CERFACS

EUROPEAN CENTRE FOR RESEARCH AND ADVANCED TRAINING IN SCIENTIFIC COMPUTING

esiwace

CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Parallel programming in practice: Scaling algorithms and **Code Coupling**

Sophie Valcke - Cerfacs

August 24th 2020

Summer School on Effective HPC for Climate and Weather

Objectives

Coupling of codes is an example of course grained task parallelism where each task (code) is itself parallel

- Describe the concepts of code coupling
- Evaluate qualitatively the impact of different coupling configurations (sequential vs concurrent, multi vs mono-executable, …) on coupled model performance
- Classify coupling software implementations given their main characteristics
- Describe the most used coupling software in climate and weather applications

Outline

- Introduction
- Sequential and concurrent coupling
- Global performance of a coupled system
- Different technical solutions to coupling
- Few coupling software used in climate modelling
- Coupling algorithms in selected CGCMs
- Summary

Outline

- Introduction
- Sequential and concurrent coupling
- Global performance of a coupled system
- Different technical solutions to coupling
- Few coupling software used in climate modelling
- Coupling algorithms in selected CGCMs
- Summary

What does coupling of codes mean ?

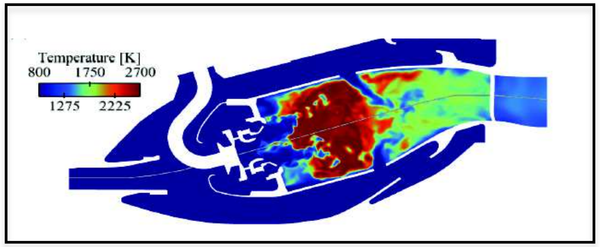
- Exchange and transform information at the code interface
- Manage the execution and synchronization of the codes

Why do we want to couple different codes?

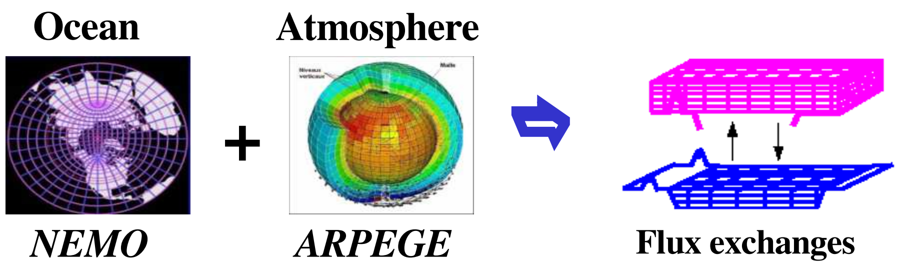
- To model a system globally taking into account interactions between its components

  - Fluid-structure coupling :

    the structure deformation has a direct impact on the fluid flow



  - Ocean-atmosphere coupling for climate modelling



What are the constraints in code coupling?

- ✓ must be **easy to implement** and **portable**
- ✓ must be **flexible** (easy change of component, coupling fields, etc.)
- ✓ very often, starts from **existing** and independently developed **codes**
- ✓ must take into account the **computing platform** and **operating system characteristics and limits**

Introduction

What are the constraints in code coupling?

✓ **Global performance of the coupled system** as a whole must be **good** :

What are the constraints in code coupling?

✓ **Global performance of the coupled system** as a whole must be **good** :

load balancing

- are all computing resources mobilized used all the time?
- is one component model waiting for the other, leading to a waste of resources?

What are the constraints in code coupling?

✓ **Global performance of the coupled system** as a whole must be **good** :

load balancing

- are all computing resources mobilized used all the time?
- is one component model waiting for the other, leading to a waste of resources?

simulation throughput

- how fast do you get the result?
- how many Simulated Years of climate are achieved Per real Day (SYPD)?

What are the constraints in code coupling?

✓ **Global performance of the coupled system** as a whole must be **good** :

<u>load balancing</u>

- are all computing resources mobilized used all the time?
- is one component model waiting for the other, leading to a waste of resources?

<u>simulation throughput</u>

- how fast do you get the result?
- how many Simulated Years of climate are achieved Per real Day (SYPD)?

<u>CPU cost</u>

- how much computing resources do you use to get the result?
- how many Cores for how many Hours to Simulated one Year (CHSY)?

Introduction

What are the constraints in code coupling?

✓ **Global performance of the coupled system** as a whole must be **good** :

load balancing

- are all computing resources mobilized used all the time?
- is one component model waiting for the other, leading to a waste of resources?

simulation throughput

- how fast do you get the result?
- how many Simulated Years of climate are achieved Per real Day (SYPD)?

CPU cost

- how much computing resources do you use to get the result?
- how many Cores for how many Hours to Simulated one Year (CHSY)?

It is usually impossible to define a layout that optimizes at the same time the 3 criteria

Outline

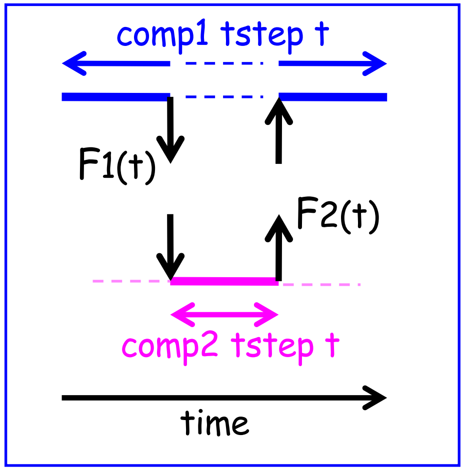
- Introduction
- Sequential and concurrent coupling
- Global performance of a coupled system
- Different technical solutions to coupling
- Few coupling software used in climate modelling
- Coupling algorithms in selected CGCMs
- Summary

Sequential and concurrent coupling

- **Nature is concurrent**, e.g. the ocean and the atmosphere evolve continuously, exchanging momentum, water and heat fluxes continuously.
- Because we solve equations discretized in space and time, **different coupling algorithms** can be implemented by playing with the **lags** of the **different coupling fields**.
- The **performance** of the coupled system will be **impacted** by the implementation of the **coupling algorithm**.
- The **science** (the physics we want to model) will **determine what** coupling algorithm **is acceptable** .
- The coupling algorithm can be **sequential** or **concurrent** (or a mix of both).

Sequential and concurrent coupling

Sequential components

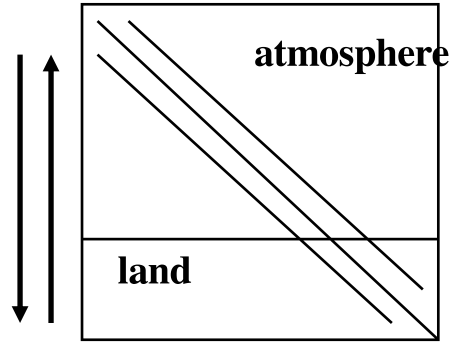


Ex: implicit resolution of heat diffusion equation from the top of the atmosphere model to the bottom of the land or ice model

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

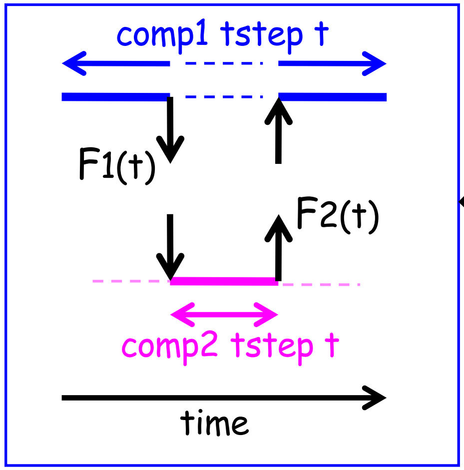
$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = K \frac{T_{k+1}^{n+1} + T_{k-1}^{n+1} + 2T_k^{n+1}}{\Delta z^2}$$

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



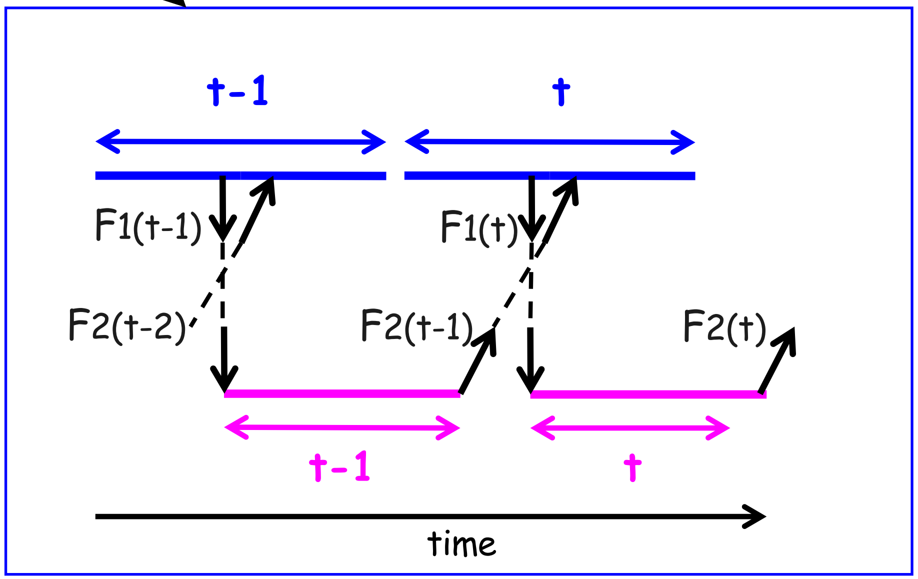
Sequential and concurrent coupling

Sequential components



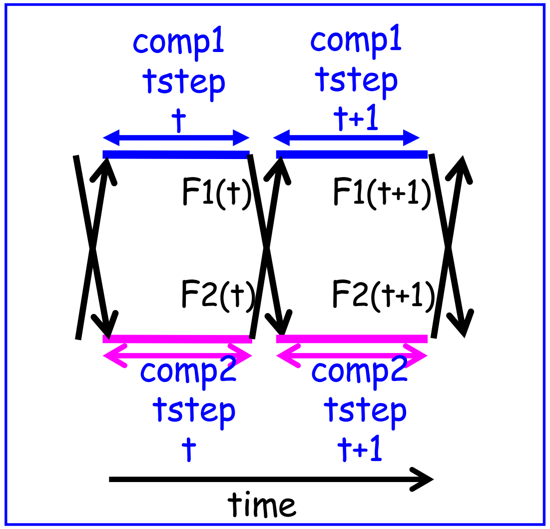
To force the parallelism,
sequential components can be run concurrently
by using the coupling fields
produced at the previous coupling time

