

The OASIS coupler: history, community and current status



NEC



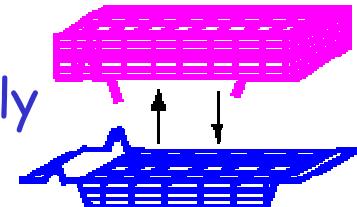
- Code coupling in climate modelling
- Technical solutions for code coupling
- OASIS: historical background
- OASIS: the community today
- OASIS3 and OASIS4
 - Model adaptation
 - Coupled model configuration
 - Communication
 - Transformations/interpolations



Code coupling in climate modelling

Why couple ocean and atmosphere (and sea-ice and land and ...) 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 (sequential vs concurrent)
- ✓ Start from existing and independently developed codes
- ✓ Global performance and load balancing issues are crucial
- ✓ Platform characteristics (CPU size, message passing efficiency, ...)
- ✓ Interaction with the Operating System must be considered

Technical solutions for code coupling (1/3)

1. merge the codes:

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



```
program prog2  
subroutine sub_prog2(data)  
...  
end prog2
```

- (:(
- (:(
- :)
- :)
- (:(
- (:(
- easy
- flexible
- efficient
- portable
- existing codes
- no use of generic transformations

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

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

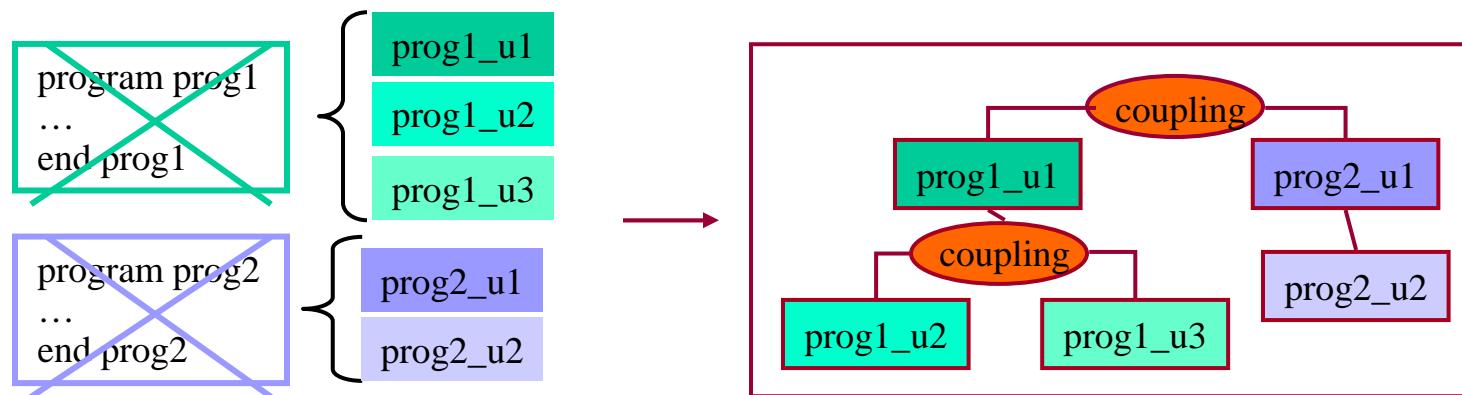
```
program prog2  
...  
call xxx_recv (prog1, data)  
end
```

- (:(
- (:(
- (:(
- :)
- :)
- :)
- easy
- flexible
- no use of generic transformations
- (efficient)
- (portable)
- existing codes

Technical solutions for code coupling (2/3)

3. use coupling framework/library (ESMF, FMS, ...)

