

Geoscientific Model Development Discussions is the access reviewed
discussion forum of *Geoscientific Model Development*

OASIS4 – a coupling software for next generation earth system modelling

R. Redler¹, S. Valcke², and H. Ritzdorf¹

¹NEC Research Laboratories, NEC Europe Ltd., Sankt Augustin, Germany

²CERFACS, Toulouse, France

Received: 19 June 2009 – Accepted: 22 June 2009 – Published: 7 July 2009

Correspondence to: R. Redler (rene@r-redler.de)

Published by Copernicus Publications on behalf of the European Geosciences Union.

GMDD

2, 797–843, 2009

**The OASIS4 software
for coupled climate
modelling**

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Abstract

In this article we present a new version of the Ocean Atmosphere Sea Ice Soil coupling software (OASIS4). With this new fully parallel OASIS4 coupler we target the needs of Earth system modelling in its full complexity. The primary focus of this article is to describe the design of the OASIS4 software and how the coupling software drives the whole coupled model system ensuring the synchronization of the different component models. The application programmer interface (API) manages the coupling exchanges between arbitrary climate component models, as well as the input and output from and to files of each individual component. The OASIS4 Transformer instance performs the parallel interpolation and transfer of the coupling data between source and target model components. As a new core technology for the software, the fully parallel search algorithm of OASIS4 is described in detail. First benchmark results are discussed with simple test configurations to demonstrate the efficiency and scalability of the software when applied to Earth system model components. Typically the compute time needed in order to perform the search is in the order of a few seconds and is only weakly dependant on the grid size.

1 Introduction

Global coupled models (GCMs) have been used for climate simulations since the late 1960s starting with the pioneering work by Manabe and Bryan (1969). With advances in computing power, coupled models have been in use for century-long simulations since the late 1980s. Since that time, we observe a rapid and continuous increase of activity in global coupled modelling as additional computer resources have become available. Climate models can be considered as multi-disciplinary or multi-physics software tools to simulate the interactions of the atmosphere, oceans, land surface, sea ice and other components of the climate system. They are used for a variety of purposes from studies of the dynamics of the weather and climate system to projections

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



of future climate. These models can range from relatively simple to quite complex, that is from zero-dimensional models and Earth-system Models of Intermediate Complexity (EMIC) to complex 3-dimensional Global Climate Models (GCM) and Regional Climate Models (RCM).

5 A GCM aims to describe geophysical flow by integrating a variety of fluid-dynamical, chemical, or even biological equations that are either derived directly from physical laws (e.g. Newton's law) or constructed by more empirical means. Classically, the two main constituents of a GCM are atmospheric and ocean circulation models. When coupled together (along with other components such as a sea ice model and a land
10 model) an ocean-atmosphere coupled general circulation model forms the basis for a full climate model. A very recent trend in GCMs is to extend these traditional coupled climate models to become full Earth system models (ESM), by including further model components e.g. for atmospheric chemistry, marine biology or a carbon cycle model (matter cycle in a more general sense) to more correctly simulate the interaction be-
15 tween the different sub-components. Typically, these different model components are developed independently by different research groups.

Numerical modelling in its full complexity is still in its infancy, primarily due to the immense complexity of the task and insufficient empirical knowledge of many aspects of climate related processes. This is not only the case for the oceans where in particular the knowledge of the subsurface circulation is rather poor, but also for the hydrological cycle where precipitation, evaporation and the three-dimensional distribution of water vapour and clouds are insufficiently known. Despite these uncertainties the development of global climate models is one of the main unifying components in the
20 World Climate Research Programme (WCRP) of the World Meteorological Organisation (WMO). The output of these models is considered as fundamental deliverables from the whole WCRP and provides the basis for the understanding and prediction of natural and human-made climate variations (CLIVAR Scientific Steering Group, 1998).

25 In the past, the important role of climate models has been addressed in a variety of research programmes on a global scale like the Climate Variability programme (CLI-

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



VAR) or the Coupled Model Intercomparison Project (CMIP) as one of the CLIVAR initiatives, not to forget efforts on a more regional scale like EuroCLIVAR, the Baltic Sea Experiment (BALTEX) or the North American Regional Climate Change Assessment Program (NARCCAP), just to mention a few. Most important, global climate models form one of the backbones of the Assessment Reports (AR) published by the Intergovernmental Panel of Climate Change (IPCC).