… run concurrently

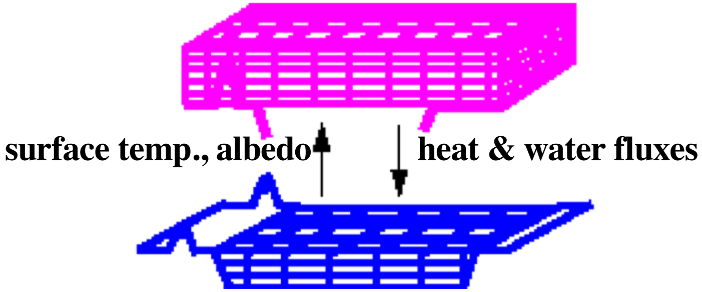


Sequential and concurrent coupling

Concurrent components

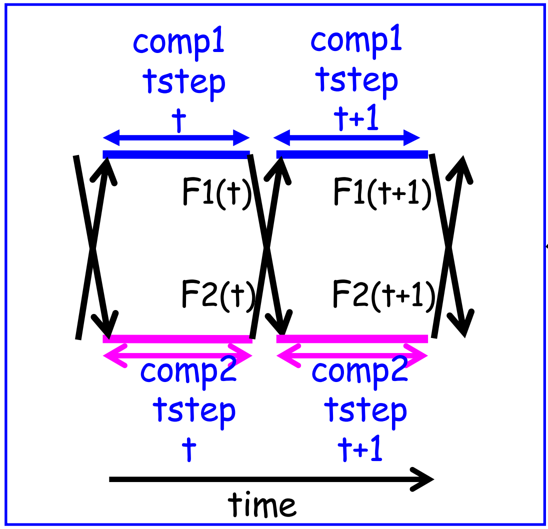


Ex: traditional asynchronous ocean-atmosphere coupling



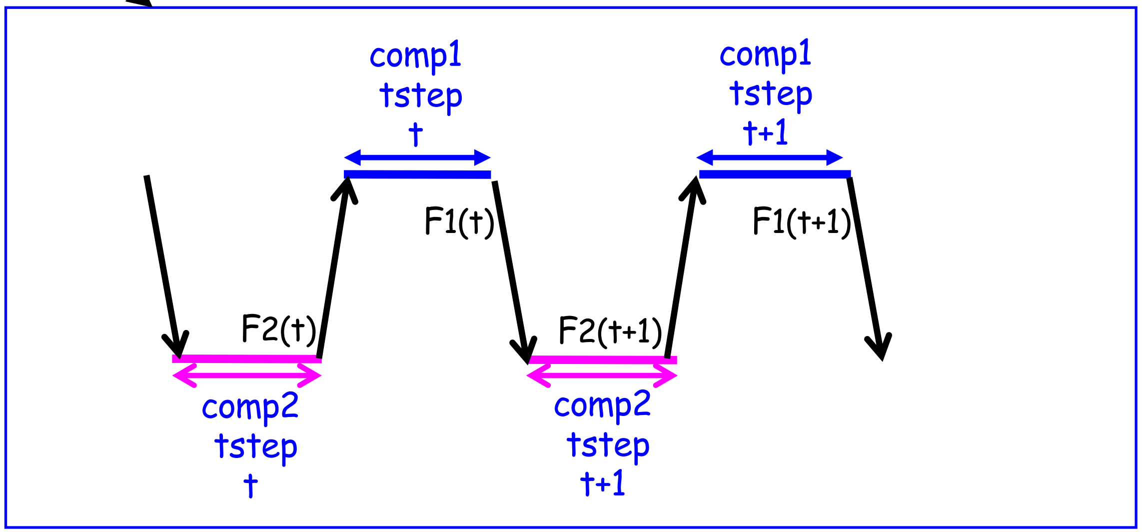
Sequential and concurrent coupling

Concurrent components



Concurrent components can be run sequentially by using for one component the coupling fields produced by the other at the same time

… run sequentially



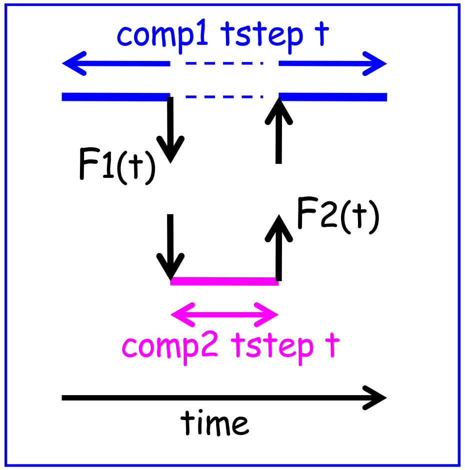
Outline

- Introduction
- Sequential and concurrent coupling
- Global performance of a coupled system
- Different technical solutions to coupling
- Few coupling software used in climate modelling
- Coupling algorithms in selected CGCMs
- Summary

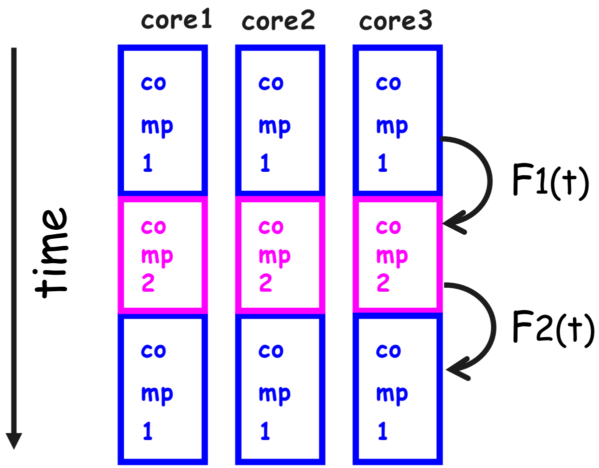
Global performance of a coupled system

Sequential components

sequential coupling



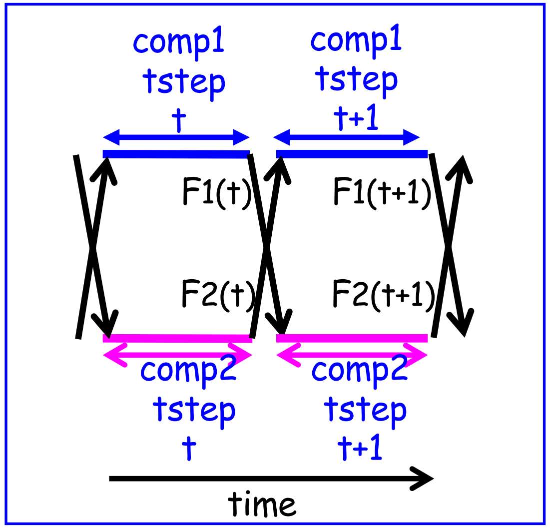
=> sequential execution of components, one after the other on the same set of cores



- ☺ Automatic load balancing: all resources used all the time
- ☹ Not optimal for CPU cost: components most probably not both run at their optimal scaling point
- ☹ Not optimal for throughput: no component parallelism
- ☹ Possible conflicts as components merged in one executable (I/O, units, int. comm, etc.)
- ☺ Efficient coupling exchanges through the memory
- ☹ No flexibility in coupling algorithm

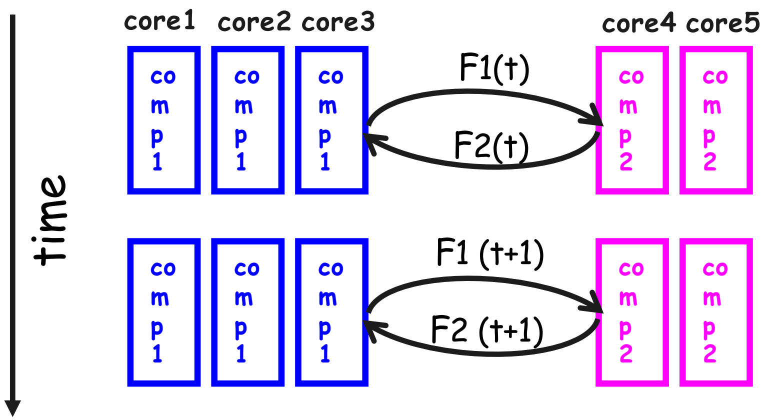
Global performance of a coupled system

Concurrent components



concurrent coupling

=> concurrent execution on different sets of cores within one or multiple executables



- ☺ Good for throughput: component parallelism is activated
- ☹ Load balanced configuration most probably does not lead to optimal CPU cost as components will most probably not both run at their optimal scaling point
- ☺ No conflicts if components are run as separate executables (I/O, units, internal comm, etc.)
- ☺ Flexible coupling algorithm (exchanges possible within the timestep)
- ☹ Less efficient coupling exchanges (no shared memory), through MPI or other protocol