- 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



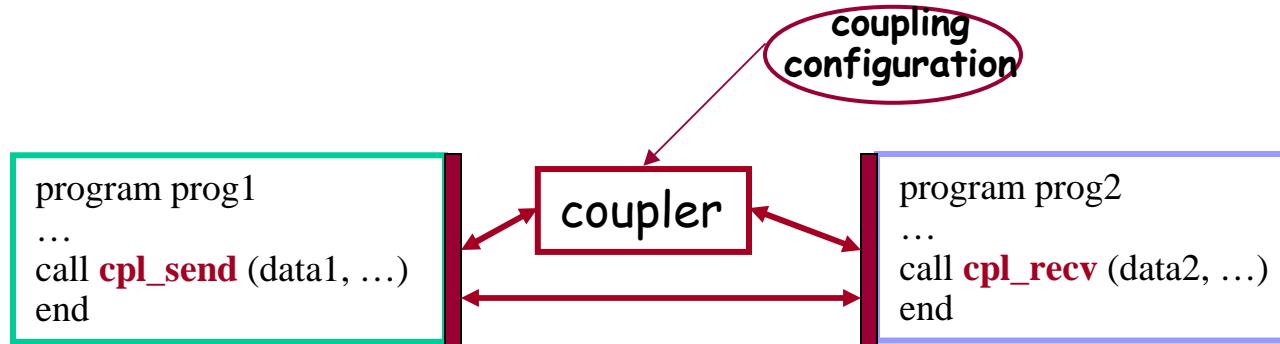
- 😊 flexible
- 😊 efficient
- 😊 portable
- 😊 use of generic utilities (parallelisation, regridding, time management, etc.)
- 😊 sequential and concurrent components

- 😢 existing codes
- 😢 (easy)

→ best solution in a controlled development environment

Technical solutions for code coupling (3/3)

4. use a coupler (OASIS, PALM, MPCCI ...)



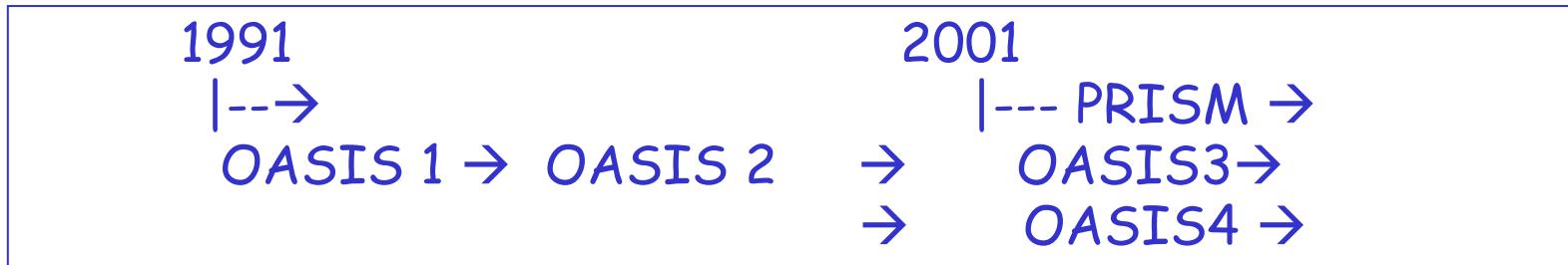
- 😊 existing codes
- 😊 flexible
- 😊 portable
- 😊 use of generic transformations/regridding
- 😊 concurrent coupling

- 😢 sequential coupling
- 😐 (efficient)

→ best solution to couple independently developed codes

OASIS: historical background

- ❖ developed by CERFACS since 1991 to couple existing GCMs
- ❖ since 2001, collaboration between NLE-IT, CNRS and CERFACS



OASIS1, OASIS2, OASIS3:

- low resolution, low number of 2D fields, low coupling frequency:
 - **flexibility** very important, efficiency not so much!
 - ❖ New OASIS3_3 official release very soon!

OASIS4:

- high resolution parallel models, massively parallel platforms, 3D fields
 - need to **optimise** and **parallelise** the coupler
 - ❖ OASIS4 beta version available
- F90 & C, open source (LGPL), public domain libraries (MPI, NetCDF, libXML, GFDL mpp_io, LANL SCRIP)

