CERFACS

EUROPEAN CENTRE FOR RESEARCH AND ADVANCED TRAINING IN SCIENTIFIC COMPUTING

Code Coupling with OASIS3-MCT

S. Valcke, CERFACS

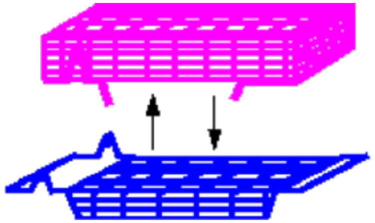
Training course, April 2021 29th

Outline

- Introduction
- Global performance of a coupled system
- Different technical coupling solutions
- OASIS historical overview, generalities, community
- Use of OASIS3-MCT *** this on-line course ***
  - code interfacing
  - coupled model configuration
- OASIS3-MCT parallel communication
- Interpolations et transformations
- Performances
- Conclusions and perspectives

Introduction

Why couple ocean, atmosphere, land, ocean, sea-ice models?



- Of course, to treat the Earth System globally

What does "coupling of codes" imply?

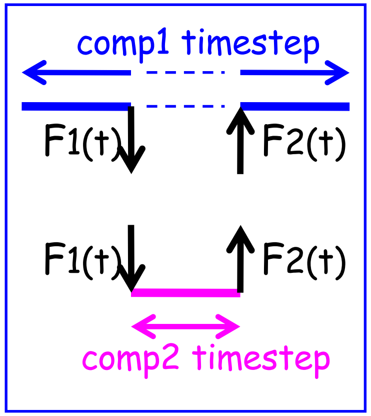
- Exchange and transform information at the code interface
- Manage the execution and synchronization of the codes

What are the constraints?

- ✓ Coupling should be easy to implement, flexible, efficient, portable
- ✓ Coupling algorithm dictated by science (sequ. vs conc. coupling)
- ✓ Start from existing and independently developed codes
- ✓ Global performance and load balancing issues are crucial
- ✓ Platform characteristics (OS, CPU, message passing efficiency, …)

Global performance of a coupled system - overview

Sequential coupling :

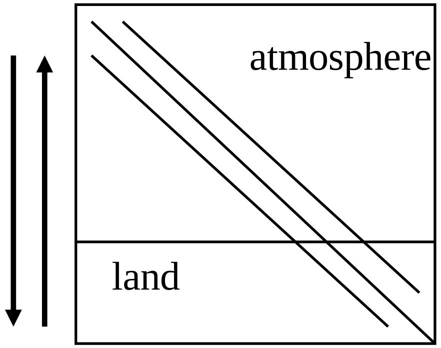


Implicit resolution of heat diffusion equation from the top of the atmosphere to the bottom of the land

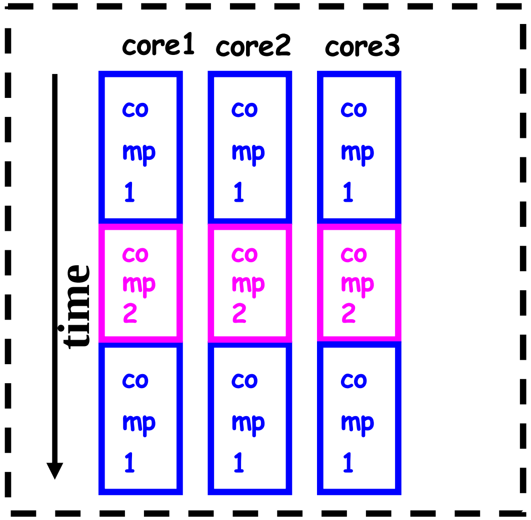
$$\partial T / \partial t = K \, \partial^2 T / \partial z^2$$

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

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



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



☺ Efficient coupling exchanges through the memory

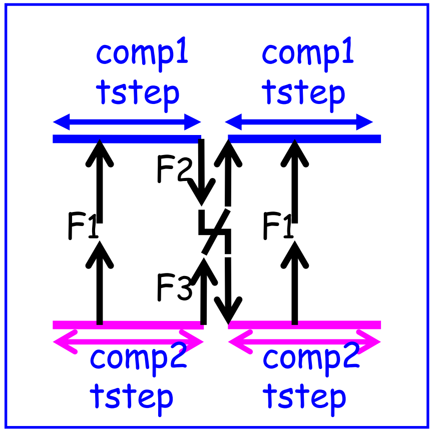
☺ Optimal for load balancing if components can run efficiently on same number of cores

☹ Possible conflicts as components are merged in one executable (I/O, units, internal comm, etc.)

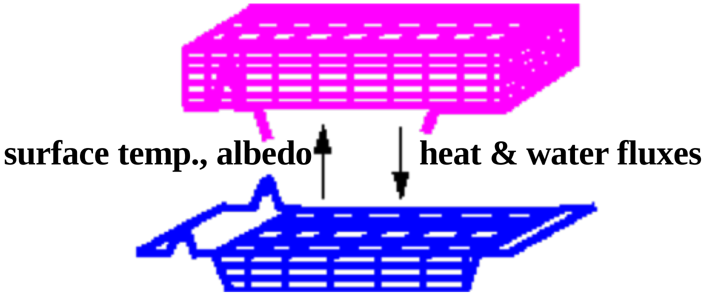
☹ No flexibility in coupling algorithm

Global performance of a coupled system - overview

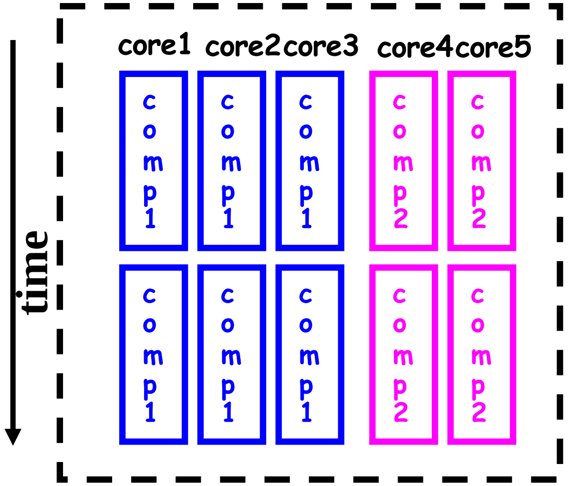
Concurrent coupling:



Traditional asynchronous ocean-atmosphere coupling



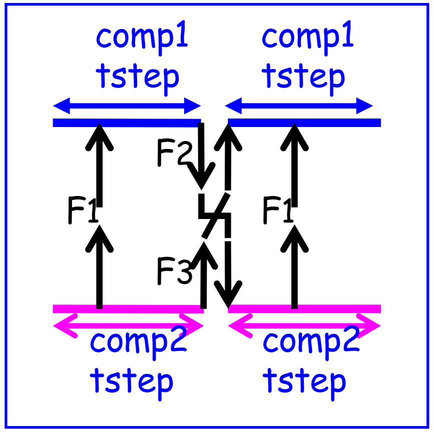
=> concurrent execution on different sets of cores within one executable



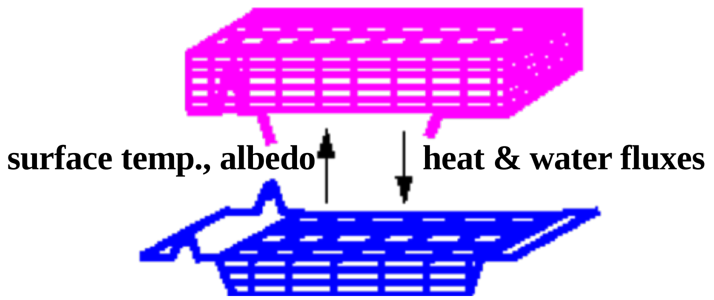
- ☺ Flexible coupling algorithm (exchanges in timestep)
- ☹ Possible conflicts as components are merged in one executable (I/O, units, internal comm, etc.)
- ☹ Less efficient coupling exchanges as components may be on different nodes (no shared memory)
- ☹ Harder load balancing

