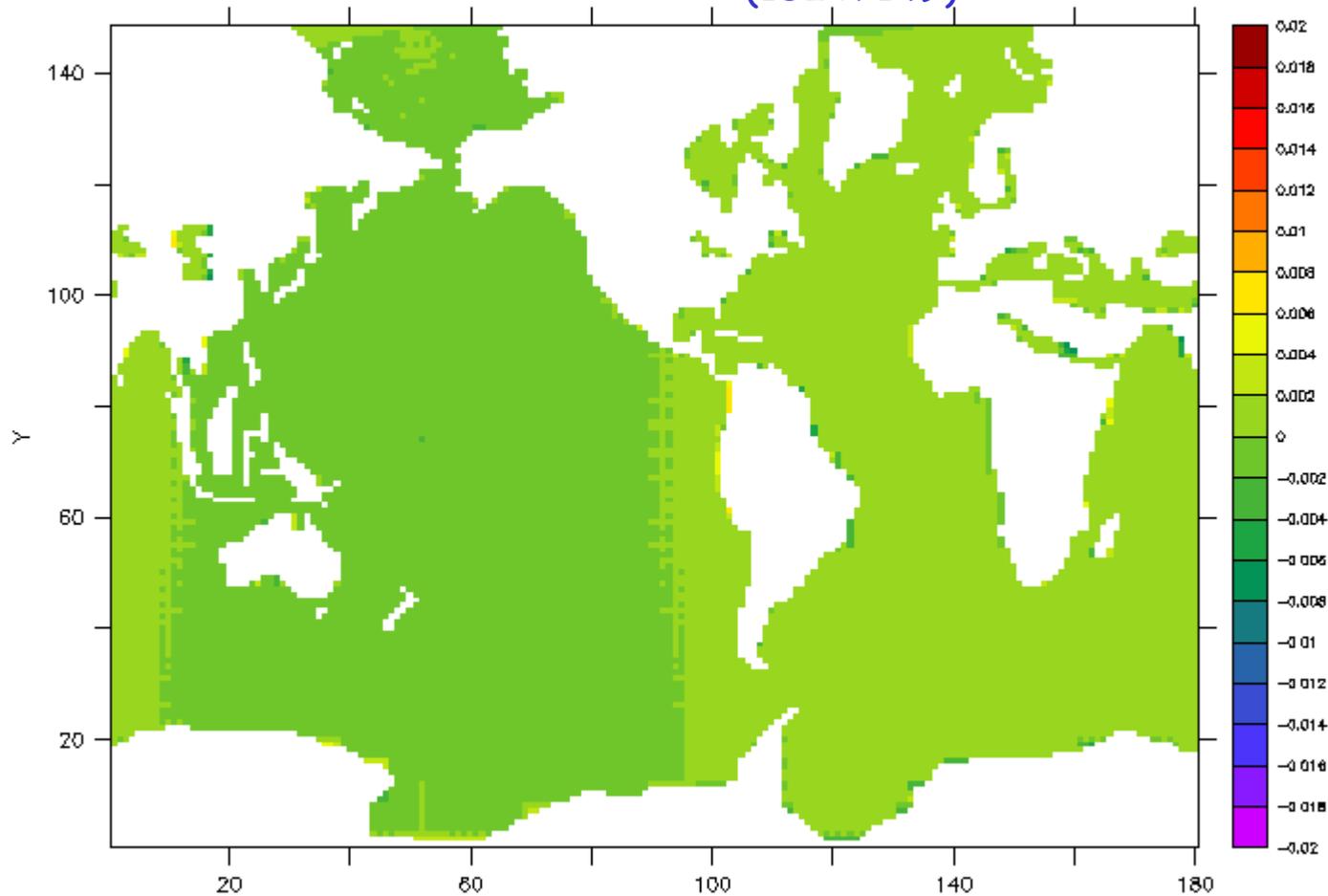


2. Différents outils et infrastructures de couplage - OASIS

Un exemple de l'erreur d'interpolation bilinéaire

$$F = 2 + \cos[\pi * \text{acos}(\cos(\text{lon})\cos(\text{lat}))]$$

LMDz grid (96 x 72) → ORCA2
(182 x 149)