Outline

- Introduction
- Sequential and concurrent coupling
- Global performance of a coupled system
- **Different technical solutions to coupling**
- Few coupling software used in climate modelling
- Coupling algorithms in selected CGCMs
- Summary

Different technical solutions to coupling

1. merging the codes

IFS NEMO coupling at ECMWF

```
program prog1
…
call sub_prog2(data_in, data_out, …)
…
end prog1
```

```
program prog2
subroutine sub_prog2(data_in, data_out, …)
…
end prog2
```

- ☺ efficient (memory exchange)
- ☺ one executable: easier to debug, easier for the OS

- ☹ 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 solutions to coupling

2. existing communication protocole

MPI (Message Passing Interface)
PVM (Parallel Virtual Machine)
TCPIP (Transmission Control Protocol/Internet Protocol)
SVIPC (System V Inter Process Communication)
CORBA (Common Object Request Broker Architecture)
Files on disk

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

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

☺ existing codes

☹ not easy to implement (need protocol expert)
☹ not flexible (hard coded exchanges)
☹ no use of generic transformations/interpolations
😐 Less efficient exchanges

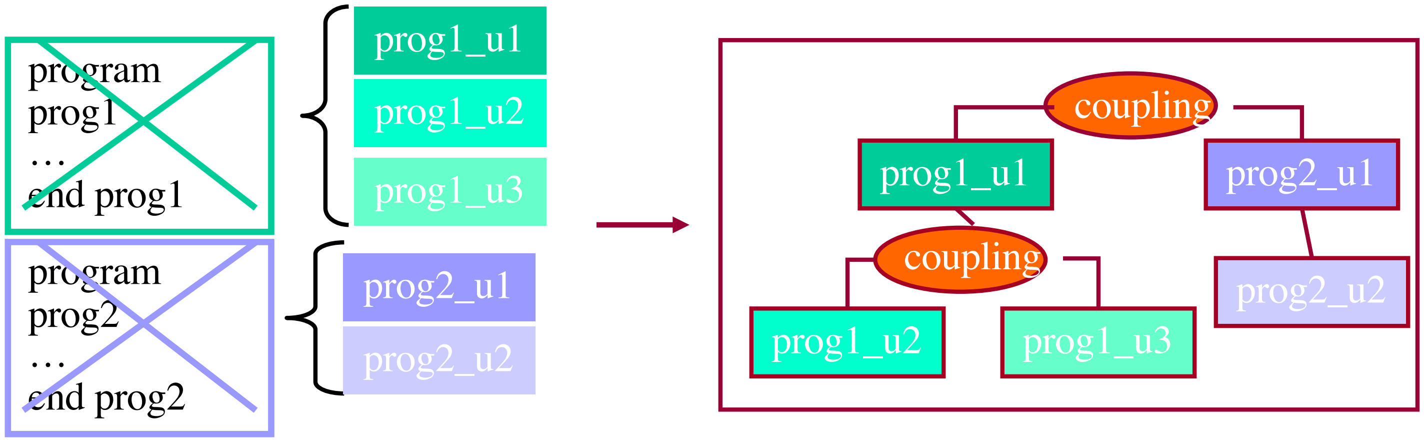
Different technical solutions to coupling

3. integrated coupling framework

ESMF **FMS**(GFDL) *CPL7* (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.)

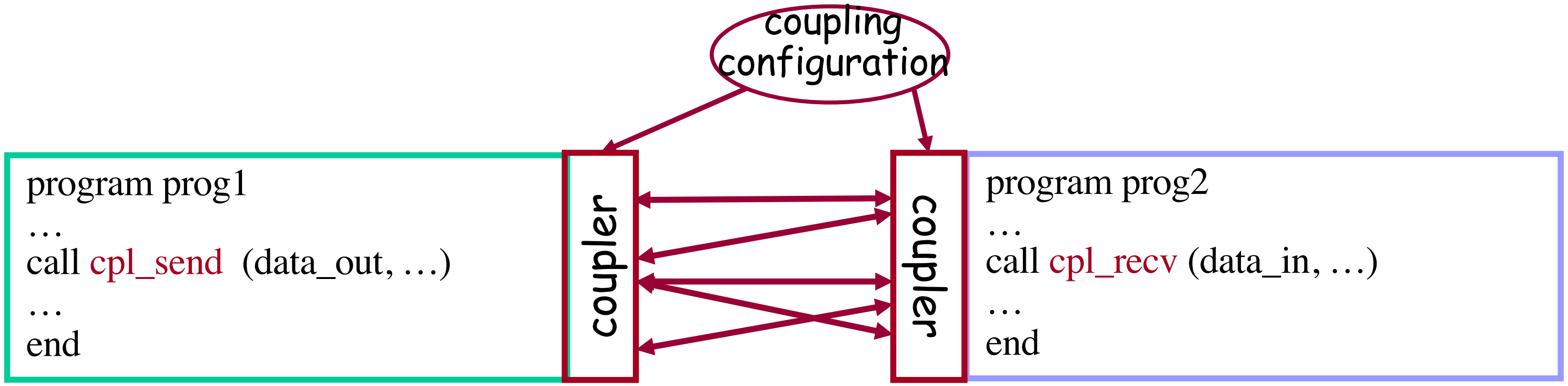
☹ existing codes
☹ not easy

→ probably best solution in controlled development environment

Different technical solutions to coupling

4. coupler or coupling library

C-Coupler



☺ existing codes

☺ use of generic transformations/regridding

☺ concurrent coupling (parallelism)

☹ efficient

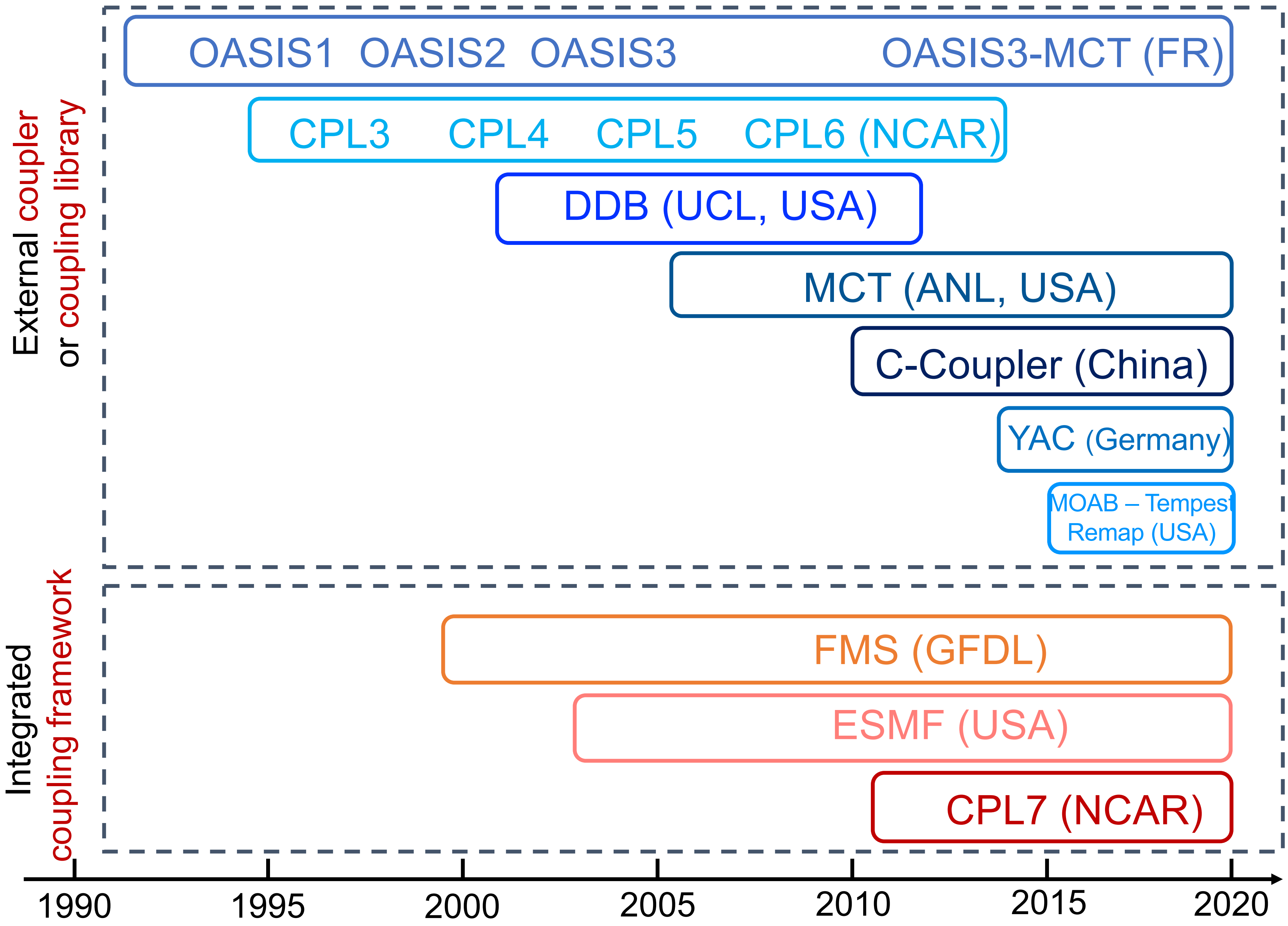
☹ multi-executable: good to avoid conflicts but may be more difficult to debug; harder to manage for the OS