In addition to the model components used in an ESM, each community representing one of the physical components still feels a strong need for having separate standalone special-purpose versions of the individual components of the coupled climate models to investigate processes in the different subsystems and to test new physical parameterisations in controlled scenario runs. The existence of different research objectives and the need to estimate model uncertainty from model intercomparison (like for the IPCC AR) support the idea of a multi-component approach and the need for the interoperability of the different components. A multi-component approach helps to avoid complex technical problems arising from an all-in-one code integrating independently developed components. Furthermore independent codes are of great advantage for the flexibility of the system (Döscher et al., 2002). It is therefore only natural to consider a coupling of any two or more physical component models through some independent coupling software ensuring the lowest possible degree of interference in the component codes themselves.

With OASIS4, we continue the software development of the Ocean Atmosphere Sea Ice Soil (OASIS) coupling software which has its origins in the early 90ies at the European Centre for Research and Advanced Training in Scientific Computing (CERFACS, Toulouse, France). With the emerging complexity of ESMs, their increasing size (with respect to the number of grid points to be treated) and higher degree of parallelism, the limitations of OASIS3 and its predecessors have become visible and the need has emerged to provide the ESM community with a new fully parallel coupling software. The development of OASIS4 started during the Partnership for Research Infrastructures in earth System Modelling (PRISM) project funded by the European Commission from

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



2001 to 2004. PRISM (Valcke et al., 2007) was organized by the European Network for Earth System Modelling (ENES), and is currently continuing within the new Framework 7 Programme funded by the European Commission with the IS-ENES (Infrastructure for the European Network for Earth System modelling) project.

5 At the time when the OASIS4 development started, coupling software performing field transformation already existed, such as the Mesh based parallel Code Coupling Interface (MpCCI) (Joppich and Kürschner, 2006) or the Community Climate System Model (CCSM) Coupler 6 (Cpl6) (Buja and Craig, 2002). MpCCI is not available as source code, and for this particular reason the MpCCI software has not been accepted
10 by the ESM community. With recent new versions MpCCI has abandoned some important aspects of parallel data exchange: compared to previous versions all grid information is now assembled in a single instance for performing the neighbourhood search and regridding. This approach makes it even less attractive for the usage within an ESM with a high degree of parallelism. The concept of CCSM and Cpl6 is very much
15 similar to OASIS4. However, earlier versions of Cpl6 have targeted the specific needs of the CCSM and the coupler could only be executed within the complete CCSM. The Earth System Modelling Framework (ESMF) software environment (Hill et al., 2004) provides another approach to create coupled modelling systems. ESMF is a high-performance software infrastructure that can be used to build complex application as
20 a hierarchy of individual units, each unit having a coherent function, e.g. modelling a physical phenomenon or managing the input and output (IO), and a standard calling interface. While an ESMF application, being more integrated, will most probably be more efficient compared to an OASIS4 coupled system, ESMF requires a deeper level of intervention in the application code and imposes strict coding rules in order to take
25 full advantage of its functionality. In contrast to ESMF, the OASIS4 application programmer interface (API) can be integrated with only very minor modifications in the original application code, and is for this particular reason more attractive to the European ESM community with its less centralised modelling efforts.

OASIS4 does not include any assumptions about the physical nature of the coupling

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



fields. It may be considered a slight disadvantage that apart from providing some general arithmetic functions (like adding or multiplying with constants or time averaging) any specific physically motivated pre-and post-processing regarding the coupling of physical quantities remains under the full responsibility of the application code. In compensation, the flexibility of OASIS4 provides the ESM community with the ability to couple a wide range of ESM components.

With this article we address the questions we have received from the growing user community during the last couple of years. The originality of OASIS4 relies in its great flexibility which is explained in Sects. 2 to 4, where the software design ideas, the driving and configuration mechanisms and the application programmer interface to the communication library are detailed. Another new concept introduced with OASIS4 is the parallel 3-dimensional neighbourhood search and regridding based on the geographical description of the process local domains (see Sects. 5 and 6). The data flow, in particular the common treatment of coupling and IO exchanges (a concept already available with OASIS3) is briefly described in Sect. 7.

We conclude the technical description with first performance measurements shown in Sect. 8 and outlook to ongoing and future work in Sect. 9.