➤ < 0.2% en général ~1% près des cotes



2. Différents outils et infrastructures de couplage - OASIS

Les algorithmes d'interpolation dans OASIS

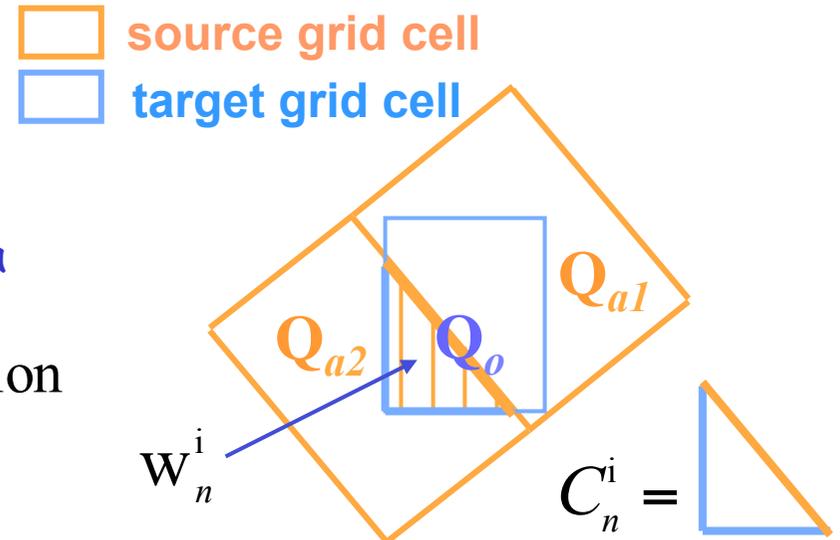
• 1st order conservative remapping:

- conserves integral of extensive properties
- weight of a source cell α to intersected area

$$Q_o^i = \frac{1}{A_o} \sum_{n=1}^N Q_{an} W_n^i \quad \text{with} \quad W_n^i = \int_{C_n^i} -\sin(\text{lat}) d\text{lon}$$

❖ assumes borders are linear in (lat,lon)

- Lambert equivalent azimuthal projection near the pole for intersec. calc.



Actual limitations:

- assumes $\sin(\text{lat})$ linear function of lon (for line integral calculation)
 - need to use a projection near the pole (as done for intersect. calc.)
- exact calculation is not possible as "real shape" of the borders are not known
 - could use of border middle point
 - to ensure conservation, need to normalize by true area of the cells