Global performance of a coupled system - overview

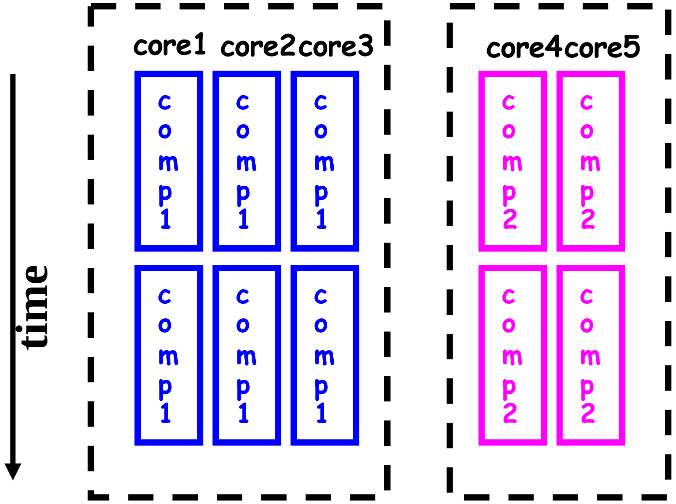
Concurrent coupling:



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.)
- ☹ Harder load balancing
- ☹ Less efficient coupling exchanges as components may be on different nodes (no shared memory)

Different technical coupling solutions

1. merging the codes:

```
program prog1
…
call sub_prog2(data)
…
end prog1
```

→

```
~~program prog2~~
subroutine sub_prog2(data)
…
end prog2
```

- ☺ efficient (memory exchange)
- ☺ as portable as the codes
- ☺ one executable: easier to debug, easier for the OS
- ☺ sequential execution of the components

- ☹ not easy to implement with existing codes (splitting, conflicts in namespaces and I/O)
- ☹ not flexible (coupling algorithm hard coded)
- ☹ no use of generic transformations/interpolations

Different technical coupling solutions

2. existing communication protocole (MPI, CORBA, UNIX pipe, files, …)

```
program prog1
…
call xxx_send (prog2, data, …)
end
```

```
program prog2
…
call xxx_recv (prog1,data)
end
```

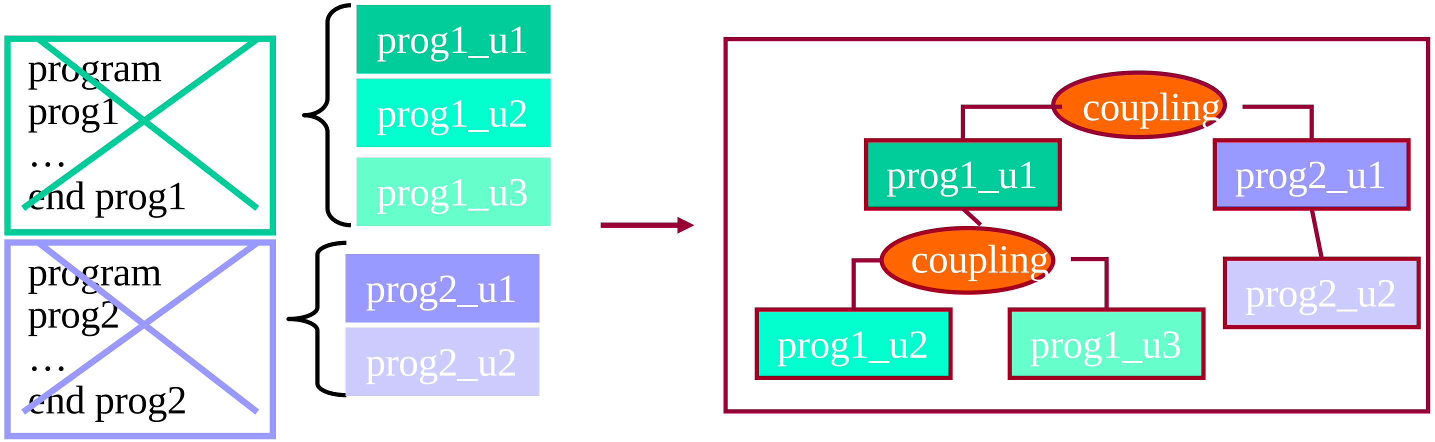
- ☺ existing codes
- ☺ natural parallelism in the execution of the components

- ☹ not easy to implement (needs protocol expert)
- ☹ not flexible (hard coded exchanges)
- ☹ no use of generic transformations/interpolations
- 😐 efficient, portable

Different technical coupling solutions

3. integrated coupling framework ESMF **FMS**(GFDL) *CESM* (NCAR)

- Split code into elemental units
- Write or use coupling units
- Use the library to build a hierarchical merged code
- Adapt code data structure and calling interface



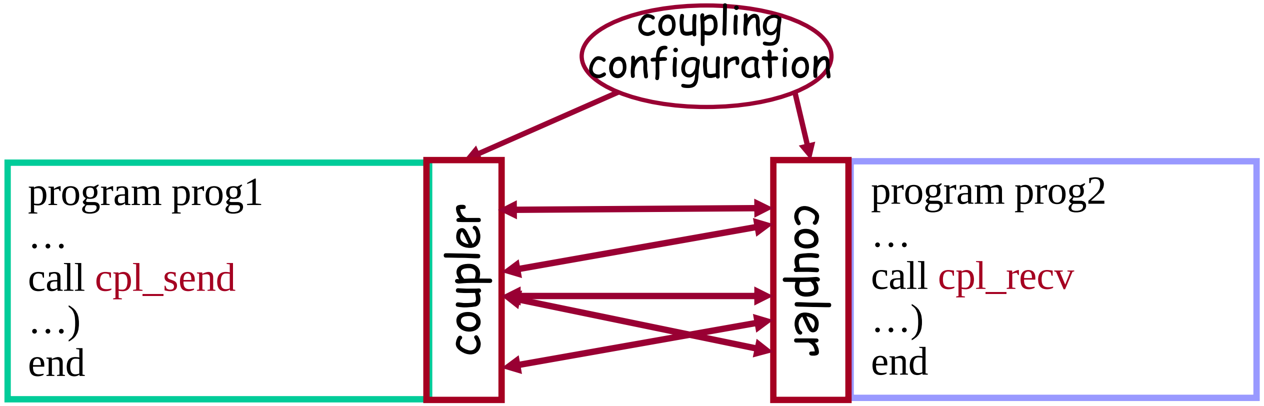
☺ efficient,
☺ sequential and concurrent components
☺ use of generic utilities (parallelisation, regridding, time management, etc.)

☹ existing codes
😐 (easy)

→ probably best solution in controlled development environment

Different technical coupling solutions

4. coupler or coupling library


coupling configuration

program prog1
…
call cpl_send
…)
end

coupler

coupler

program prog2
…
call cpl_recv
…)
end

- ☺ existing codes
- ☺ use of generic transformations/regridding
- ☺ concurrent coupling (parallelism)

- ☹ efficient
- ☹ multi-executable: more difficult to debug; harder to manage for the OS