→ probably best solution to couple independently developed codes

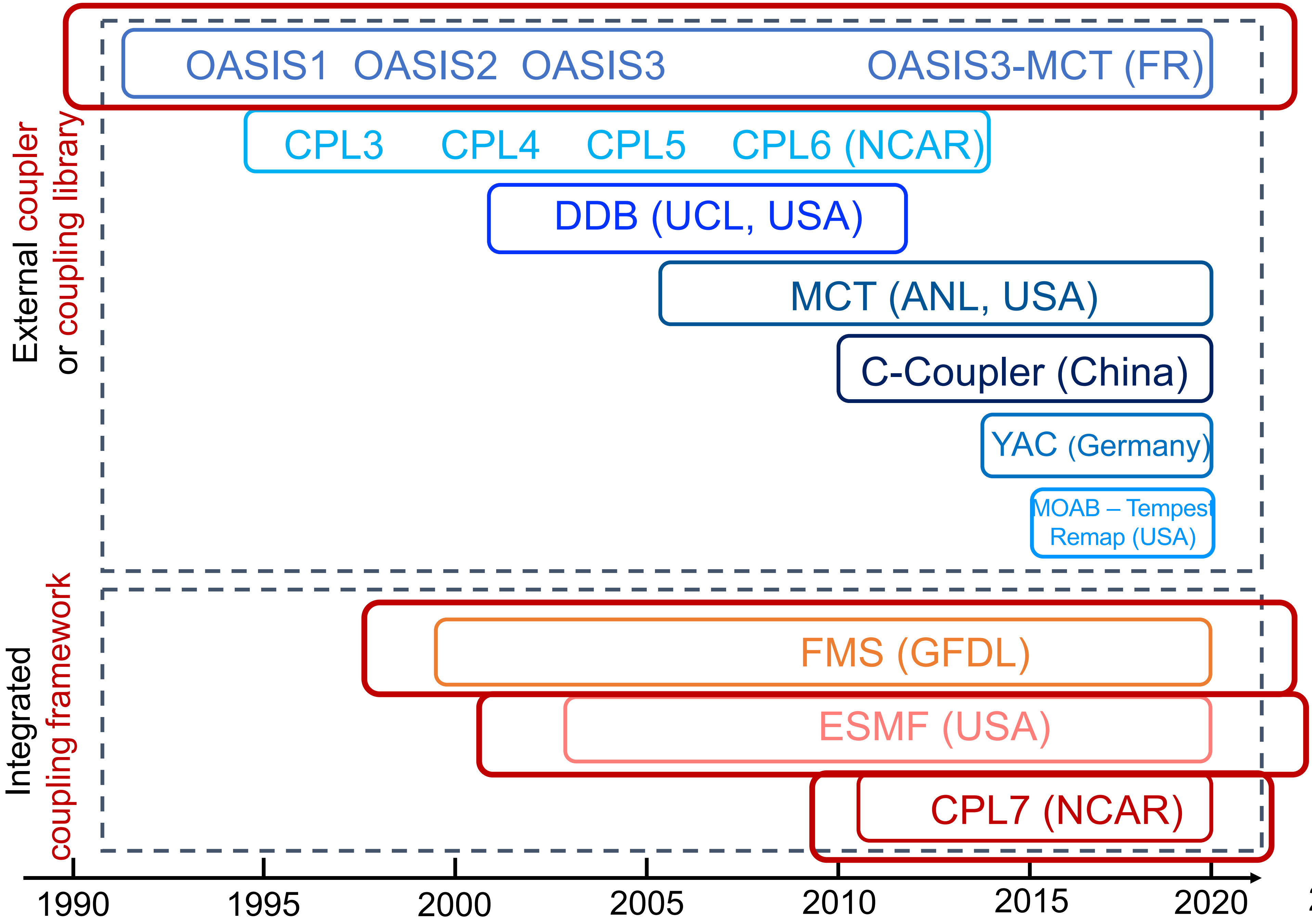
Outline

- Introduction
- Sequential and concurrent coupling
- Global performance of a coupled system
- Different technical solutions to coupling
- Few coupling software used in climate modelling
- Coupling algorithms in selected CGCMs
- Summary

Few coupling software used in climate modelling



Few coupling software used in climate modelling



Few coupling software used in climate modelling ESMF

ESMF Earth System Modeling Framework

Open source software for coupling model components to form weather, climate, coastal, and other Earth science related applications

- to favour exchanges of components between US groups

• Funded by NASA, DoD, NSF, NOAA; other partners : Air Force Weather Agency, Argonne National Lab, U. Michigan, MIT, UCLA, U. Maryland, COLA, CCA, GFDL, …

• Free and active user support

• Written in C++ , with F90 interface and partial interface to C/C++ and Python

• Over thousands of tests run automatically every night on different platforms

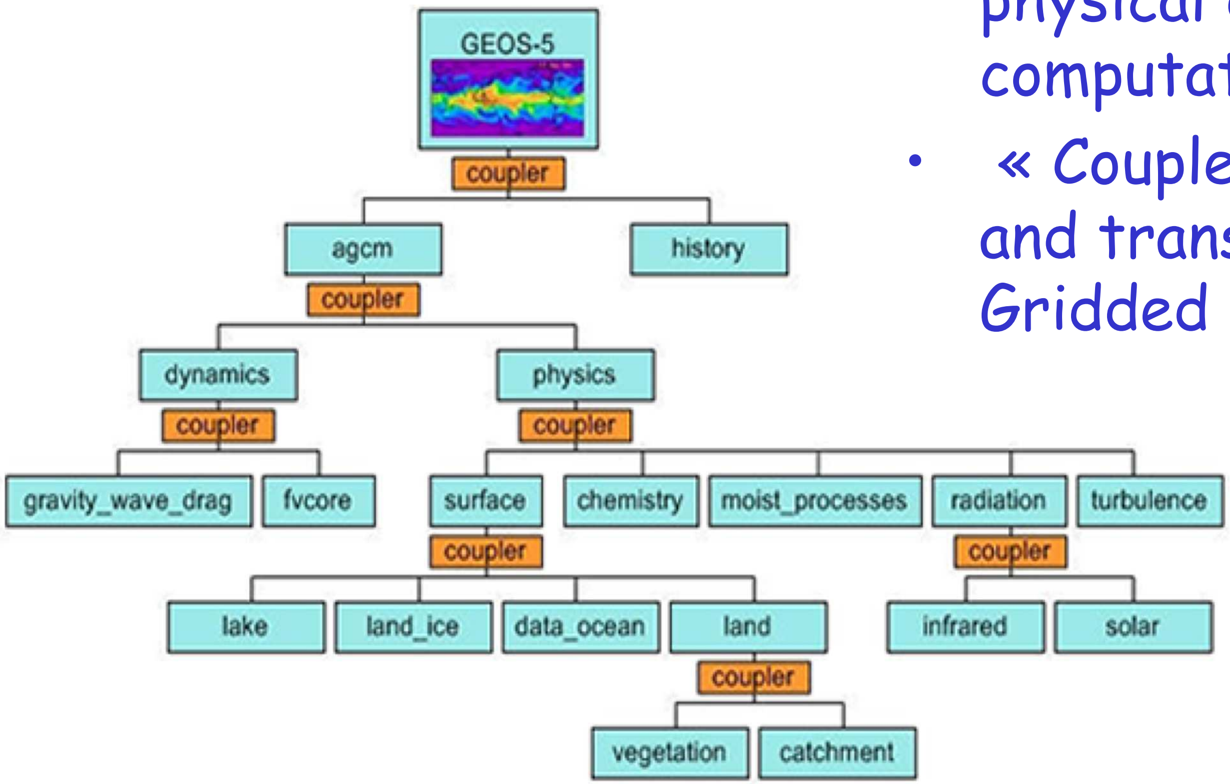
• ESMF/NUOPC used in major coupled systems at NASA, US Navy, NCAR, and NOAA and in other modelling applications from universities and major U.S. research centres.

• More than 30 ESMF/NUOPC-compliant models including atmosphere, ocean, sea ice, land ice, hydrology, land surface, chemistry/aerosol, ionosphere, and wave components

Few coupling software used in climate modelling ESMF

Earth System Modeling Framework

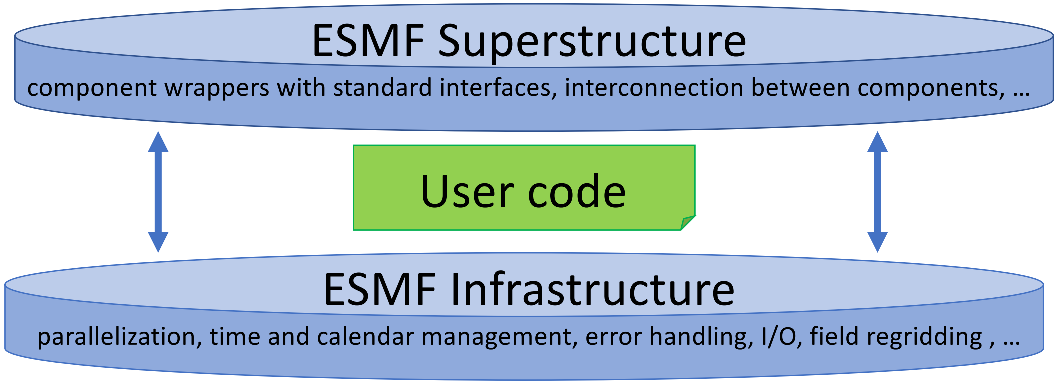
Component-based design: specific function, well-defined interface



- « Gridded » component: models a physical domain or realizes a computational function
- « Coupler » component: transforms and transfers physical fields between Gridded components

The scientist adapts its gridded components to ESMF standard calling interface and ESMF standard data structures.