OASIS: the community today

• CERFACS (France)	ARPEGE5-ORCA1, ARPEGE4-NEMO/LIM-TRIP, ...
• METEO-FRANCE (France)	ARPEGE4-ORCA2, ARPEGE3-OPAmed ARPEGE3-OPA8.1/GELATO
• IPSL- LODYC, LMD, LSCE (France)	LMDz-ORCA2/LIM LMDz-ORCA4 ORCA4
• MPI-M&D (Germany)	ECHAM5-MPIOM, ECHAM5-CHOPE, PUMA-CHOPE, EMAD-EHOPE
• ECMWF	IFS - CTM (GEMS), IFS - ORCA2 (MERSEA), EC-Earth
• MetOffice (UK)	MetOffice ATM - NEMO
• IFM-GEOMAR (Germany)	ECHAM5 - NEMO (OPA9-LIM)
• NCAS / U. Reading (UK)	ECHAM4 - ORCA2 HADAM3-ORCA2
• SMHI (Sweden)	RCA(region.) - RCO(region.)
• NERSC (Norway)	ARPEGE - MICOM, CAM - MICOM
• KNMI (Netherlands)	ECHAM5 - TM5/MPI-OM
• INGV (Italy)	ECHAM5 - MPI-OM
• ENEA (Italy)	MITgcm - REGgcm
• JAMSTEC (Japan)	ECHAM5(T106) - ORCA $\frac{1}{2}$ deg
• IAP-CAS (China)	AGCM - LSM
• KMA (Korea)	CAM3 - MOM4
• BMRC (Australia)	BAM3-MOM2, BAM5-MOM2, TCLAPS-MOM
• CSIRO (Australia)	Sea Ice code - MOM4
• RPN-Environment Canada (Canada)	MEC - GOM
• UQAM (Canada)	GEM - RCO
• U. Mississippi (USA)	MM5 - HYCOM
• IRI (USA)	ECHAM5 - MOM3
• JPL (USA)	UCLA-QTCM - Trident-Ind4-Atlantic

Use of OASIS3 and OASIS4

To use OASIS3 or OASIS4:

- Identify your component models
- Identify the coupling fields to be exchanged between those models
- Adapt your model i.e. insert calls to OASIS communication library (PSMILe)
- Choose the coupling parameters (source and target, frequency, fields transformations, etc.) and write the configuration file
- Compile OASIS and the components models linked with PSMILe
- Start OASIS and the models and let it manage the coupling exchanges

OASIS3 & OASIS4: model code adaptation

PRISM System Model Interface Library (PSMILe) API :

	OASIS3	OASIS4
Initialization	<code>call prism_init_comp_proto(...)</code>	<code>call prism_init_comp(...)</code>
Grid definition	<code>call prism_write_grid(...)</code> ➤ optional: global grid definition by master process only	<code>call prism_def_grid(...)</code> <code>call prism_set_corners(...)</code> <code>call prism_set_points(...)</code> ➤ mandatory: local grid definition by each process
Partition definition	<code>call prism_def_partition_proto(...)</code>	<code>call prism_def_partition(...)</code>
Coupling field declaration:	<code>call prism_def_var_proto (...)</code>	<code>call prism_def_var (...)</code>

OASIS3 & OASIS4: model code adaptation

PRISM System Model Interface Library (PSMILe) API :

	OASIS3	OASIS4
Coupling field exchange	<code>call prism_put_proto (..., time, var_array, ...)</code> <code>call prism_get_proto (..., time, var_array, ...)</code>	<code>call prism_put (..., date, date_bounds, var_array, ...)</code> <code>call prism_get (..., date, date_bounds, var_array, ...)</code>

OASIS3 & OASIS4: coupled model configuration

- For OASIS3: in a **text** file *namcouple*:
- For OASIS4: in **XML** (Extensive Markup Language) files:
 - total run time
 - component models
 - for each exchange of coupling field :
 - source and target (end-point communication)
 - coupling or I/O period
 - transformations/interpolations

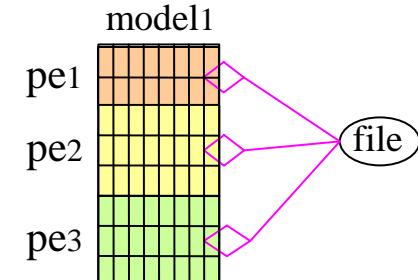
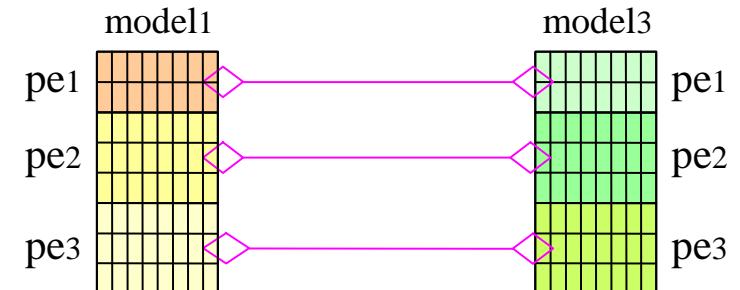
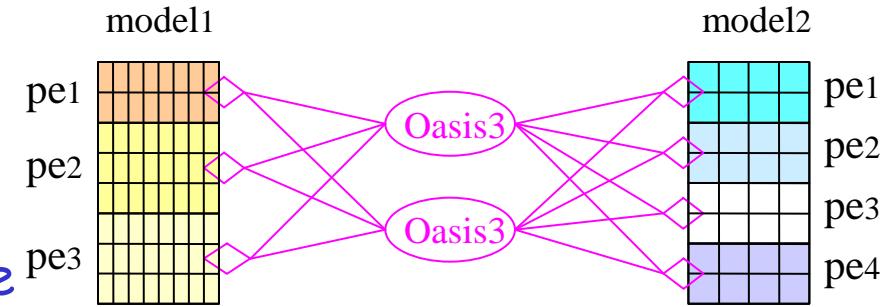
OASIS3: communication