2 General design and overview

The key concept behind the OASIS4 software is to provide a tool to the ESM community which is portable to any existing computing environment. Mainly this is achieved by adhering to programming standards that have emerged over time. For a complete list of systems and environments which are supported by OASIS4, the reader is invited to visit the OASIS4 Wiki page¹.

The OASIS4 software package provides the source code for a Driver-Transformer executable and a library (called PSMILe hereafter). At run-time, the OASIS4 Driver-

¹<https://oasistrac.cerfacs.fr/>

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Transformer and the component models remain separate (possibly parallel) executables. To communicate with the rest of the coupled system, each component model needs to be linked against the PSMILe. To be as user-friendly as possible, the PSMILe API includes a limited number of required function calls, at the same time supporting all typical ESM component data structures. While it is not designed to handle the component internal communication, the library completely manages the coupling data exchanges with other model components and the details of the I/O file access. In this article, we will use the term “coupled application” to describe the ensemble formed by the component models coupled through the OASIS4 Driver-Transformer.

The major task of OASIS4 is to handle the data exchange between ESM components. These exchanges are completely built upon the Message Passing Interface (MPI) (Snir et al., 1998; Gropp et al., 1998). The major justification for this decision is the fact that MPI has become a quasi standard to handle parallel applications. MPI libraries are available on almost every parallel architecture, be it a vendor-specific implementation or provided as one of the many publicly available MPI packages. The complexity of MPI is nevertheless hidden behind the OASIS4 API.

In contrast to previous versions of OASIS, the coupled configuration has to be described by the user with the help of Extensible Markup Language² (XML) files. In order to read in this information, linking to an XML library is required. Similar to OASIS3, NetCDF (Rew and Davis, 1997) file IO is supported to optionally read and write forcing input, diagnostic output and coupling restart files; in this case, a NetCDF or a parallel NetCDF (Li et al., 2003) library needs to be provided. Last but not least, standard-conform Fortran90 and C compilers are required.

With the current version bilinear, trilinear, bicubic and nearest-neighbour parallel interpolation is provided for the most prominent grid types used in ESM components to map the source grid values to the target grid in a data exchange. Furthermore 2-D conservative remapping is available which is usually the preferred way to interpolate fluxes.

²<http://www.w3.org/XML/>

**The OASIS4 software
for coupled climate
modelling**

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



In particular OASIS4 supports 2- and 3-dimensional coupling between any combination of logically-rectangular, i.e. 2-D structured, or Gaussian Reduced grids in the horizontal longitude-latitude plane, each horizontal layer repeating itself at different vertical levels. OASIS4 is also able to handle the exchange of non-geographical data, i.e. data which are solely provided in an (i, j, k) index space not associated to any geographical domain. To properly handle non-geographical data, it is required that the pairs of source and target grids work over the same global index range, although the data can be partitioned differently on the source and target side.

3 The driving and configuration mechanisms

The OASIS4 Driver-Transformer executable can consist of one or more processes. During the initialisation, the root process of this instance acts as a Driver while during the integration of the coupled model the root and additional processes act as Transformer processes performing the regridding of the coupling fields. The Transformer task is further discussed in Sects. 6 and 8. In this section we focus on the Driver aspect.

OASIS4 supports two ways of starting a coupled application. With a complete implementation of the MPI2 standard, only the Driver processes have to be started by the user. All remaining physical components are then spawned by the OASIS4 Driver at run time. In this scenario, no extra programming work has to be invested in the component internal parallelisation as any original MPI communication will work as is. If the available MPI library does not support the MPI2 standard, all processes of the whole coupled application will have to be started simultaneously in an “multiple program multiple data” (MPMD) mode; the disadvantage of this approach is that each component must use a specific MPI communicator for its own internal communication. This MPI communicator is created and provided by the OASIS4 software.

The whole coupled configuration is described with the help of XML files. XML is a simple, very flexible text format. Originally designed to meet the challenges of large-

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. An XML document is simply a file which follows the XML format. The OASIS4 configuration XML files must be created by the coupled model developer. A Graphical User Interface (GUI) is currently being developed to facilitate the creation of those files.

The “Specific Coupling Configuration” (SCC) XML file specifies the general characteristics of a coupled model run, for example the components to be coupled and the number of processes for each component. The Specific Model Input and Output Configuration (SMIOC) XML files (one for each component) specify the relations each component model will establish with the external environment through inputs and outputs for a specific run. In the SMIOC files, the user specifies the coupling exchanges, for example the source or target of each input or output field (either another component or a disk file), the exchange frequency, and the transformations to be performed by OASIS4 on this field.