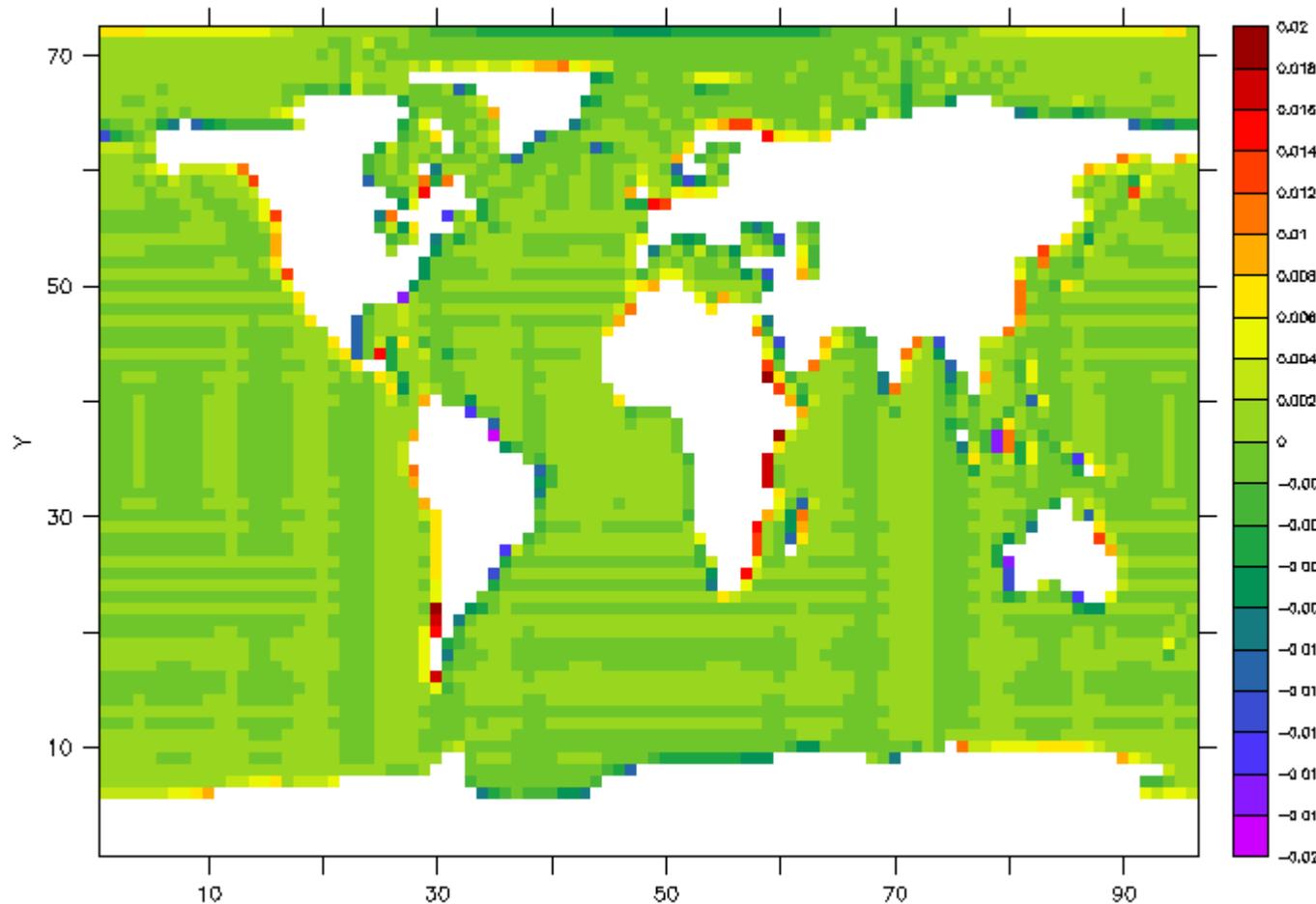


2. Différents outils et infrastructures de couplage - OASIS

- One example of conservative remapping error

$$F = 2 - \cos[\pi * \text{acos}(\cos(\text{lon})\cos(\text{lat}))]$$

ORCA2 -> LMDz (96x72)