PSMILe based on MPI1 or MPI2 message passing

- Parallel communication between parallel models and interpolation executable(s): each executable treats a subset of the coupling fields described in its own namcouple configuration file.

- Direct communication between models with same grid and partitioning

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

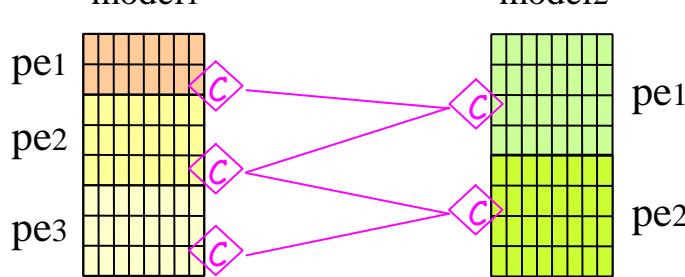


OASIS4 communication

Model interface library: PSMILe based on MPI1 or MPI2

- Parallel communication based on repartitioning and/or regridding needs:

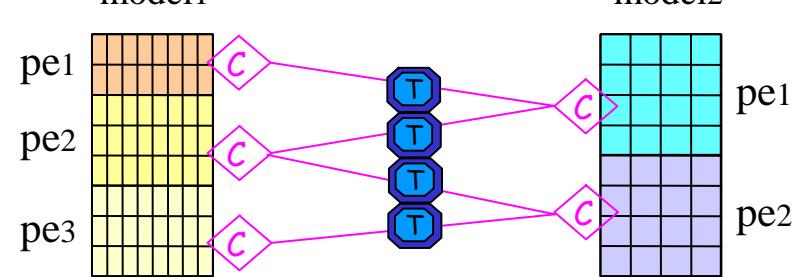
➤ parallel communication patterns based on geographical description of the partitions



Same grid & different decomposition or
gridless fields

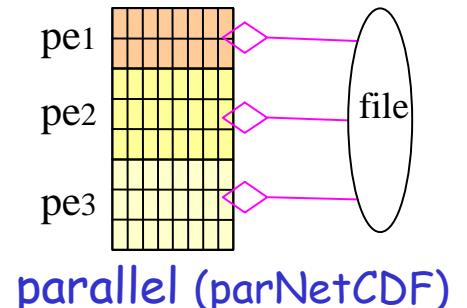
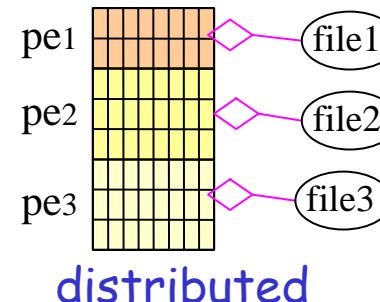
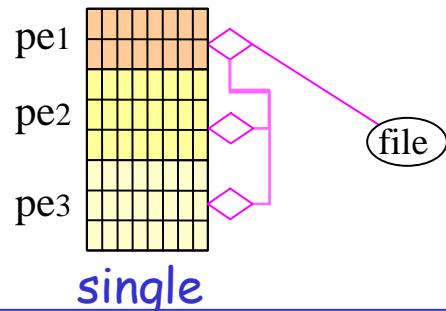
➤ direct repartitioning