The contents of the XML files is read and interpreted by the Driver, and information relevant for the physical components is communicated to the respective PSMILe and Transformer processes on case-by-case basis.

4 The application programmer interface

In this section, we briefly describe the design ideas of the OASIS4 API to the PSMILe and highlight the differences compared to OASIS3.

The API function calls (see Table A1) can be split into three different phases. During the initialisation phase, some basic initialisation routines have to be called and the grid and exchange fields have to be announced to the PSMILe. The initialisation phase is concluded with a call to PRISM_Enddef (see Sect. 5). The second phase comprises the exchange of data further described in Sect. 7 while the third phase denotes the termination of the coupling. For a detailed description of the OASIS4 API, we refer the reader to the OASIS4 User Guide (Valcke and Redler, 2006).

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



4.1 General design

The general design follows several principles. First, the PSMILe API routines that are defined and implemented shall not be subject to modifications between the different versions of the OASIS4 software. However, new routines may be added in the future to support new functionality. In addition to that, the PSMILe has been kept extendable to new types of coupling data and new types of grids by using Fortran90 function overloading. Some complexity of the API arises from the need to transfer not only the coupling data but also the associated metadata, i.e. mainly the information about the associated grid. Second, a trade-off has been chosen between a large number of subroutines provided each with a shorter list of parameters and a small number of subroutines, each one having a longer and more complex parameter list.

Like in the MPI, the memory that is used for storing internal representations of various data objects is not directly accessible by the user and the objects can only be addressed indirectly by the component via their handle. Handles are of type integer and represent an index to an entry in a list of the respective objects. The object and its associated handle are significant only on the process where it is created.

Furthermore, the internal usage of MPI is completely hidden from the user, except when the available MPI library does not offer the MPI2 spawning functionality (which implies that all processes have to be started simultaneously in an MPMD mode, see Sect. 3). In this case, each component has to retrieve for its internal communication a special MPI communicator created by the PSMILe and accessible with a specific function call to PRISM_Get_localcomm (see Table A1); this call remains the only MPI-related call.

When using MPI for internal communication in parallelised components, the calling sequence of some MPI- and PRISM-related calls has to follow a certain rule which is sketched in Fig. 1. The function calls to PRISM_Init and PRISM_Init_comp can be used as a full substitute of MPI_Init, while the PRISM_Terminate can be used as a full substitute for MPI_Finalize.

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



4.2 Data exchanges

The PSMILe manages the coupling data flow between any two (possibly parallel) component models with a communication pattern completely hidden from the component codes, following a principle of “end-point” data exchange. When producing data, no assumption is made in the source component code concerning which other component will consume these data or whether they will be written to a file, and at which frequency; likewise, when asking for data, a target component does not know which other component model produces them or whether they are read in from a file. The target or the source (another component model or a file) for each field is defined by the user in the SMIOC XML file (see Sect. 3) and the coupling exchanges and/or the IO actions take place according to the user external specifications. The switch between the coupled mode and the forced mode is therefore totally transparent for the component model.

The sending and receiving PSMILe calls PRISM_Put and PRISM_Get can be placed anywhere in the source and target code and possibly at different locations for the different coupling fields. These routines can be called by the model at each timestep. The actual date at which the call is performed and the date bounds for which it is valid are given as arguments; the sending/receiving is actually performed only if the date and date bounds correspond to a time at which it should be activated, given the field coupling or IO dates indicated by the user in the SMIOC; a change in the coupling or IO dates is therefore also totally transparent for the component model itself.

When the action is activated at a coupling or IO date, each process sends or receives only its local partition of the data, corresponding to its local grid defined previously. The coupling exchange, including data repartitioning if needed, then occurs, either directly between the component models, or via additional Transformer processes if regridding is needed.

If the user specifies that the source of a PRISM_Get or the target of a PRISM_Put is a disk file, the PSMILe exploits the GFDL mpp_io package Balaji (2001) for its file IO (see Sect. 7 for more detail).

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



4.3 OASIS3 and OASIS4 APIs

The major difference between the OASIS3 and OASIS4 APIs from a user point view is in the initialisation phase. In OASIS3, the grid information has to be provided in separate OASIS3 specific NetCDF grid files which are accessed by OASIS3 routines.