Few coupling software used in climate modelling ESMF



The user code lies between:

- The "**Superstructure**", coupling the components
- The "**Infrastructure**" ensuring an efficient parallel execution on different computer architectures.

<u>ESMF "Infrastructure" :</u>

- internal parallelization,
- time and calendar management
- error handling, I/O
- regridding: 2D or 3D spherical or cartesian coordinates with nearest-neighbor, bilinear, higher order based on patch recovery, first- and second-order conservative

ESMF "Superstructure"

1. Define gridded components and split them into « init », « run » et « finalize » methods.
2. Adapt data structures to ESMF standards
3. Write or use coupler components
4. Write the driver code:

   Method registration and sequential or concurrent execution
5. Compile and run ESMF coupled application

```
subroutine myOceanRun (.. , impState, expState, clock, …)
 type(ESMF_State)    :: impState
```

```
subroutine  oceanToAtmCpl (.. ,)
 call ESMF_FieldRedist(oceanField, atmField, …)
```

```
...
call ESMF_GridCompSetEntryPoint
     (oceanComp, ESMF_SETRUN,
     myOceanRun, …)
…
call ESMF_GridCompRun(oceanComp,  …)
call ESMF_CplCompRun (oceanToAtmCpl,
     …)
call ESMF_GridCompRun(atmComp,  …)
```

Few coupling software used in climate modelling ESMF

National Unified Operational Prediction Capability (NUOPC) layer

A set of conventions and generic higher-level templates, to increase interoperability of ESMF components :

- **Connectors**, i.e. code implementing the communication between components
- **Mediators** wrapping custom coupling code (e.g. flux calculations)
- a **Driver** is specialized by harnessing specific NUOPC components, Mediators and Connectors

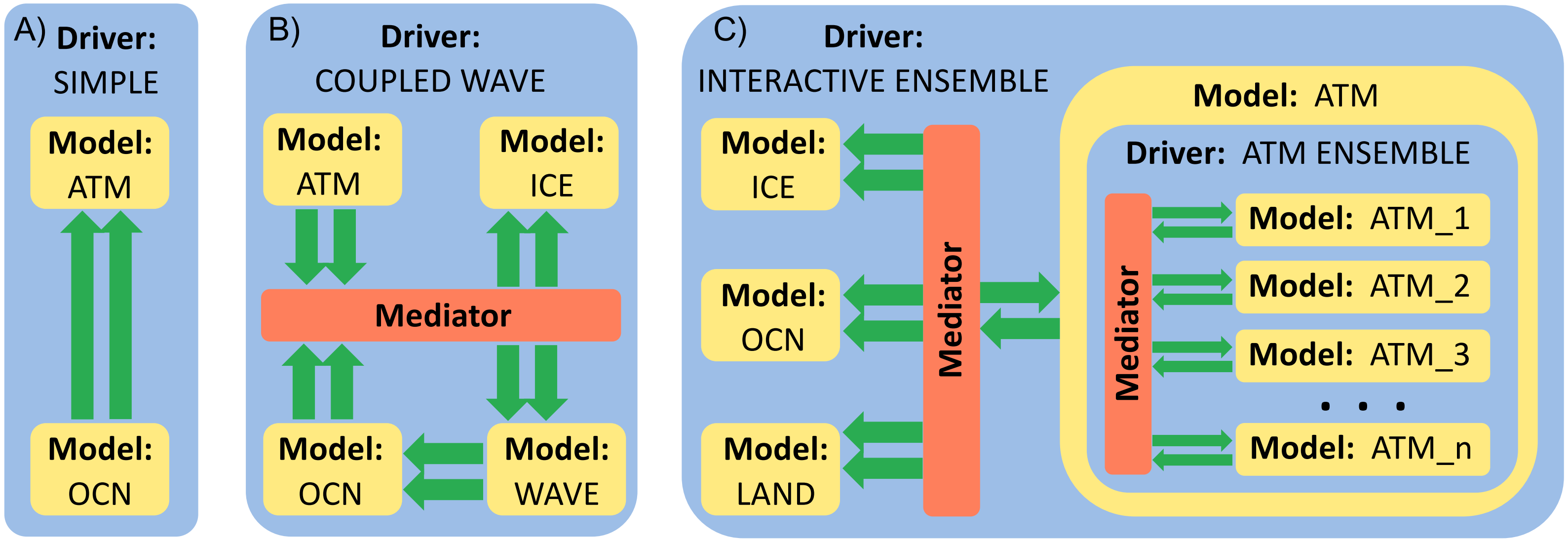


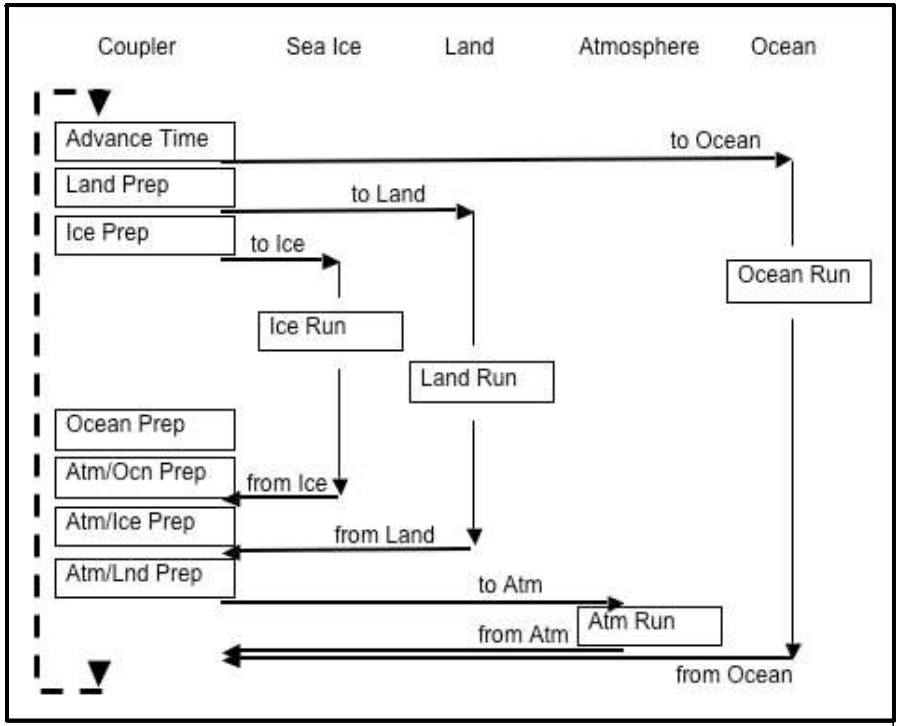
Figure courtesy of Gerhard Theurich from NRL/ESMF/SAIC

Few coupling software used in climate modelling CPL7

CPL7 for CCSM4 and CESM2

Software architecture with top-level driver and coupler component for flexible assembling of atmosphere, ocean, land and sea ice models into one executable via standard init/run/finalise interfaces