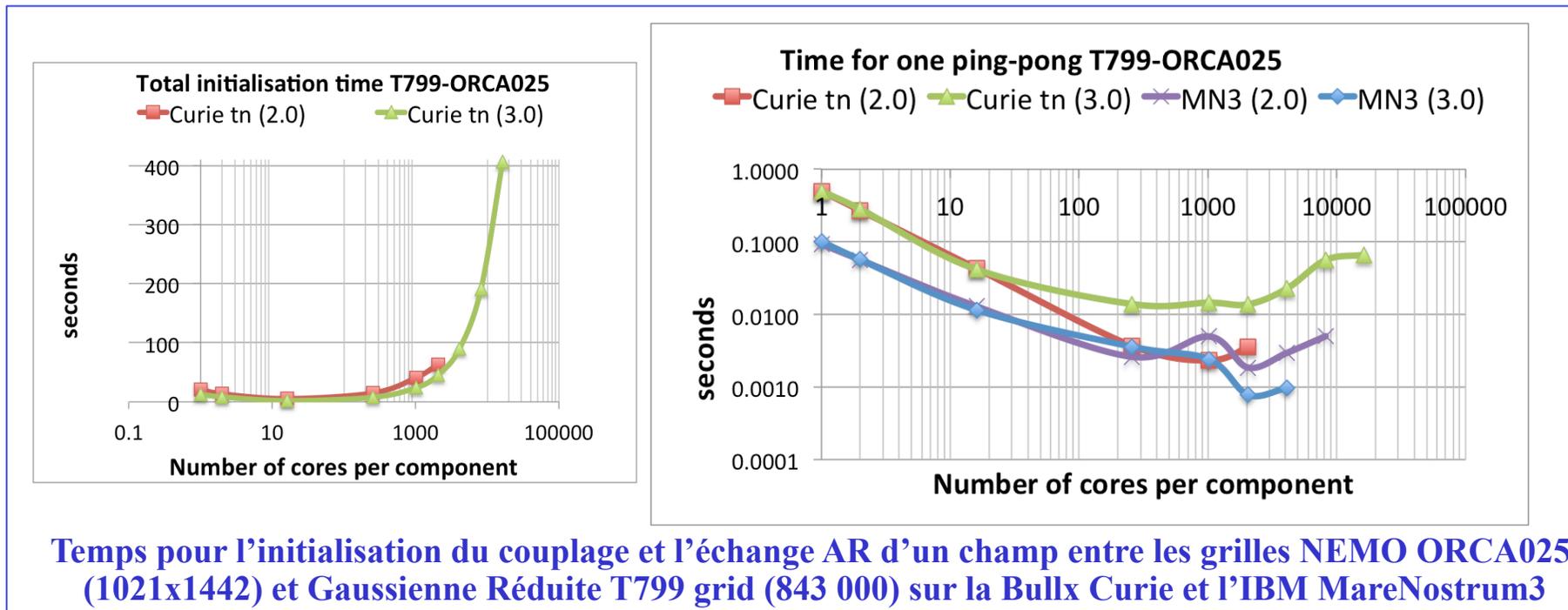


- < 0.2% everywhere except
~ 0.8% for LMDz last row close to the North pole
~ 2% near the coastline



2. Différents outils et infrastructures de couplage - OASIS

- Toy coupled model: ping-pong exchanges between NEMO ORCA025 grid (1021x1442) and Gaussian Reduced T799 grid (843 000)
- Bullx Curie thin nodes, IBM MareNostrum3

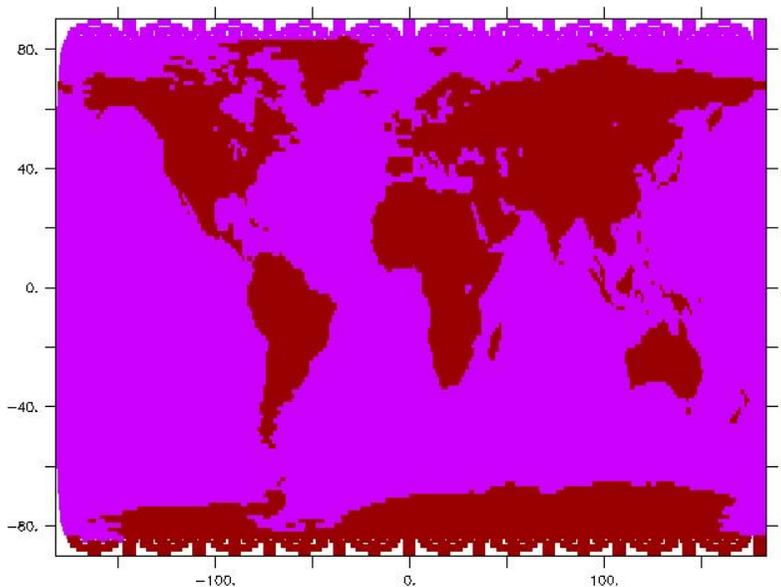


Coupling overhead for one-year long simulation with one 1 coupling exchange every 3 hours in each direction between codes with $O(1\text{ M})$ grid points running on $O(10\text{K})$ cores/component:

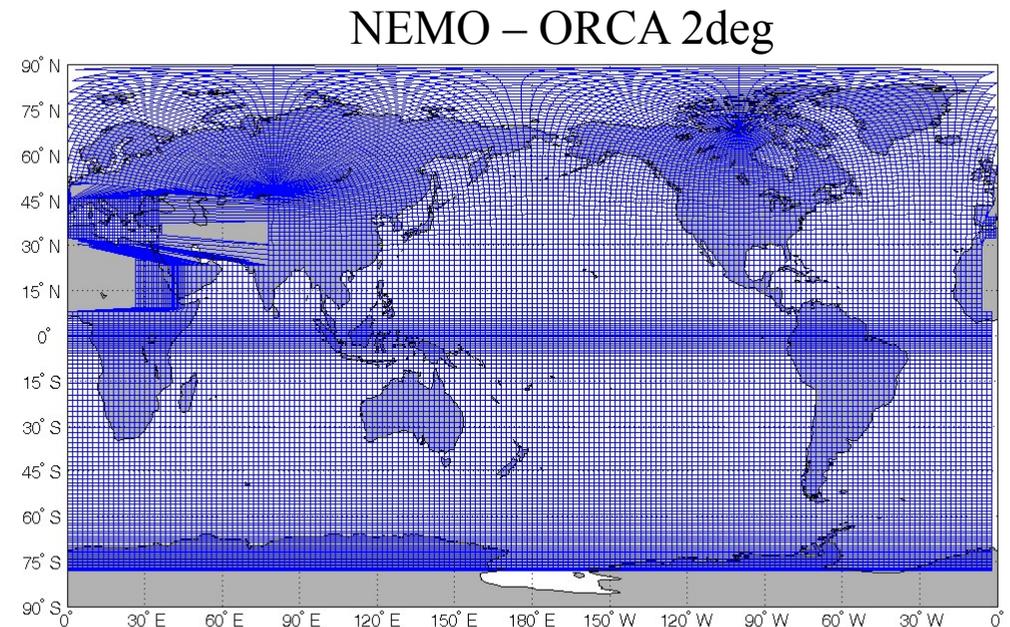
- ~7 mins for initialisation, ~5 mins for data exchange