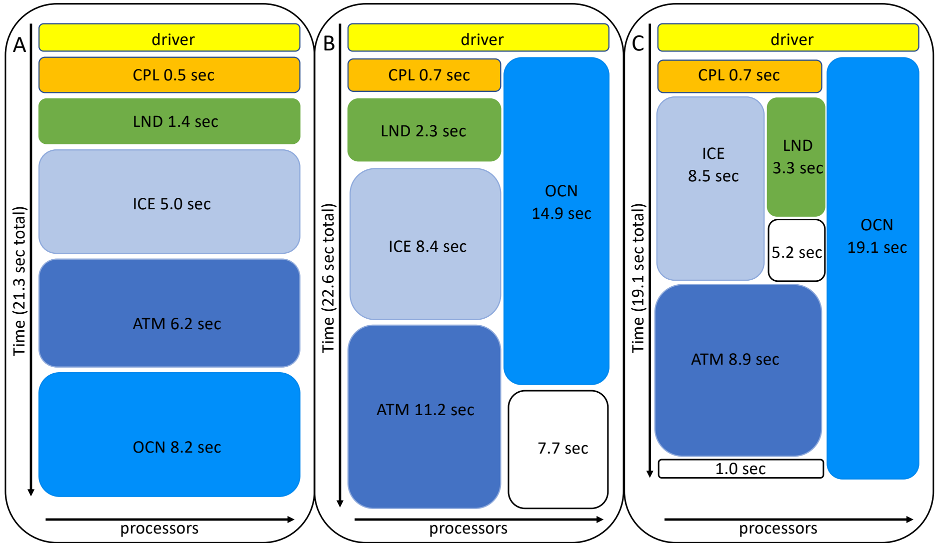
→ probably best solution to couple independently developed codes

Different technical coupling solutions

integrated coupling framework

- **ESMF/NUOPC**: Earth System Modeling Framework / National Unified Operational Prediction Capability (US)
- **FMS**: Flexible Modelling System (GFDL)
- **CPL7/CESM:**
  Community Earth System Model (NCAR)

The driver can launch the different components following different layouts



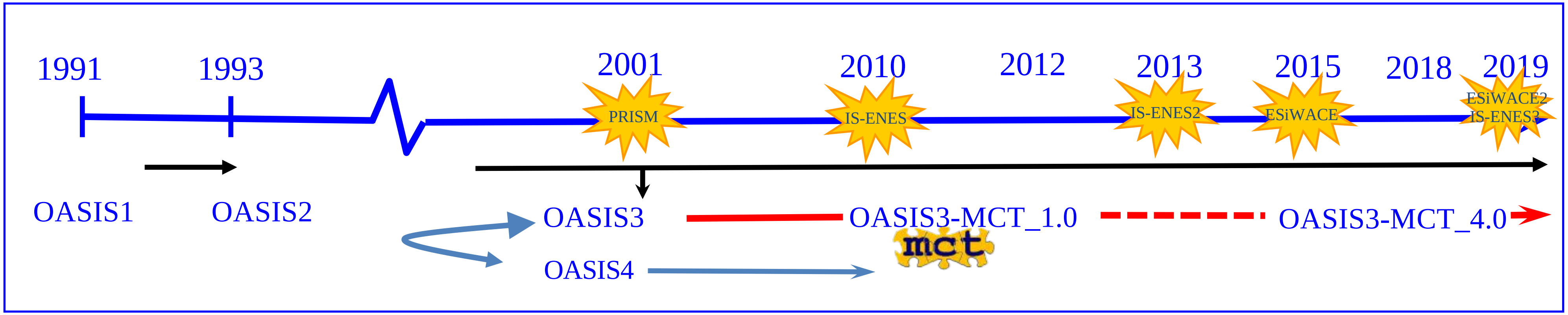
coupler or coupling library

- **OASIS3-MCT , Open-PALM** (CERFACS)
- **MCT** (USA) : Argonne National Lab
- **YAC** (Yet Another Coupler) (DKRZ, Germany)
- **C-Coupler2** (China)
- **MOAB-TempestRemap** (USA): Energy Exascale Earth System Model

Outline

- Introduction
- Global performance of a coupled system
- Different technical coupling solutions
- OASIS historical overview, generalities, community
- Use of OASIS3-MCT:
  - code interfacing
  - coupled model configuration
- OASIS3-MCT parallel communication
- Interpolations et transformations
- Performances
- Conclusions and perspectives

OASIS historical overview



1991 1993 2001 2010 2012 2013 2015 2018 2019

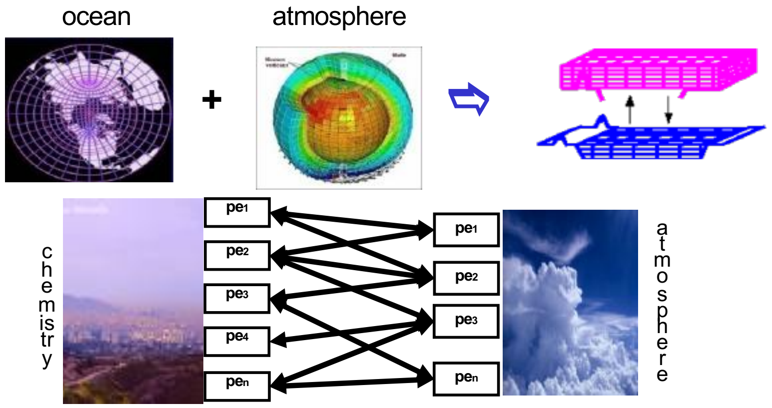
PRISM IS-ENES IS-ENES2 ESiWACE ESiWACE2 IS-ENES3

OASIS1 OASIS2 OASIS3 OASIS3-MCT_1.0 OASIS3-MCT_4.0

OASIS4

mct

- OASIS1 -> OASIS2 -> OASIS3:
  - 2D ocean-atmosphere coupling
  - low frequency, low resolution :
  - → Flexibility, 2D interpolations
- OASIS4 / OASIS3-MCT:
  - 2D/3D coupling of high-resolution parallel components
  - →Parallelism, performance



OASIS3-MCT community

OASIS3-MCT current users
2019 survey

67 climate modelling groups around the world use OASIS3-MCT …

…. to assemble more than 80 coupled applications !!

OASIS3-MCT is used in 5 of the 7 European ESMs participating to CMIP6

OASIS3-MCT: some generalities

- All sources are written in F90 and C
- Uses the Model Coupling Toolkit (MCT) from Argonne National Lab
- Open source product distributed under a LGPL license
- All external libraries used are public domain (MPI, NetCDF) or open source (LANL SCRIP, MCT)
- Current developers are:
  - 1 permanent FTE (CERFACS, CNRS)
  - 2 consultants : Anthony Craig (also CPL7 and ESMF), Andrea Piacentini

ESiWACE H2020 EU Centre of Excellence

- ➢ ESiWACE1 (2015-2019): 18 pm
- ➢ ESiWACE2 (2019-2022): 16 pms

IS-ENES EU FP7 project

- ➢ IS-ENES2 (2014-2017): 27 pm
- ➢ IS-ENES3 (2019-2022): 35 pms

IS-ENES3 and ESiWACE2 have received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824084 & No 823988

Outline

- Introduction
- Global performance of a coupled system
- Different technical coupling solutions
- OASIS historical overview, generalities, community
- Use of OASIS3-MCT:
  - code interfacing
  - coupled model configuration
- OASIS3-MCT parallel communication
- Interpolations et transformations
- Performances
- Conclusions and perspectives

Outline

- Introduction
- Global performance of a coupled system
- Different technical coupling solutions
- OASIS historical overview, generalities, community
- Use of OASIS3-MCT: *** this on-line course ***
  - code interfacing
  - coupled model configuration
- OASIS3-MCT parallel communication
- Interpolations et transformations
- Performances
- Conclusions and perspectives

Use of OASIS3-MCT

At run time, OASIS3-MCT acts as a communication library linked to the models.