- 5 With OASIS4 it is assumed that each component has all knowledge about its own geographical grids at some point during the initialisation phase. This grid information has to be provided through the API to initialise the PSMILe search.

- Beside this difference, we have kept the design of the OASIS3 and OASIS4 PSMILe API as similar as possible, both following in particular the principle of “end-point” data exchange explained above, in order to allow an easy transition from OASIS3 to OASIS4 for the community of users.
- 10

5 Neighbourhood search

- The PSMILe library performs the exchanges of coupling data between source and target components. Usually those components express their coupling fields on different numerical grids and a regridding operation has to be performed on the coupling data. The regridding implies: 1-identifying the “neighbours” of each target point, i.e. the source grid points that will contribute to the calculation of the target grid point value in the regridding process, 2-calculating the weights of the different neighbours and 3-performing the calculation of the grid point values. To minimize the transfer of source data, it was decided to perform the neighbour search (1-) in the source PSMILe and to transfer only the useful source grid points to the Transformer which performs the regridding calculation per se (i.e. 2- and 3-). In this section, we describe the neighbourhood search performed by the source PSMILe.
- 15
- 20

- The goal here is to provide a highly efficient search algorithm which allows to determine neighbourhood relations between source and target grid points without generating significant overhead with respect to the model compute time. For ESMs the typical
- 25

GMDD

2, 797–843, 2009

**The OASIS4 software
for coupled climate
modelling**

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



elapsed time for a job is in the order of hours and the search should not take more time than a few seconds. At the same time the algorithm has to rely on only a very limited number of a-priori assumptions in order to be flexible and applicable to more than just a few special grid configurations.

5 Compared to typical physical algorithms for numerical models, the result or the output of a search algorithm is already well determined when the problem is posed, i.e. when the numerical grids are defined. For any given regridding scheme a unique solution exists regarding the neighbourhood between target and source points. When programming a library for such purposes, the task is to have flexible algorithms that are
10 able to deal with any peculiarities of a given grid configuration. While an application programmer typically has a complete control over the local grid, this is not the case for the library programmer as the library has to work based on the information provided through the user API without any or with only very limited a-priori assumptions.

In order to initialise the search, the components have to provide geographical information about the position of the grid points and about the reference volume or area that is associated to the grid points. This information is transferred as simple n-dimensional arrays depending on the grid type and it cannot provide information about discontinuities at specific grid points or alike. Nevertheless the correct neighbours have to be identified. In a simple implementation of such search algorithm, this task can be
20 achieved by comparing the distances for each target point with all source points available. This approach will be of the order of N^2 w.r.t. to the compute time (N being the number of target points to be processed, assuming that source and target grids are about the same size). While such an approach may still be justified for small problem sizes, this approach will fail when grids become larger (e.g. the horizontal resolution increases).
25 For OASIS4, we have chosen a hierarchical approach in order to minimise the workload during the search. We briefly describe here key aspects of the search algorithm which ensure its efficiency. An additional difficulty arises with parallelised components where the source grid is partitioned. For selected target points, the required "neighbour" source locations may reside on different processes and the search

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



has to be extended across process boundaries. The algorithm that is described below characterises what is happening behind the scene inside the PRISM_Enddef call.

5.1 Point-based search for block-structured grids

The completion of the definition of grids and coupling fields is announced in the code with a call to PRISM_Enddef (end of definition) on each process. At this stage, the coupling information from the XML configuration files about which components have to exchange data with each other is already available. In an initial step, envelopes of the locally defined grids are determined for all coupled components. The envelopes are exchanged between those component processes that have to exchange data with each other. Each component process now has a global view on grids and processes with which it may have a coupling interface. As a first step towards an efficient search, regions outside the envelop intersections between source and target grids are cut off.

The task of the second stage of the search is to find a reference source point for each target point which can later be used as a starting point to build the required interpolation star, i.e. the set of source neighbour points, or to locate additional neighbours based on the user choice.

Lists of target points are transferred to those source processes for which an overlap has been determined, and a grid hierarchy is established out of the source grid on the source side. For block-structured grids, a coarse grid is easily constructed by sub-sampling the grid points of the source grid. A grid hierarchy is constructed with a refinement factor of two similar to a multi-grid approach until the finest level is reached (see Fig. 2). Bounding boxes are constructed by simply evaluating the minimum and maximum cell extent in a given i-j-k index range. Due to this simple approach, neighbouring bounding boxes may overlap. If some target point is located in such an overlap region, the algorithm is prepared to investigate the neighbouring boxes until it is able to continue going up the level hierarchy onto the next finer level. This multi-grid construct is now used to determine the source cell (defined by the vertices of an area or volume element) that contains the projection of a given target point. From this stage onwards

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



the following steps are only local operations in the grid point space while the costly search on the global domain is already completed with the multi-grid algorithm.