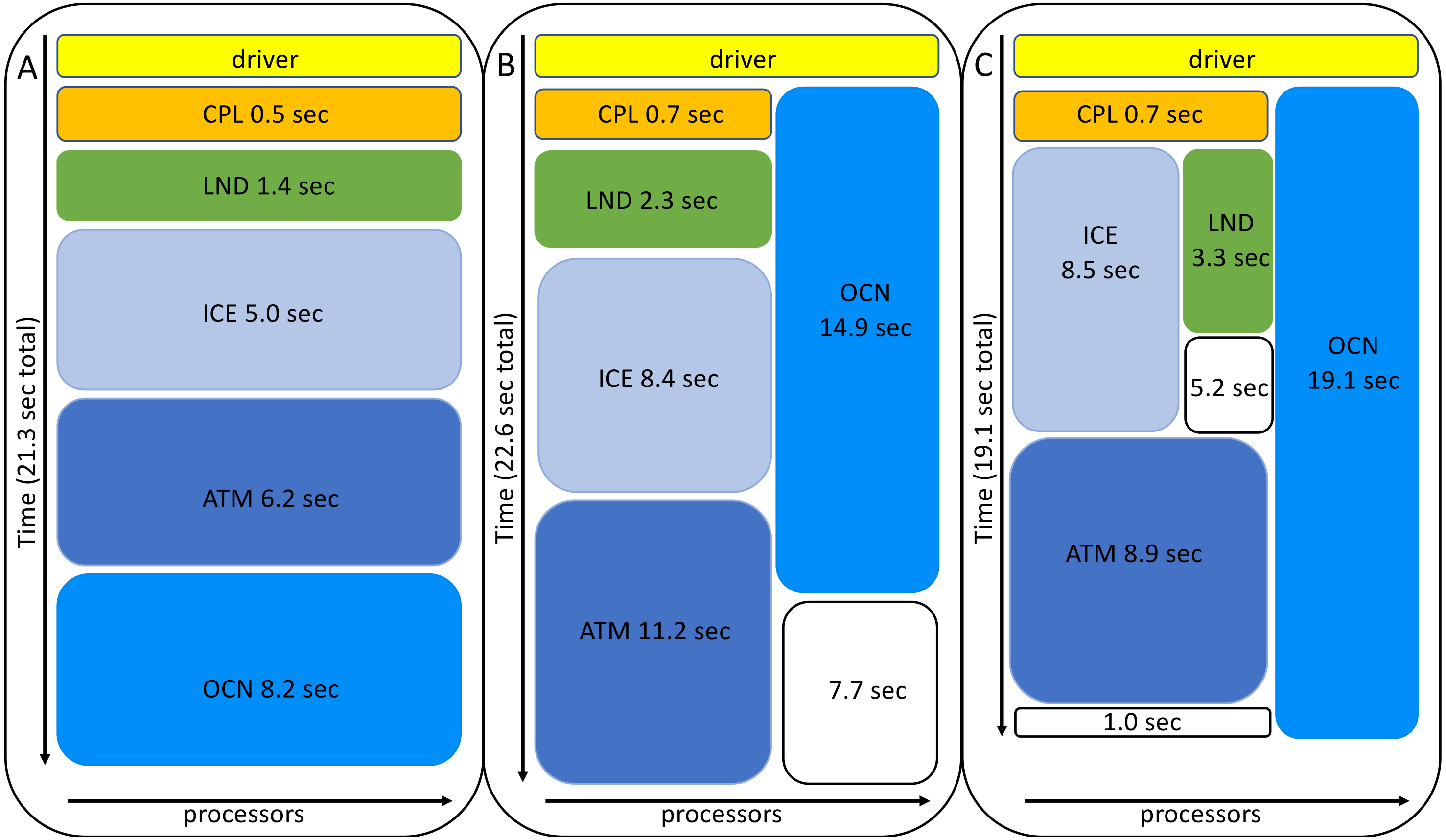
Driver Loop Sequencing



- Developed by the NCAR Earth System Laboratory, uses Argonne Nat Lab MCT for data regridding and exchange
- From multiple concurrent executables (cpl6) to one executable: time flow easier to understand, easier to debug
- Ability to add new components, new coupling fields, new capabilities (e.g. data assimilation) ; interface compatibility for ESMF-compliant components
- Ported to IBM p6, Cray XT4/XT5, BGP, Linux Clusters, SGI

Few coupling software used in climate modelling CPL7

➢ Varying levels of parallelism via external configuration (metadata) for proc layout:



➢ Scaling evaluated on up to 10 000 processors:

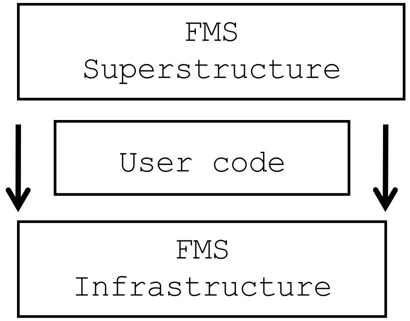
Craig et al., Int. J. High Perform. C, 2012

- flop intensive kernels: linear
- memory intensive operations: linear at low proc counts, flattens at high proc counts
- comm-dominated kernels: sub-linear at low proc counts; drops off for + 1000 procs.

Few coupling software used in climate modelling FMS

The Flexible Modeling System (FMS)

Software to assemble a climate model with a pre-defined driver managing the synchronization of domain-specific "slots" for atmosphere, ocean, ocean surface including sea ice and land surface



- Active for more than two decades at GFDL
- Use in CMIP6: GFDL-CM4 and GFDL-ESM4
- FMS shown to be scalable with up to O(10000) pes

FMS "Infrastructure":

- I/O, time management
- diagnostics and error handling
- operations on distributed gridded fields for different platforms

FMS "Superstructure":

- Domain-specific coupling layer ("stubs" (no component), or "data" also possible)
- Components "wrapped" in FMS-specific data structures and procedure calls
- Single executable with serial or concurrent execution of components
- Regridding, redistribution, or direct (hard-coded) exchanges between components

Few coupling software used in climate modelling FMS

Interface fluxes must be globally conserved

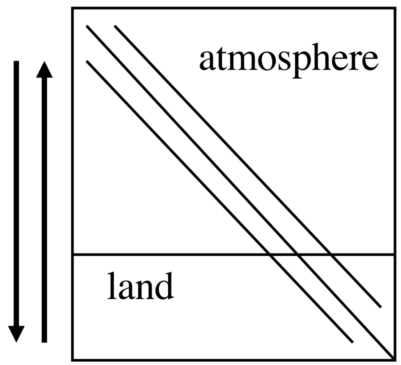
- quantities are transferred from the parent grids to the ***exchange grid***, where fluxes are computed; they are then averaged on the receiving grid

- Implicit calculation of vertical diffusive fluxes over the whole column
- Up-down sweep for tridiagonal matrix resolution through the exchange grid

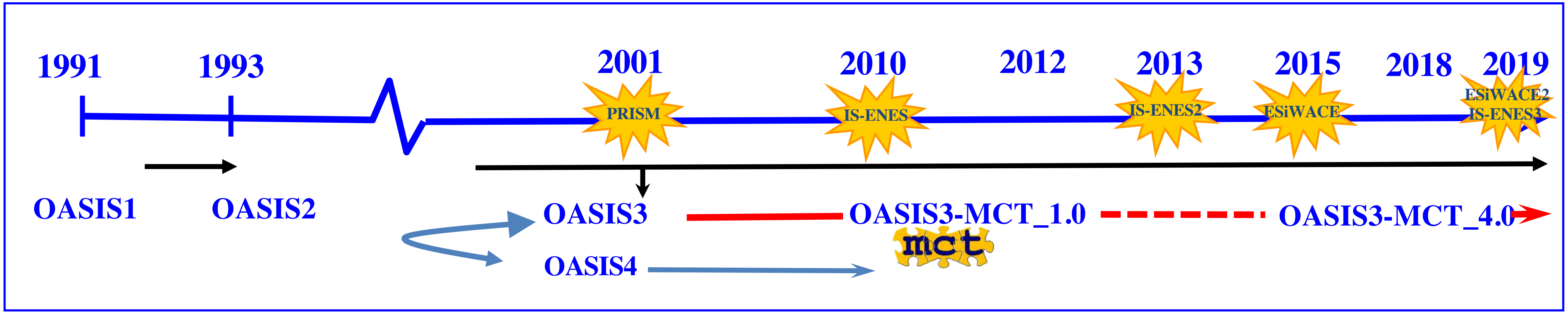
$$\partial T / \partial t = K \, \partial^2 T / \partial z^2$$

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

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



Few coupling software used in climate modelling OASIS

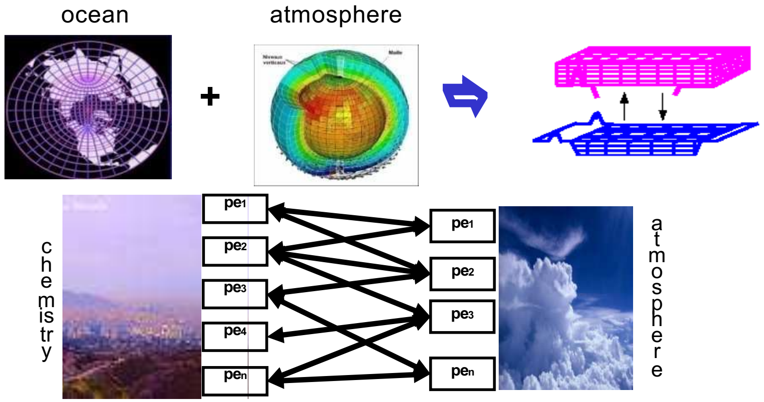


•OASIS1 -> OASIS2 -> OASIS3:

2D ocean-atmosphere coupling

low frequency, low resolution :

→ Flexibility, 2D interpolations



•OASIS4 / OASIS3-MCT:

2D/3D coupling of high-resolution parallel compo

→Parallelism, performance

➢ F90 & C, LGPL licence, public domain library (MPI, NetCDF, libXML, mpp_io, SCRIP)

Few coupling software used in climate modelling OASIS

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: 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

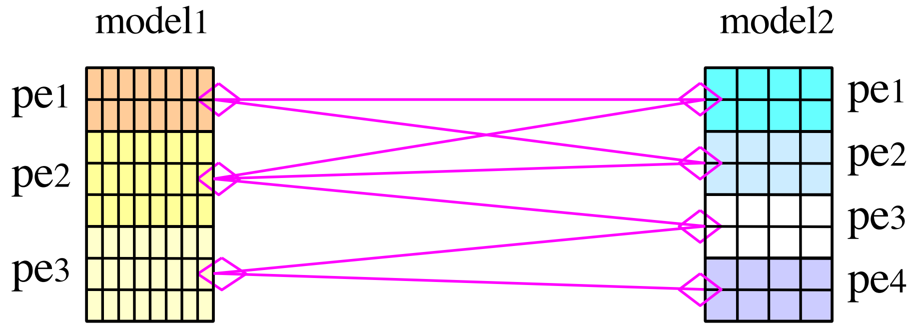
```
call oasis_put (…, date, var_array. …)
call oasis_get (…, date, var_array, …)
```

  - user defines the source or target in the external configuration file
  - 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 parallel communication

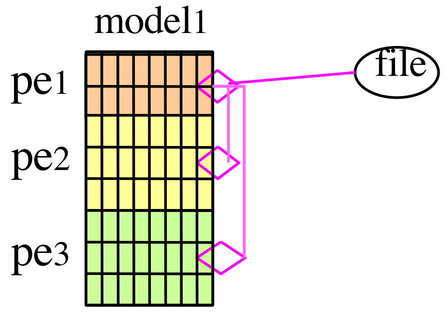
•Fully parallel communication between parallel models based on Message Passing Interface (MPI)



If required, the interpolation weights and addresses are calculated onto one model processes

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):