To use OASIS3-MCT:

- Follow the on-line course “Code coupling with OASIS3-MCT” !
- Identify your component models, grids, coupling fields to be exchanged
- Identify the transformations to go from the source to the target grids
- Use the “spoc/spoc_regridding” environment (offline) to test their quality
- Adapt your codes i.e. insert calls to OASIS3-MCT communication library
- Choose the other parameters (source and target, frequency, transformations, etc.) and create the *namcouple* configuration file
- Compile OASIS3-MCT, your components **with same compiler**, and link the components models with OASIS3-MCT library
- Start the models and let OASIS3-MCT manage the coupling exchanges

OASIS3-MCT: code interfacing

- Initialization: `call oasis_init_comp(...)`
- Local partition definition: `call oasis_def_partition (...)`
- Grid definition: `call oasis_write_grid (...)`
- Coupling field declaration: `call oasis_def_var (...)`
- End of definition phase: `call oasis_enddef (…)`
- **Coupling field exchange**:
  - in model time stepping loop

```
call oasis_put (…, date, var_array. …)
call oasis_get (…, date, var_array, …)
```

  - user defines externally the source or target
  - sending or receiving at appropriate time only
  - automatic averaging/accumulation if requested
  - automatic writing of coupling restart file at end of run
- Termination: `call oasis_terminate (…)`

OASIS3-MCT coupled model configuration

Configuration in a **text** file *namcouple*

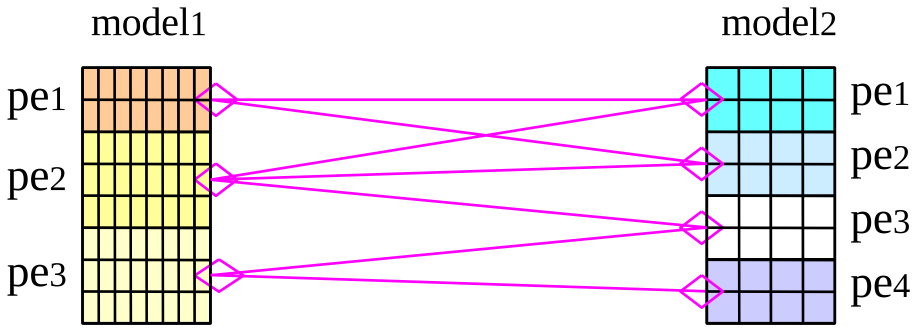
- general characteristics of a coupled run
    - total duration
    - debug level
    - …
- for each exchange of coupling field :
    - source and target symbolic name (end-point communication)
    - exchange period
    - transformations/interpolations

Outline

- Introduction
- Global performance of a coupled system
- Different technical coupling solutions
- OASIS historical overview, generalities, community
- Use of OASIS3-MCT:
  - code interfacing
  - coupled model configuration
- **OASIS3-MCT parallel communication**
- Interpolations et transformations
- Performances
- Conclusions and perspectives

OASIS3-MCT parallel communication

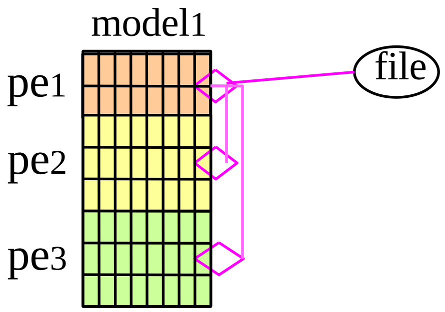
- Fully parallel communication between parallel models based on Message Passing Interface (MPI)



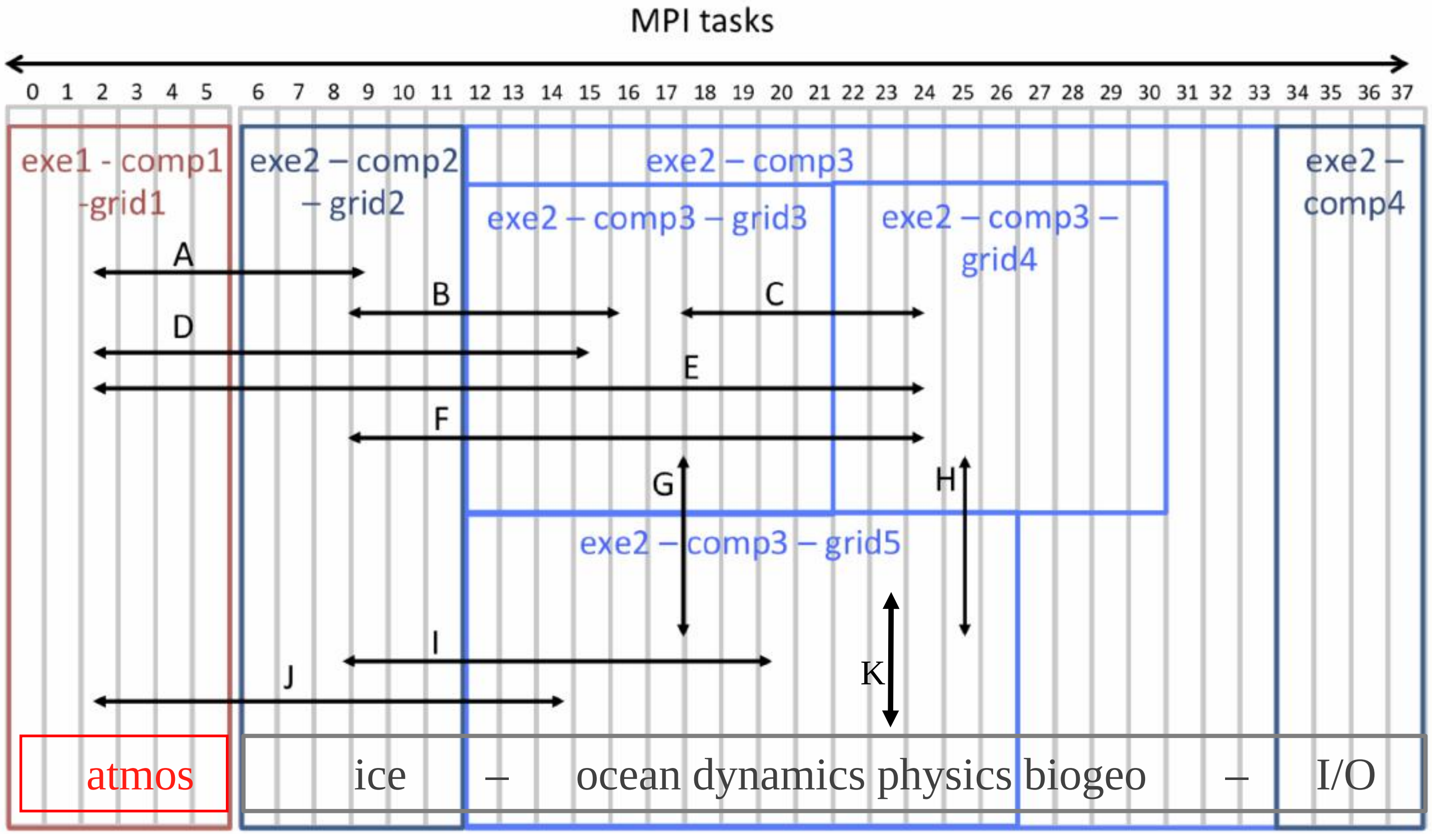
If required, the interpolation weights and addresses are calculated in parallel by the SCRIP library (spoc_regridding)

Interpolation per se from the source grid to the target grid is done in parallel on the source or on the target processes

- I/O functionality (switch between coupled and forced mode):