5 A great advantage of the multi-grid algorithm is that it is only weakly dependent on the problem size. When the grid size is doubled in each horizontal direction this would only introduce one more multi-grid level. For very small problem sizes one may argue that the multi-grid algorithm creates too much overhead and a simple search would be more efficient. For larger grids however, the overhead to set up the multi-grid hierarchy is more than compensated by the speedy search and even allows to handle very large problem sizes (e.g. a global grid with $1/12^\circ$ horizontal resolution) in a reasonable amount of time. We postpone this discussion to Sect. 8.

10 In an intermediate step, the source points are now used to construct an auxiliary cell grid where the source points serve as corners of the cells (see dashed and solid lines in Fig. 3). The next task is to locate the source-point cell which contains the projection of the respective target point. This is performed with a local search by investigating each of the four source-point cells that surround the point which belongs to the marked source cell. In this example, the source-point box built with points at indices $[i,j]$, $[i+1,j]$, $[i,j+1]$, and $[i+1,j+1]$ is chosen and the source-cell of the point $[i+1,j+1]$ (green box in Fig. 3) is found. This immediately directs us to identify the point with index $[i,j]$ as the “lower-left” source-point. This location is now stored as the reference location for the particular target point. In a third step, the remaining neighbours are identified which are required to perform the interpolation requested by the user. For a bilinear interpolation, we are now able to identify the four corner points of the auxiliary cell (black open and solid bullets connected by solid lines) as the four neighbour source points needed. For a bicubic interpolation, sixteen neighbours have to be identified (the aforementioned bilinear points plus the open blue circles shown in Fig. 3).

25 Typically, block-structured grids include masked points in their compute domain. The algorithm as it is described so far does not distinguish between masked and non-masked points but solely works on all grid coordinates in the source grid compute domain. For some target points, it may be that not all source points first identified as

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



neighbour points are available for the interpolation as they are masked out and the physical fields do not necessarily contain meaningful values on those points. The user API provides different choices to the user via the XML file in such cases. As a default option, the remaining valid points will be used for a weighted-distance interpolation.

- 5 As an alternative, the choice is provided to not interpolate onto such target points. In the extreme case when all neighbour source points identified by the default search are masked, OASIS4 offers the possibility to search for the non-masked nearest-neighbour value and use this value for the target point. For this extra nearest-neighbour search, the algorithm starts from the original masked source grid point taken as an initial guess.
- 10 Starting from the finest level containing the initial guess, the multi-grid hierarchy is then used again to locate the closest non-masked source points by moving in so-called w-cycles through the multi-grid hierarchy, that is going down to a coarser level and going up again into another branch of the hierarchy repeatedly until the closed point is found.

5.2 Search on domain-partitioned source grids

- 15 On modern parallel systems, the application domain is in general distributed to many processes and/or processors. If the interpolation of fields is performed using locally available data only, i.e. data which is available on the actual process, the result of the interpolation may depend on the actual partitioning. For example, if a target point is located close to the internal border of a partitioned domain of the source application,
- 20 the standard interpolation stencil may require source points which are not located on the actual process but which are located on a neighbour process. In such a case, the OASIS4 coupler provides the possibility to use only locally available data or to use also the data of other process domains for the interpolation. In the first case, the interpolation result (and thus the result of the entire simulation) will depend on the partition of application domains; in the second case, the interpolation result is independent of the
- 25 partitioning. In the OASIS4 context, the latter approach is called global search.

In order to support the interpolation based on the “global” search, the data for source points located in the domains on remote processes must be provided together with lo-

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



cal interpolation data to the OASIS4 coupler. Since the definition of application grids is quite general and since the full connectivity information for partitioned grids is not provided to the OASIS4 library (and is not explicitly available in most of today's climate model component codes), missing source points for interpolation stars require an additional search step. In this additional search step, the missing source points for the interpolation are searched in neighboured domains. For block-structured grids, neighbouring domains are identified by exchanging information about the global index space which is derived using the information provided through the PRISM_Def.Partition function call. If these source points are found, the full information including the mask data are returned to the process which has initiated this additional search step. This process collects the data from possibly different processes and forwards the data for the full interpolation to the coupler.