➤ one-to-one, one-to-many; extraction of useful part of source field only



Different grid and decomposition
➤ interpolation in parallel Transformer

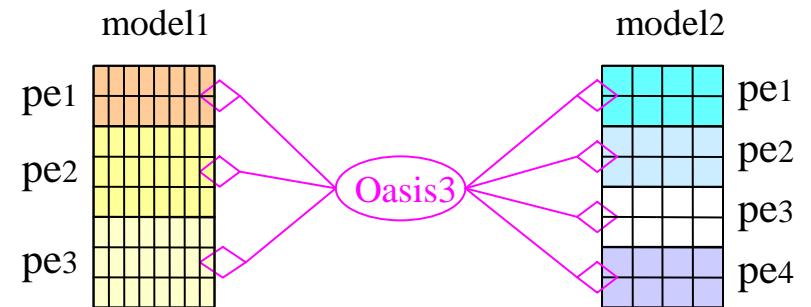
- Parallel I/O : GFDL mpp_io + parNetCDF



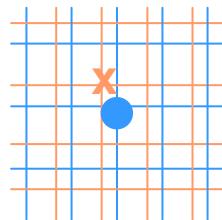
OASIS3: interpolations & transformations

➤ separate sequential executable 

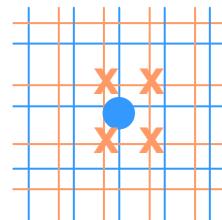
- ✓ neighbourhood search
- ✓ weight calculation
- ✓ interpolation per se during the run



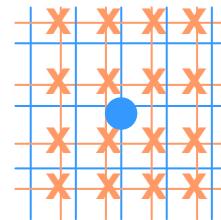
- SCRIP 1.4 library, RPN Fast Scalar INTerpolator:



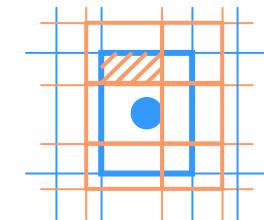
nearest-neighbour
interpolation



bilinear
interpolation



bicubic
interpolation

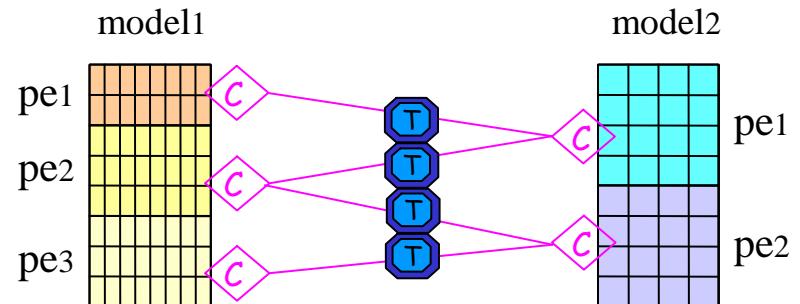


conservative
remapping

- Statistics, addition/multiplication by scalar
- Flux correction, combinations, global conservation, subgrid variability, user-defined regridding, etc.
- on 2D scalar or vector fields
- on different types of grids: lat-lon, stretched or rotated (logically rectangular), gaussian reduced, unstructured

OASIS4 interpolations & transformations

- global neighbourhood search done in parallel in source PSMILe
- multigrid algorithm
- weight calculation and interpolation done in parallel in Transformer T
 - statistics, addition/multiplication by scalar
 - gathering/scattering
 - interpolation/regridding:
 - 2D nearest-neighbour, bilinear, bicubic, conservative remapping
 - 3D nearest-neighbour, trilinear, tricubic
- currently, parallel global neighbourhood search implemented for all types of logically-rectangular and Gaussian-reduced grids all interpolations except 2D conservative remapping (under active work)



OASIS: developments & perspectives

- IS-ENES EU project (2009-12): 58 pm for CERFACS, 35 pm for DKRZ
 - On-going call for Dedicated User Support (3pm/year)
- OASIS3
 - New pseudo-parallel OASIS3_3 release very soon (beta version available) !
 - Active user support and maintenance but (almost) no new developments
- OASIS4
 - Version available to tester groups:
 - IFM-GEOMAR (Kiel) in pseudo-models to interpolate high-resolution fields.
 - EU project GEMS: atmospheric dynamic and chemistry coupling
 - SMHI: ocean-atmosphere regional coupling
 - CICLE project (ANR-05-CIGC-04) : CERFACS, Meteo-France, IPSL.
 - MPI-M for CMIP5 (see L. Kornblueh's talk)
 - Current and planned developments:
 - Parallel global search 2D conservative remapping
 - Full validation of current transformations
 - User-defined set of weights and addresses
 - Graphical User Interface for XML configuration files
 - Support of unstructured grids (collaboration with AWI - ScalES project)
 - Others, depending on User Meeting recommendations !