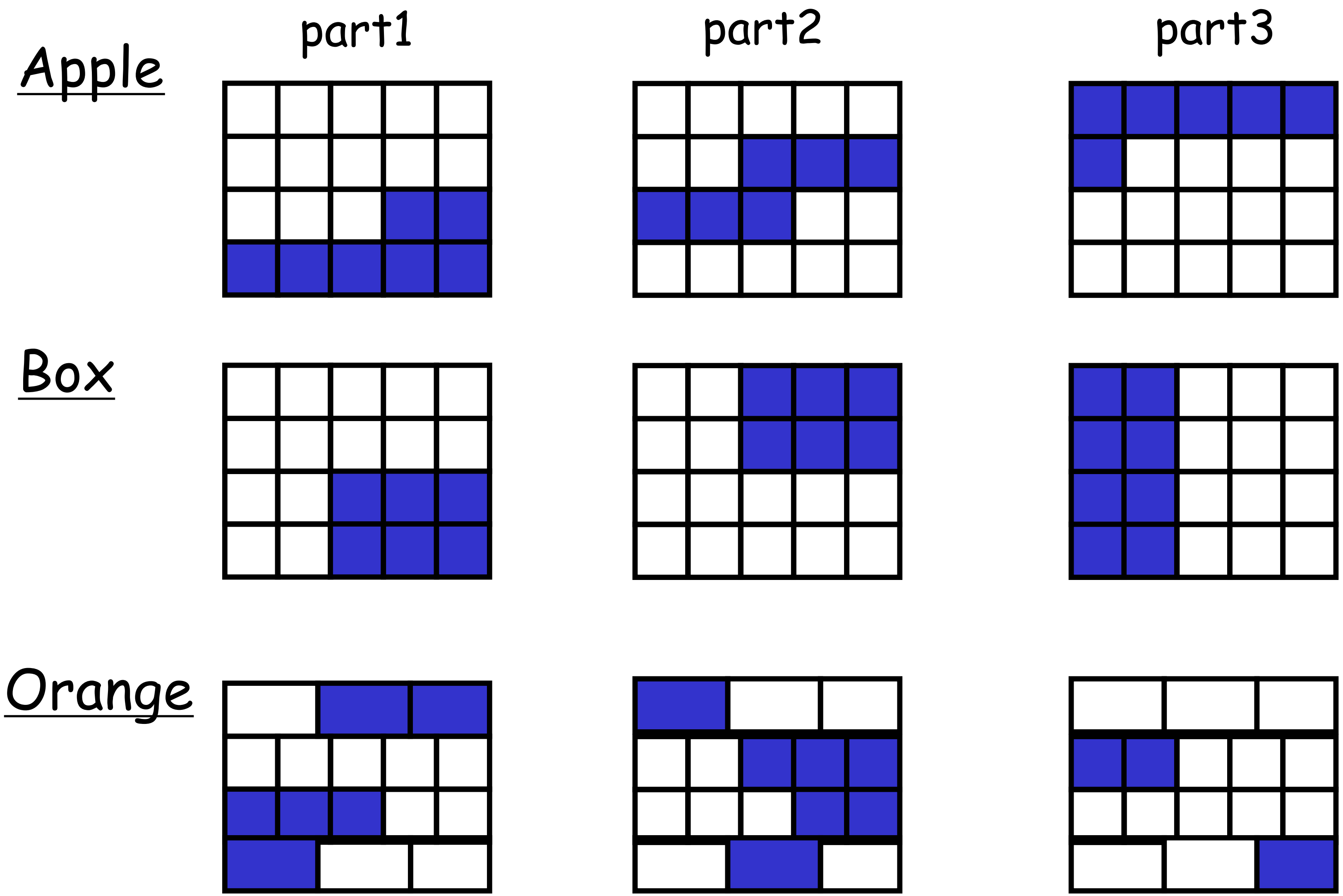


OASIS3-MCT parallel communication: layouts supported



- System has 2 executables; exe1 (atm) and exe2 (ocn_ice)
- Executable 1 has 1 component and 1 grid (atm)
- Executable 2 has 3 components; comp2 (ice), comp3 (ocn), and comp4 (io)
  - comp2 has 1 grid; grid2 (ice) on all comp2 processes
  - comp3 has 3 grids (ocn_phy, ocn_dyn, ocn_bio); on varying processes
- Prior to OASIS3-MCT_3.0, only coupling “A” was supported
- Now supports many coupling layouts

OASIS3-MCT parallel communication



Partitioning supported;
Apple and Orange applicable to unstructured grids

Outline

- Introduction
- Global performance of a coupled system
- Different technical coupling solutions
- OASIS historical overview, generalities, community
- Use of OASIS3-MCT:
  - code interfacing
  - coupled model configuration
- OASIS3-MCT parallel communication
- **Interpolations et transformations**
- Performances
- Conclusions and perspectives

OASIS3-MCT: interpolations & transformations

- on 2D or 3D scalar fields , bundles now supported
- on different types of grids: lat-lon, rotated (logically rectangular), gaussian reduced, unstructured

❖ Transformations: statistics, addition/multiplication by scalar, global conservation

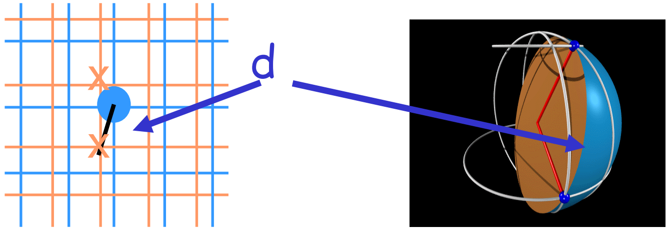
❖ Interpolations/regridding

SCRIP (Jones, 1999)

x source grid point
● target grid point

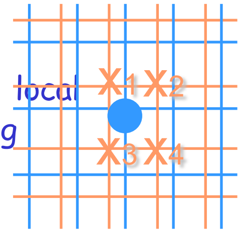
n-nearest-(gaussian-weighted)-neighbours: weight(x) α 1/d

d: great circle distance on the sphere:



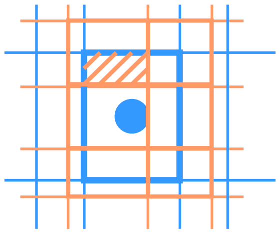
bilinear interpolation

- general bilinear iteration in a continuous local coordinate system using f(x) at x_1, x_2, x_3, x_4



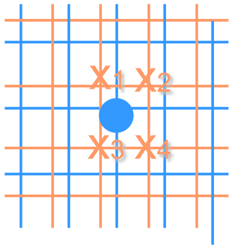
conservative remapping

- weight of a source cell % to intersected area

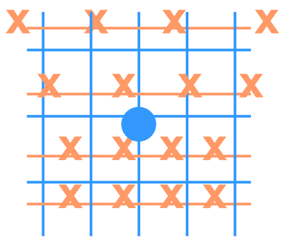


bicubic interpolation:

- general bicubic iterations in a continuous local coordinate system: f(x), $\delta f(x)/\delta i$, $\delta f(x)/\delta j$, $\delta^2 f/\delta i \delta j$ in x_1, x_2, x_3, x_4 for logically-rectangular grids (i,j)