3. Le couplage océan-atmosphère

1. Le problème des lignes de côtes différentes dans l'océan et l'atmosphère
2. L'interpolation conservative de premier ordre
3. L'interpolation des champs vectoriels
4. L'interface de couplage physique



ARPEGE T127

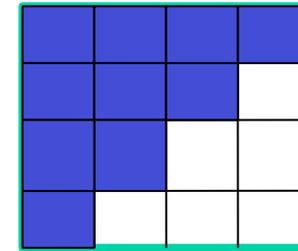


3. Le couplage océan-atmosphère

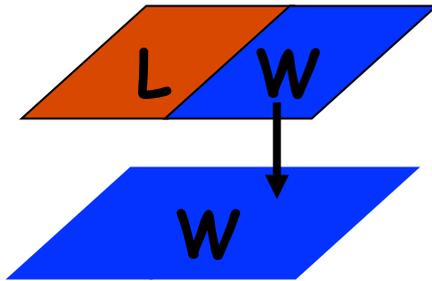
1. Le problème des lignes de côtes différentes dans l'océan et l'atmosphère

1- Support subsurfaces in the atmosphere

and use the ocean land-sea mask in the atmosphere to determine the fractional area of each type of surface



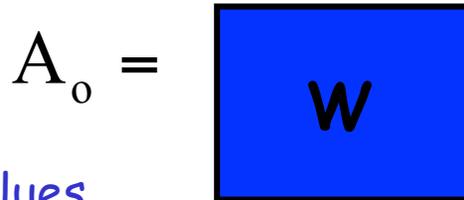
fw=10/16
fl=6/16



$$Q_o^i = \frac{1}{A_o} \sum_{n=1}^N Q_{an} W_n^i$$

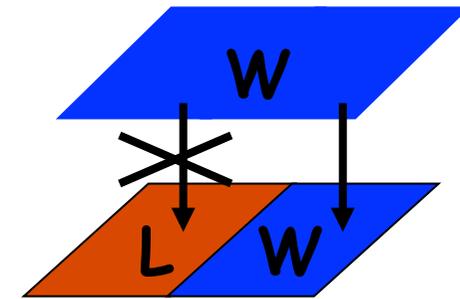
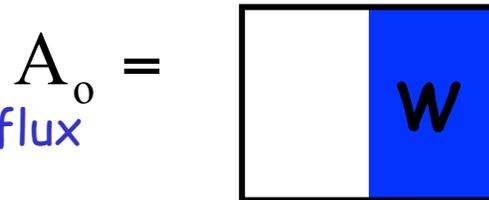
2-"DESTAREA" option

- local flux conservation
- possibly unrealistic flux values



3-"FRACAREA" option

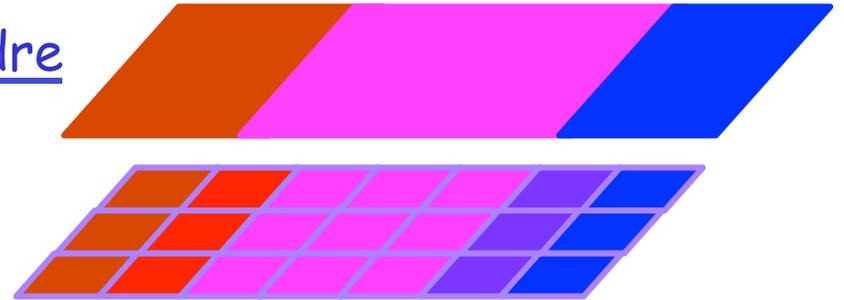
- no local conservation of flux
- realistic flux values
- + nearest non-masked value for ocean cells covered only with masked atmospheric cells