Few coupling software used in climate modelling OASIS

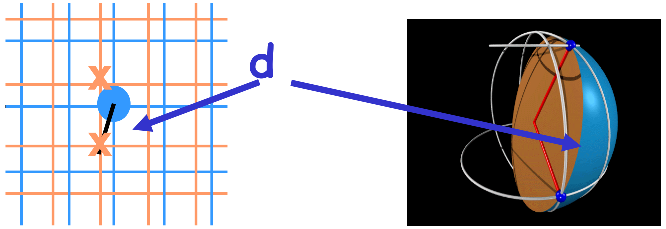
OASIS3-MCT regridding

SCRIP (Jones, 1999)

x source grid point
target grid point

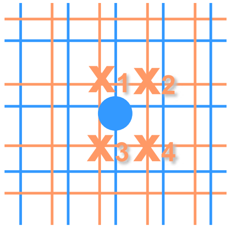
n-nearest-(gaussian-weighted)-neighbours:

weight(x) α 1/d
d: great circle distance on the sphere:



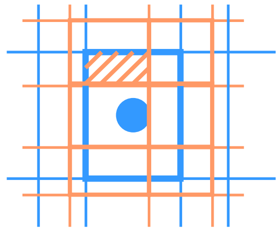
bilinear interpolation

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



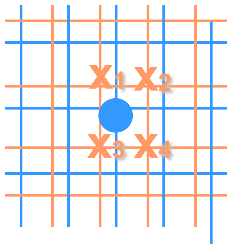
conservative remapping

➢ weight of a source cell % intersected area

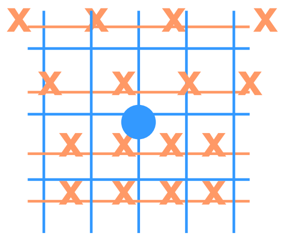


bicubic interpolation:

➢ 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

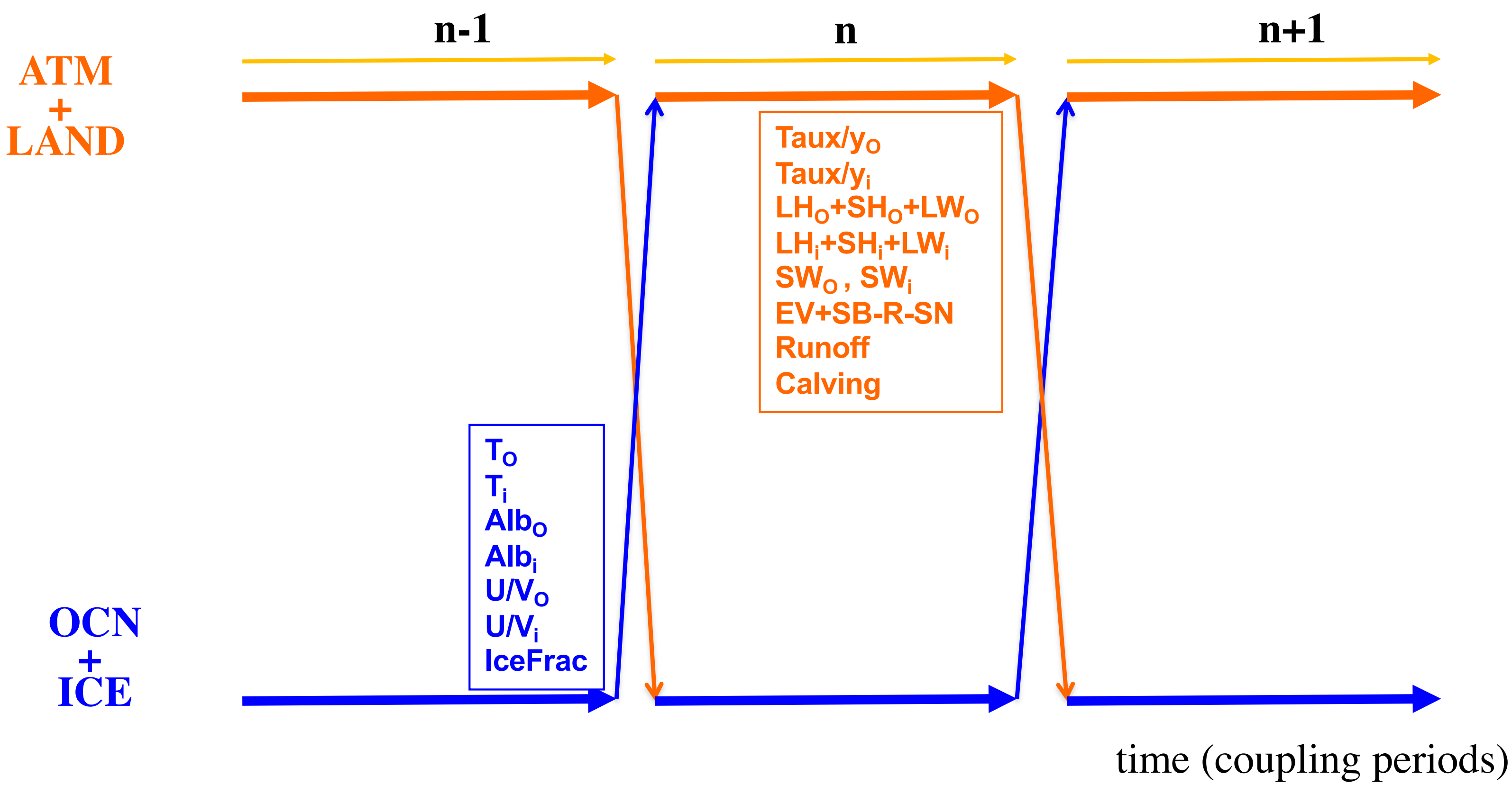


Outline

- Introduction
- Sequential and concurrent coupling
- Global performance of a coupled system
- Different technical solutions to coupling
- Few coupling software used in climate modelling
- Coupling algorithms in selected CGCMs
- Summary

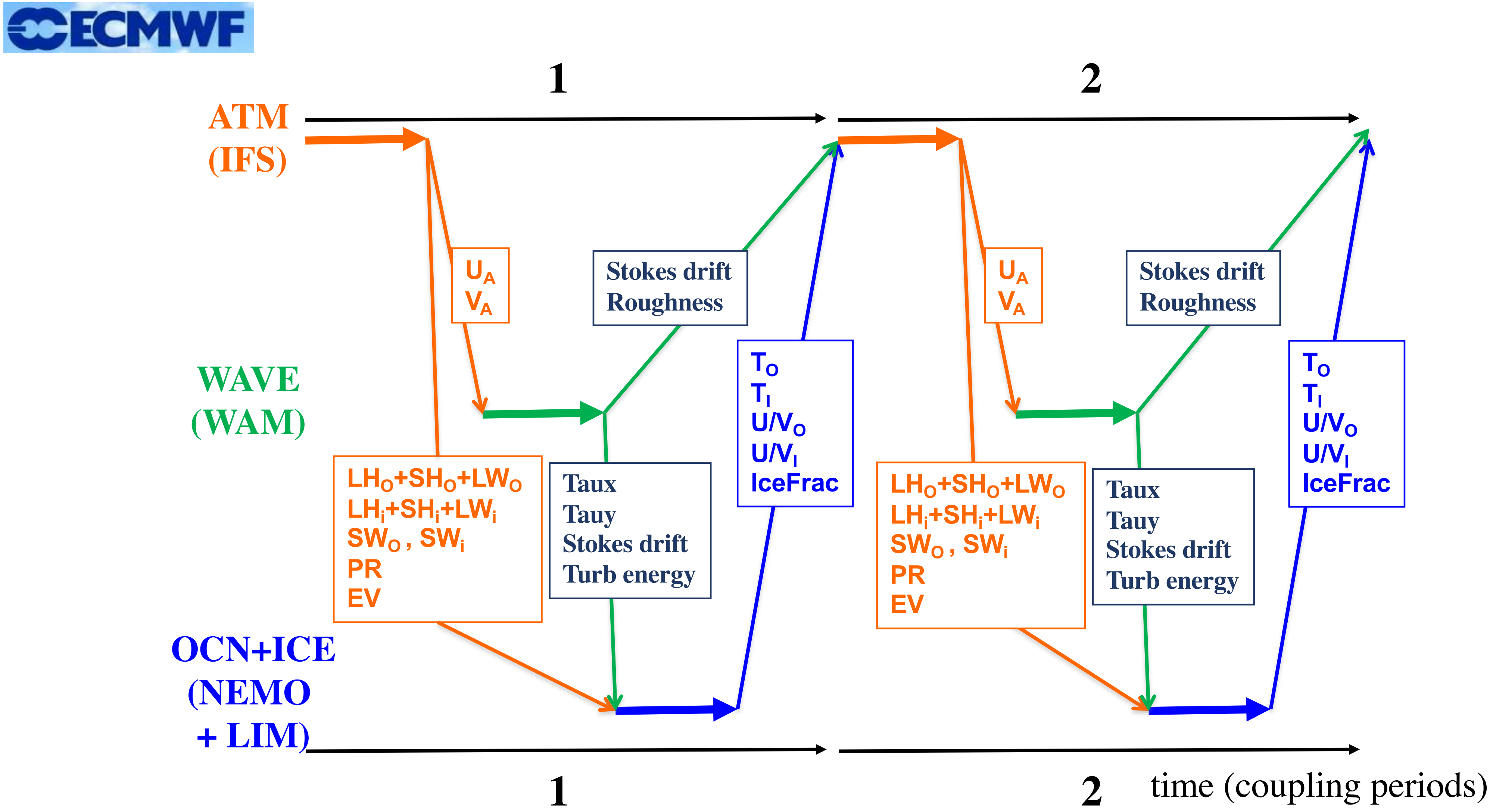
Coupling algorithms in selected CGCMs

Some European CGCMs:
CNRM-CM6, IPSL-CM6, EC-Earth3, MPI-ESM, HadGEM3



- Asynchronous coupling : models run concurrently and the coupling fields exchanged at the end of a coupling period are used in the target model as boundary condition for the next coupling period (typically 1 to 3 hours)