- standard bicubic algorithm: 16 neighbour points for Gaussian Reduced grids

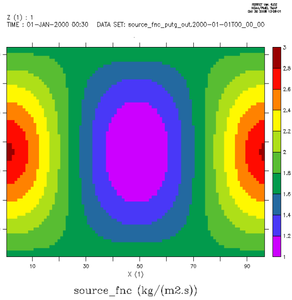


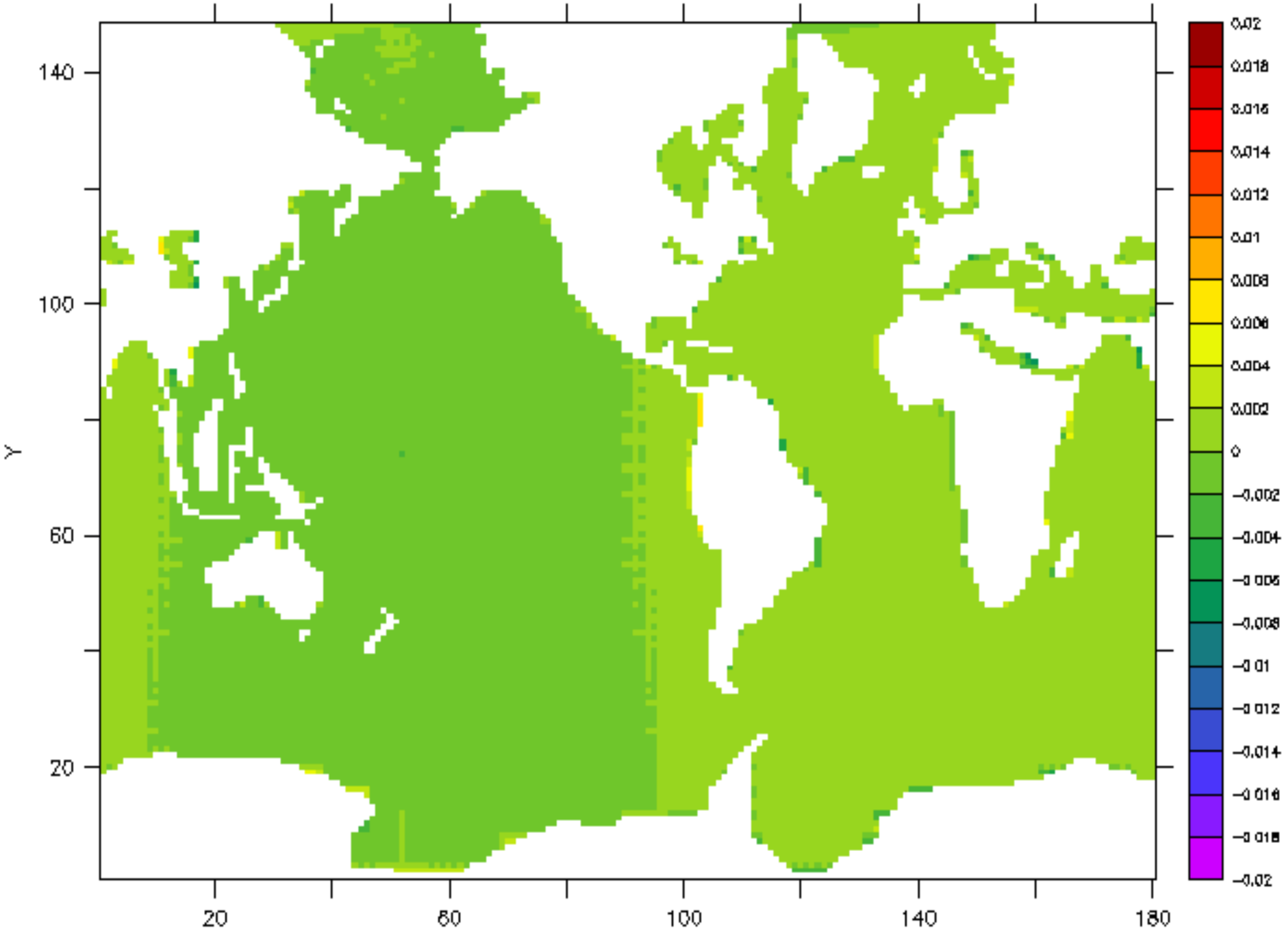
** gradients must be given as extra arguments to the oasis_put

OASIS3-MCT: interpolations & transformations

One example of bilinear interpolation error

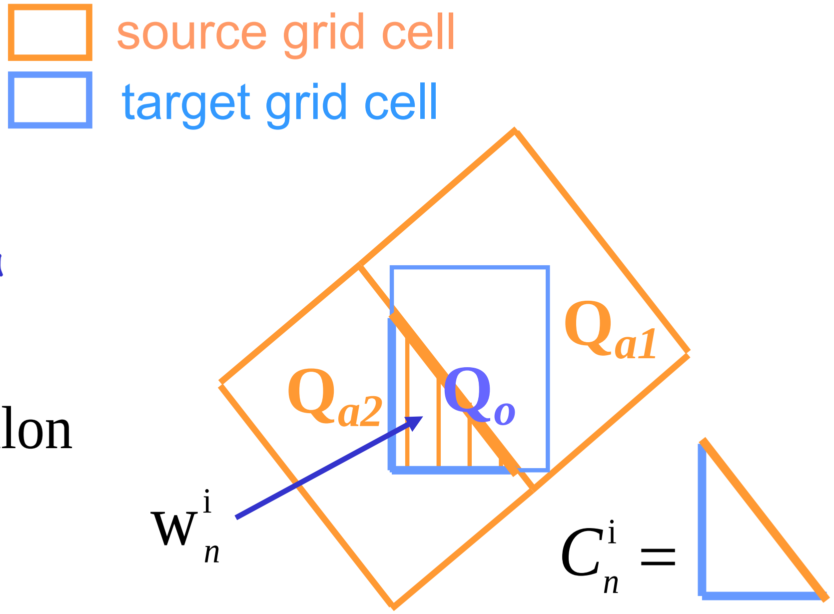
$F = 2 + \cos[\pi * \mathrm{acos}(\cos(lon)\cos(lat))]$ LMDz grid (96 x 72) -> ORCA2





- < 0.2% whole domain; ~1% near the coastline

OASIS3-MCT: interpolations & transformations



1st order conservative regridding:

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

$$Q_o^i = \frac{1}{A_o}\sum_{n=1}^{N} Q_{a_n} w_n^i \quad \text{with} \quad w_n^i = \oint_{C_n^i} -\sin(\mathrm{lat})\,\mathrm{dlon}$$

Actual limitations:

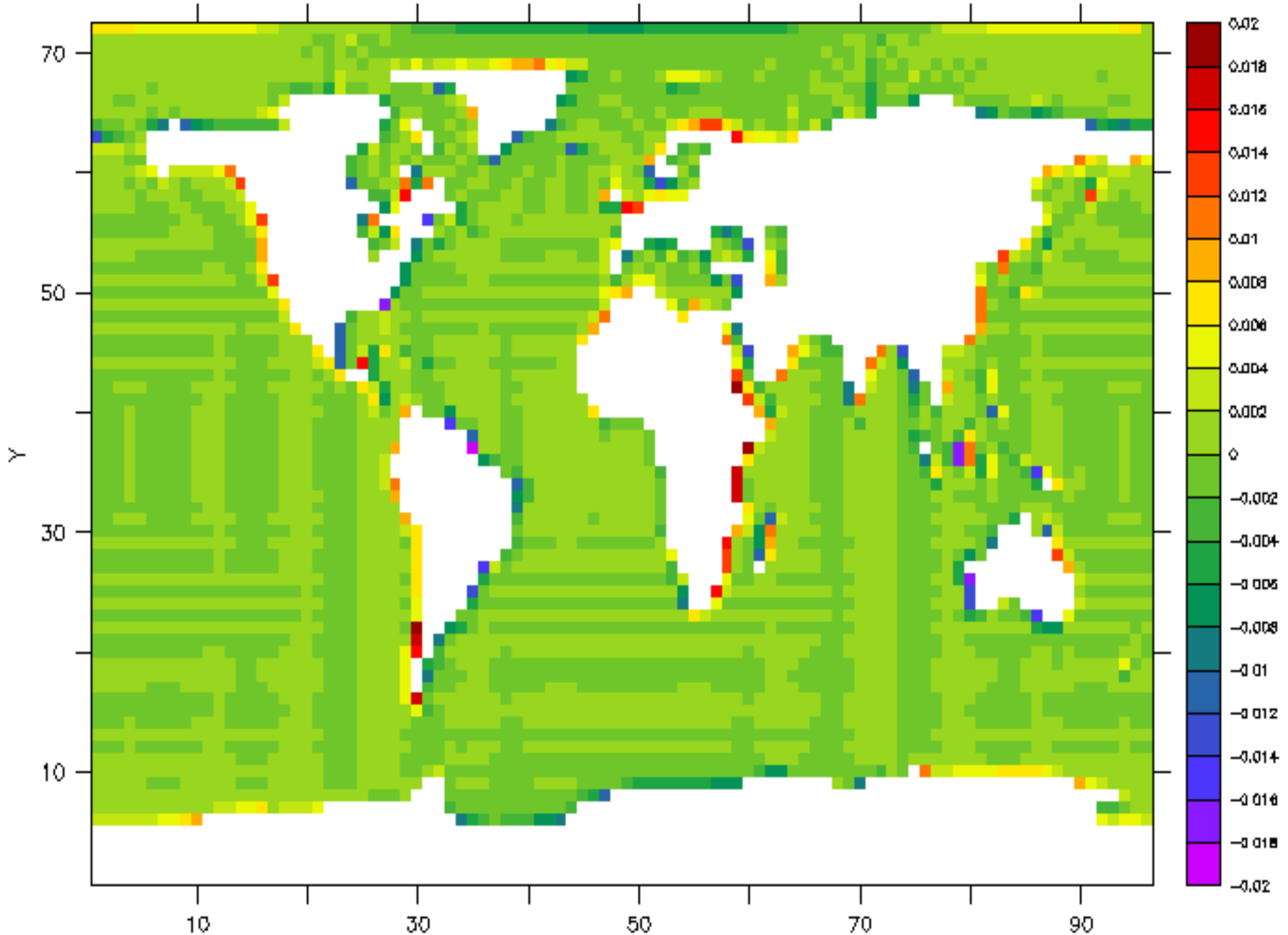
- assumes borders are linear in (lat,lon) ; uses Lambert equivalent azimuthal projection near the pole for intersection calculation
- assumes sin(lat) linear function of lon for line integral calculation
    - need to use a projection near the pole (as done for intersect. calc.)
- exact calculation is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