- no local conservation of flux
- global conservation can be artificially imposed

3. Le couplage océan-atmosphère

2. L' interpolation conservative de premier ordre (basse vers haute resolution)



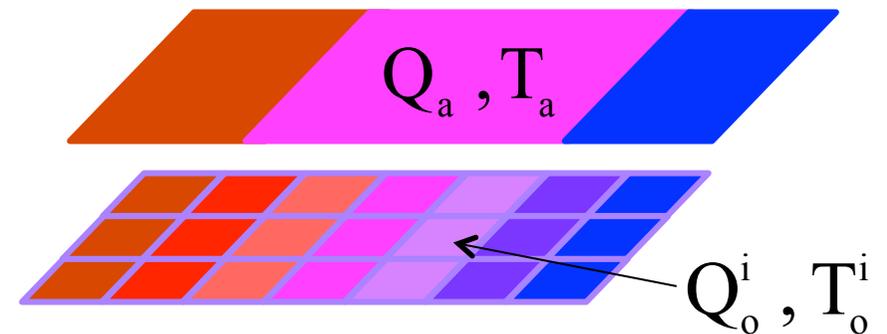
- Solution 1: use 2nd order conservative remapping:

$$Q_o^i = Q_a w_1^i + \frac{\partial Q_a}{\partial lat} w_2^i + \frac{1}{\cos(lat)} \frac{\partial Q_a}{\partial lon} w_3^i$$

- Solution 2: use SUBGRID transformation:

Solar type:
$$Q_o^i = \frac{(1 - \alpha_o^i)}{(1 - \alpha_a)} Q_a$$

Non-solar type:
$$Q_o^i = Q_a + \left. \frac{\partial Q_a}{\partial T} \right|_{T=T_a} (T_o^i - T_a)$$



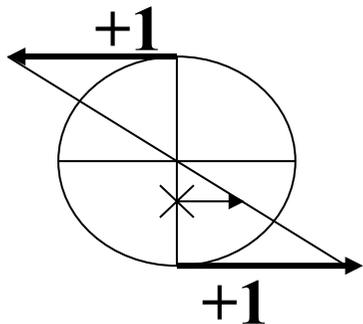
*conservative if α_a / T_a correspond to conservative remapping of α_o / T_o

3. Le couplage océan-atmosphère

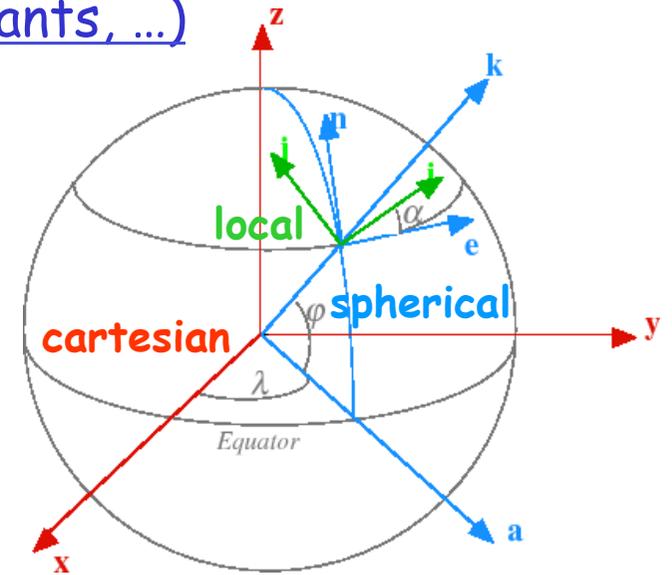
3. L'interpolation des champs vectoriels (vents, courants, ...)

- ❖ interpolation of vectors component per component in the spherical referential is not correct, especially where the referential changes rapidly

Example interpolation of a zonal wind in the spherical referential near the pole



- At x , one would expect a zonal wind between 0 and 1.
- Interpolation comp. per comp. \rightarrow zonal wind of 1.



Solution : project the vector in a cartesian referential, interpolate the 3 resulting components in this cartesian referential, and back in spherical referential

Fin