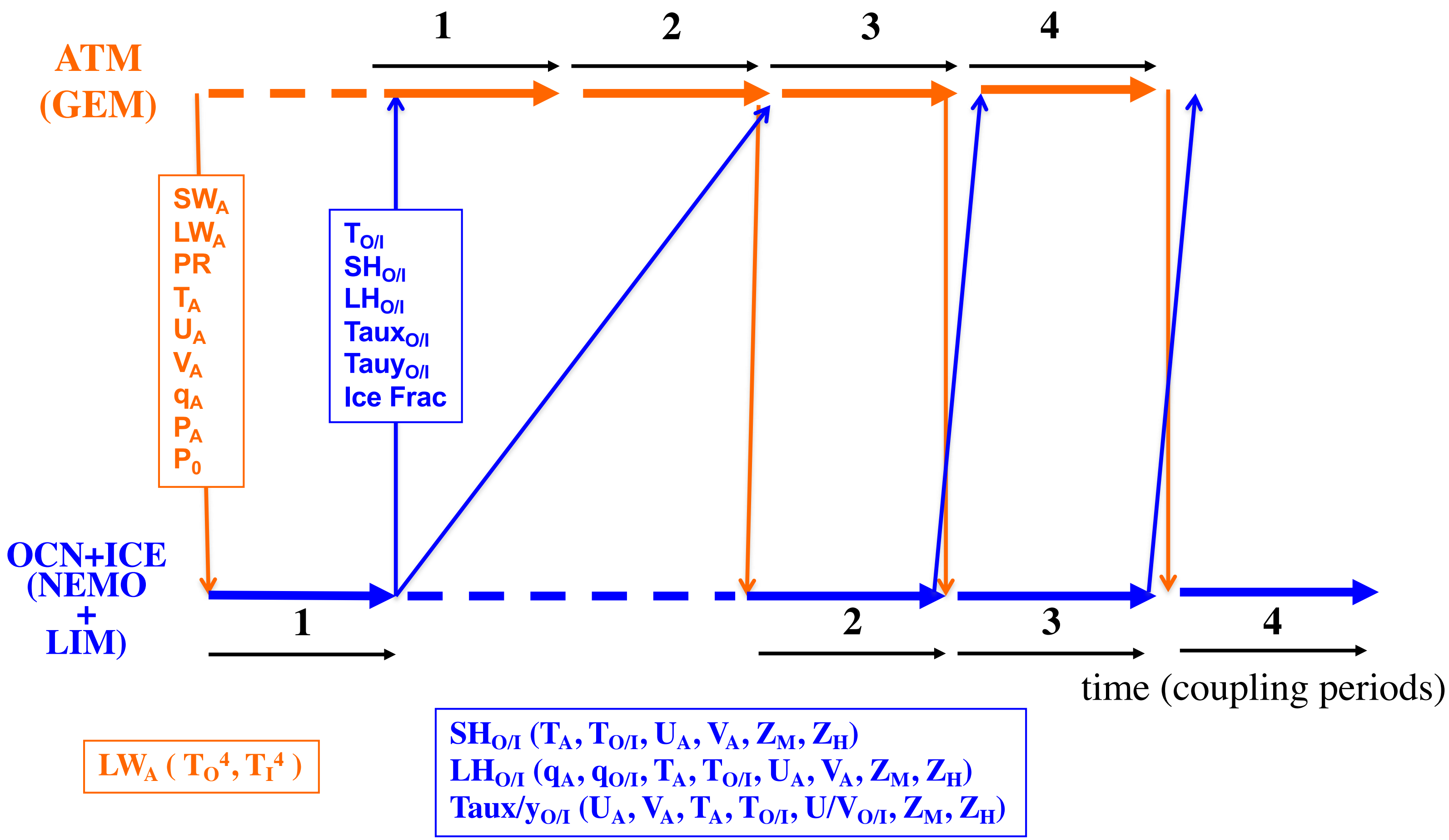
Coupling algorithms in selected CGCMs



- ✧ WAM: wave-model
- ✧ One-executable with sequential execution of components

Coupling algorithms in selected CGCMs

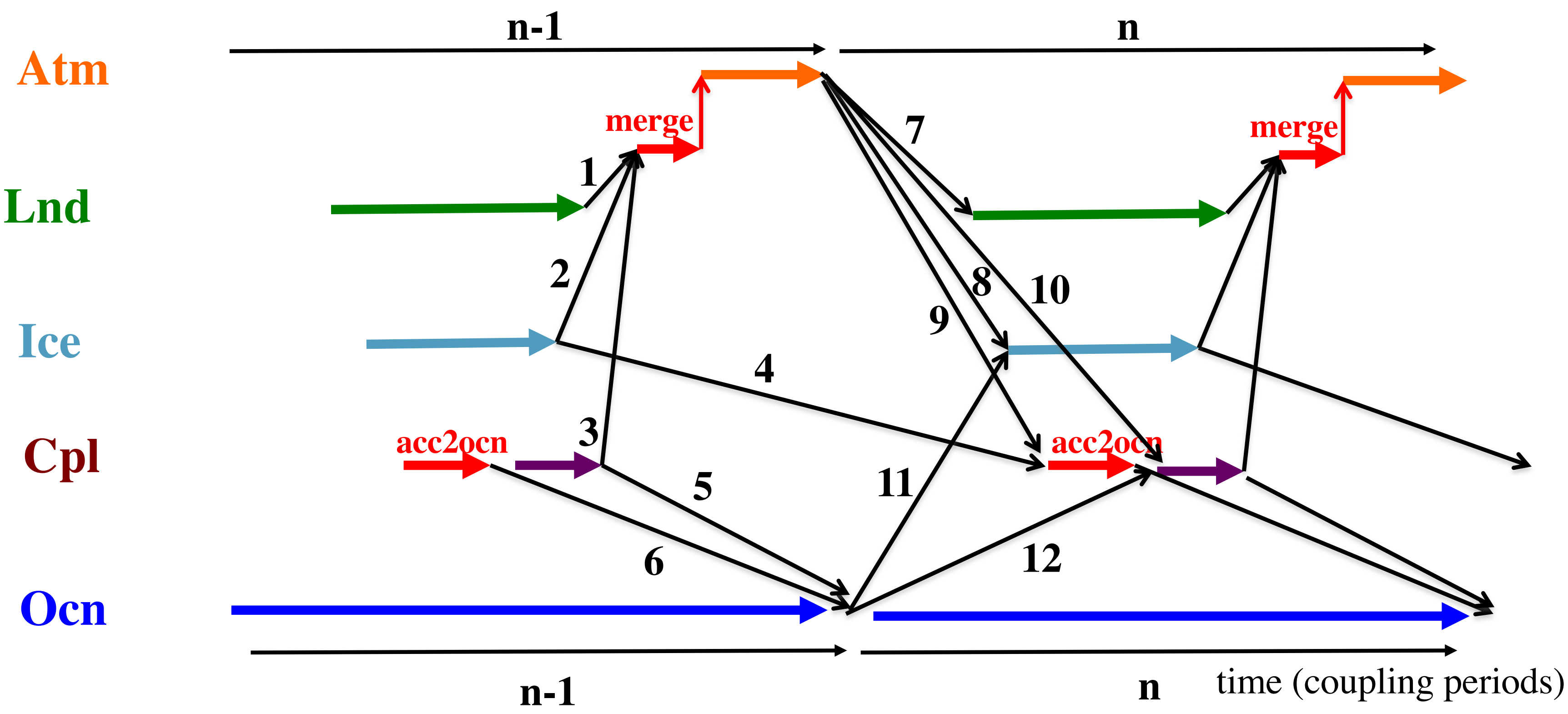
Environment Canada



- Surface turbulent fluxes (LH,SH, stress) calculated in the ocean
- Asynchronicity of 2 coupling periods for ocean->atmos fields, no asynchronicity for atmos->ocean fields

Coupling algorithms in selected CGCMs

CESM2 (NCAR) and CMCC-CM2 (CMCC)



1: LWu, LH, SH, Taux Tauy, EV, Alb, T_l, SNd, W10m, T2m
2: LWu, LH, SH, Taux Tauy, EV, Alb, T_i, SNd, W10m, T2m, IceFrac
3, 5: LWu, LH, SH, Taux Tauy, EV, Alb (for 3 only)
4, 6: IceFrac, ice/ocean heat & salt fluxes & stresses, SWp
7, 8: Ws, Ts, Ps, qs, SWd, LWd, rain, SN
9: Ps, SWd, LWd, rain, SN
10: Ws, Ts, Ps, qs
11, 12: T_o, S_o, Sl, U_o, V_o

- Turbulent diffusive fluxes (LH,SH, stresses) calculated in surface modules **Lnd**, **Ice, Cpl** are aggregated in "**merge**" for **Atm**.
- SWd, LWd, water fluxes calculated in **Atm** sent to **Lnd**, **Ice** and to **Ocn** via **acc2ocn** with a lag of 2 coupling periods.

Outline

- Introduction
- Sequential and concurrent coupling
- Global performance of a coupled system
- Different technical solutions to coupling
- Few coupling software used in climate modelling
- Coupling algorithms in selected CGCMs
- Summary

Summary

- Playing with the lags of the coupling fields, one can implement a *sequential* or *concurrent* coupling between two components.
- Advantages and disadvantages of the different implementations in terms of performances.
- Few coupling software used in climate modelling: ESMF, CPL7, FMS, OASIS
- Different coupling sofware can be classified into two main categories: "external coupler or coupling library" or "integrated coupling framework"
- Different coupling algorithms in selected CGCMs illustrating *sequential* or *concurrent* coupling , compromises between physical consistency and performances.

The end