- conservative regridding from other libraries under evaluation (ESMF, XIOS, YAC)

OASIS3-MCT: interpolations & transformations

- One example of conservative remapping error

F = 2 – cos[π*acos(cos(lon)cos(lat)]
ORCA2 -> LMDz (96x72)



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

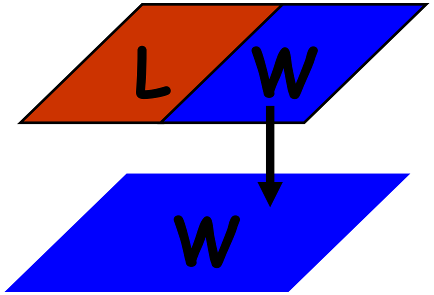
OASIS3-MCT: interpolations & transformations

Problem with non-matching sea-land masks

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

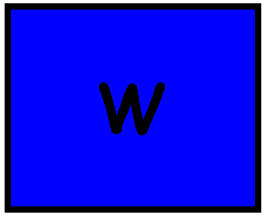
1- Support subsurfaces in the atmosphere

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



2-“DESTAREA” option

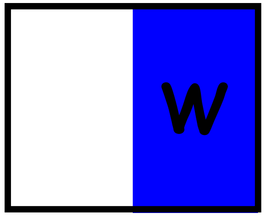
$A_o =$



- ➢ local flux conservation
- ➢ possibly unrealistic flux values

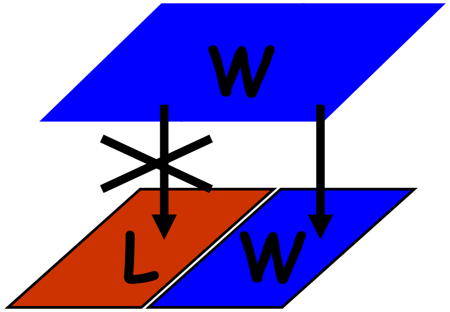
3-“FRACAREA” option

$A_o =$



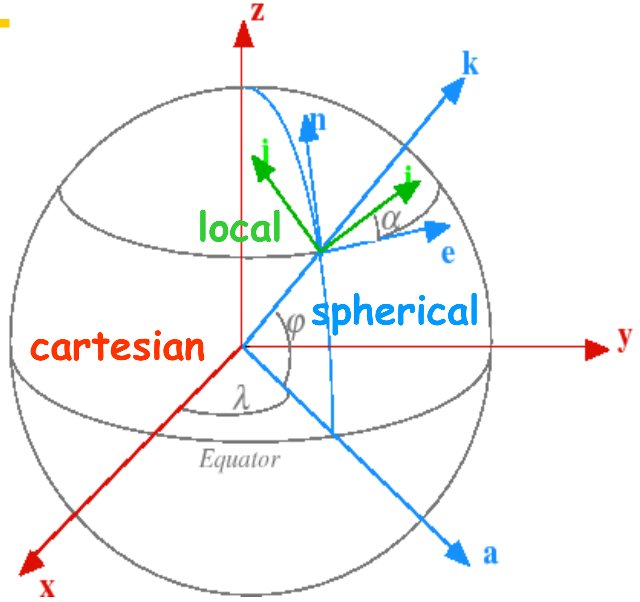
- ➢ no local conservation of flux
- ➢ realistic flux values

\+ nearest non-masked value for ocean cells covered only with masked atmospheric cells (FRACNNEI)



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

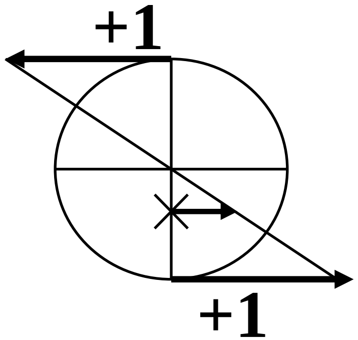
OASIS3-MCT: interpolations & transformations



Vector interpolation (winds, currents, …)

- interpolation of vectors component per component is not accurate, 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. -> zonal wind of 1.

Solution (proposed by O. Marti, LSCE):

- “turn” the vector in the spherical ref. and project the resulting vector in a cartesian ref
- send the 3 components in the cartesian referential and let OASIS3-MCT interpolate them
- project back in spherical referential; check that k component is zero
- possibly “turn” the resulting vector in the target local referential

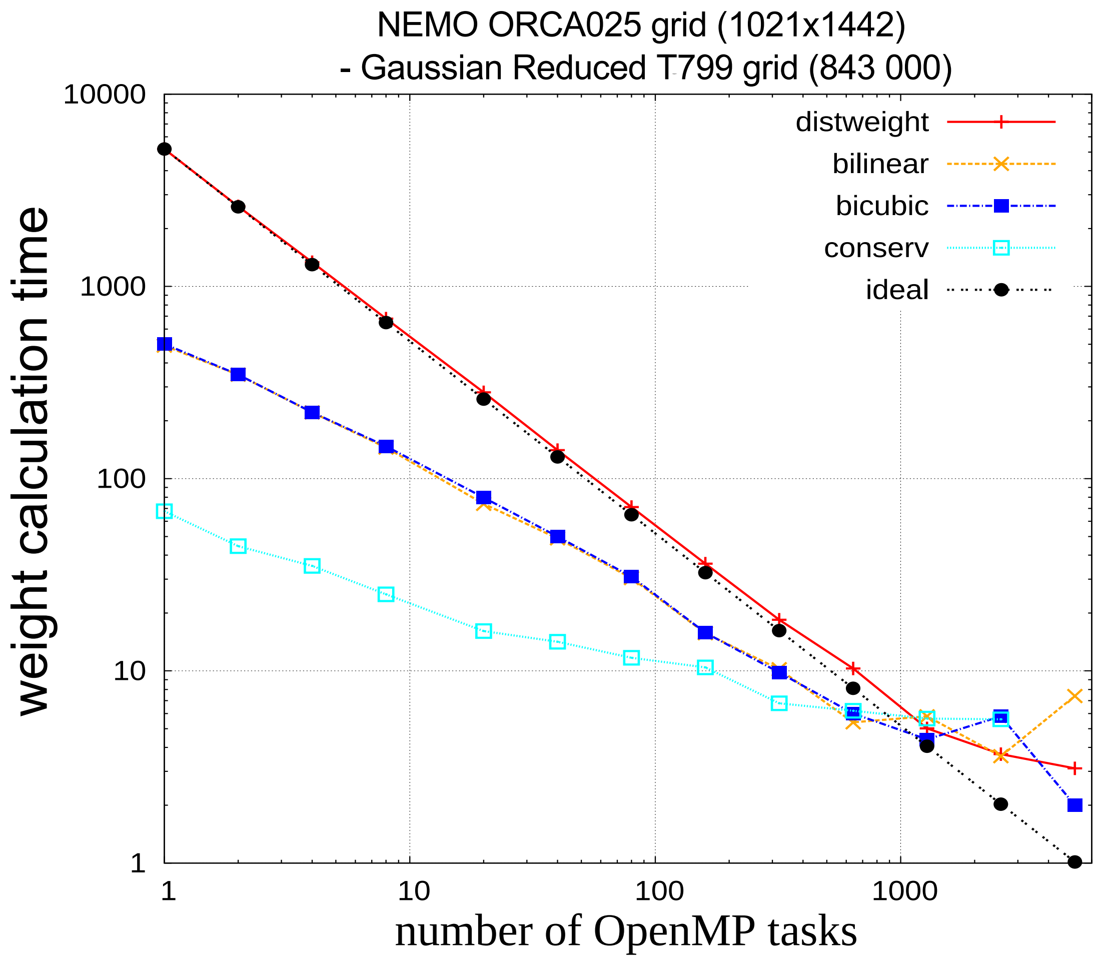
Outline

- Introduction
- Global performance of a coupled system
- Different technical coupling solutions
- OASIS historical overview, generalities, community
- Use of OASIS3-MCT:
  - code interfacing
  - coupled model configuration
- OASIS3-MCT parallel communication
- Interpolations et transformations
- **Performances**
- Conclusions and perspectives

OASIS3-MCT_4.0 performance

Hybrid MPI/OpenMP parallelisation of the SCRIP library

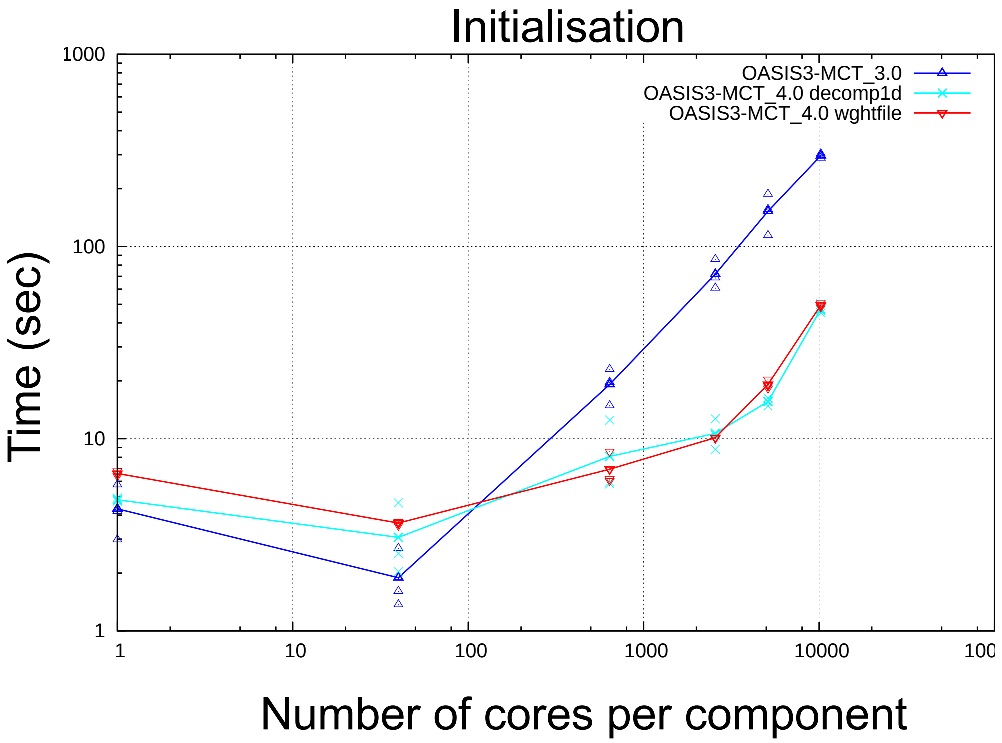
SCRIP library hybrid MPI+OpenMP parallelisation

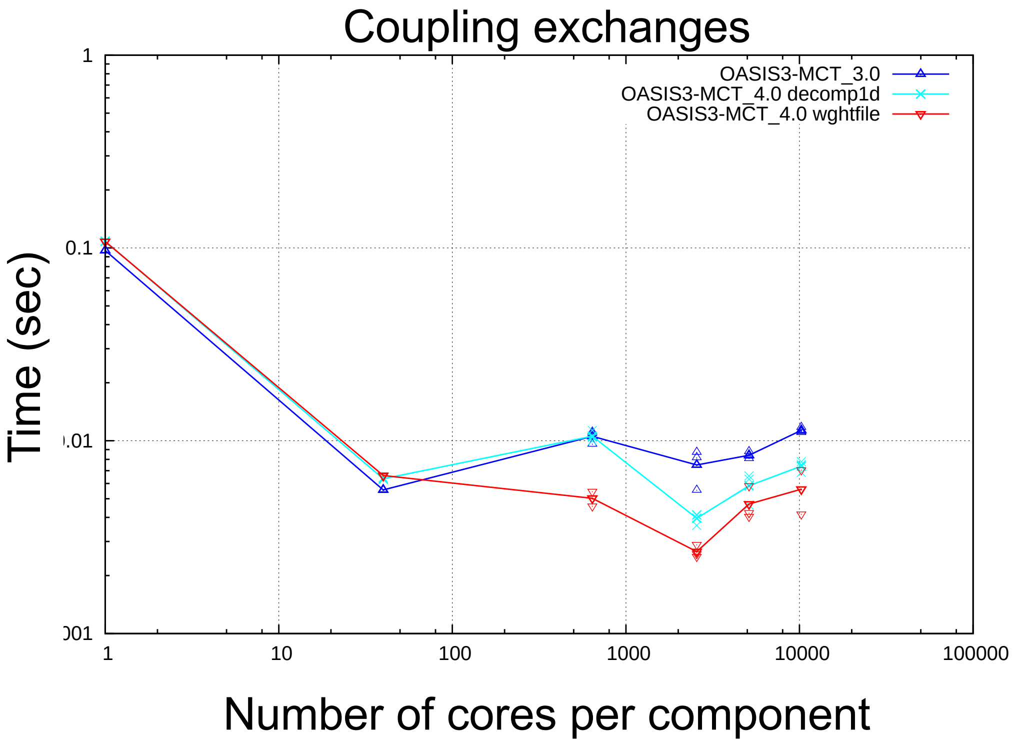


⇒ Reduction of O(100)-(1000) for weight calculation

OASIS3-MCT_4.0 performance

- Toy coupled model: ping-pong exchanges between NEMO ORCA025 grid (1021x1442) and Gaussian Reduced T799 grid (843 000)
- Bullx beaufix, Intel 16.1.150 compiler, Intelmpi 5.1.2.150 MPI library





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

- ~1 min for initialisation, ~20 secs for data exchange

OASIS3-MCT most likely provides a satisfactory solution for fully parallel coupling in our climate models at the resolutions targeted operationally for the next ~5 years.

Conclusions and perspectives

Conclusions

- OASIS3-MCT shows good parallel performance
- OASIS3-MCT offers greater flexibility to couple components, e.g.:
  - in a single executable
  - across overlapping, non-overlapping, or partly overlapping processes,
  - within a single component.

-> OASIS3-MCT most likely provides a satisfactory solution for fully parallel coupling in our climate models at the resolutions targeted operationally for the next ~5 years.

Perspectives:

- Evaluation of ESMF,XIOS, YAC for off-line precomputing of interpolation weights (on going)
- Many other more minor developments in the framework of IS-ENES3 and ESiWACE2
- Support of grids with evolving masks under consideration, a first step toward dynamic coupling, i.e. grids evolving in time !!??

The end