5.3 Point-based search for Gauss-reduced grids

Gauss-reduced grids have been introduced mainly for atmospheric models in particular the ARPEGE and IFS models (see e.g. Déqué et al., 1994) to overcome problems resulting from the convergence of meridians close to the geographical poles. As these models and grids play an important role for the atmospheric modelling community, the decision has been made to support this type of grid as a special case. These grids cannot be classified as block-structured grids, nor are they completely unstructured. So instead of treating them as unstructured grids, we are in this particular case able to make use of some implicit knowledge about how the grid is constructed and introduce an intermediate step. In order to reuse as many of the existing functions as possible, we set up an block-structured auxiliary grid which is regular in longitude and latitude and determine the corresponding mapping of the Gauss-reduced grid points onto this auxiliary grid. We are now able to perform the initial search on the auxiliary grid and derive initial locations. By mapping them back onto the Gauss-reduced grid, we obtain the correct start location to continue the local search on the Gauss-reduced source grid in a similar fashion as it has already been described for block-structured grids. In contrast

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



to block-structured grids, the connectivity cannot be derived from the i - j indices directly. Instead we have to determine the connectivity among the grid points stored in an 1-D array in order to proceed with the local search; this is done based on the description of the Gauss-reduced grid provided by calling the routine PRISM_Reducedgrid_Map in the component code.

5.4 Cell-based search

Compared to the search described above for point-based interpolation schemes, the cell-based search has to follow a slightly different approach. In order to determine all necessary information for conservative remapping, the task here is to locate for each target cell its overlapping source cells. In order to again reuse as many functionality as possible, we perform a point-based search on the corner coordinates in a very similar way to the search described above up to the point where we have obtained the start location. These indices still refer to the data structure of the corner points; with block-structured grids, they can easily be mapped to cell indices, with each cell e.g. described by its four corners. For each target cell (see Fig. 4), we can now use this initial source cell to continue the local search and identify the remaining source cells which overlap a given target cell. The intersection or overlap between a pair of source and target cells is determined by a pairwise investigation of the intersection of the edges without a detailed evaluation of the exact area of overlap. The numbering shown in Fig. 4 indicates the order in which the local search is performed on the source grid. In this particular case 21 cells are investigated starting with the initial source cell 1. For each of the source cells the overlap with the target cell is checked. Source cells with no overlap are marked as visited. Source cells that overlap with the target cell are marked and stored in memory. For an individual local search, the path way through the source cells is not fixed but dependent on the local problem.

[Title Page](#)

[Abstract](#)

[Introduction](#)

[Conclusions](#)

[References](#)

[Tables](#)

[Figures](#)

[◀](#)

[▶](#)

[◀](#)

[▶](#)

[Back](#)

[Close](#)

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)



5.5 Optimisation

Most grids used in Earth system modelling are block-structured grids. Historically, codes have been developed with non-global grids in mind and more important, for stand-alone models. Particular issues regarding the connectivity are handled internal to the code to the extend it is needed to provide all information for the advection and diffusion operators and other boundary conditions. For global grids so-called cyclic boundary conditions have been introduced to guarantee a seamless continuation of the grid in the zonal direction. When approaching the geographical poles additional ghost-cells are introduced in order to properly calculate the operators along the zonal boundaries (see e.g. Madec and Imbard, 1996). When considering the i - j domain in the case of coupling, a remote cell close to the pole may extent further than what is covered by the ghost-cell in the i - j space. As the codes lack the information about the connectivity at the poles in general an additional search in the geographical domain is necessary. If the application is able to provide information about the connectivity in an explicit way, a new interface routine could be added to the API to transfer it to the PSMIIe. This will help to increase the efficiency of the search locally and across process boundaries beyond what is achieved already today.

The OASIS4 PSMIIe search algorithms are designed to work on 3-dimensional model domains. In general, one could think about treating all grids as 3-dimensional fully irregular or unstructured grids. In this way one would ignore important a-priori information useful to speed up the search process. In order to take advantage of the existing a-priori knowledge (in particular about regularity along any of the grid axis) different grid types are introduced. The different vertical layers of grids with vertical z -coordinates are usually a simple vertical translation of the same 2-dimensional horizontal grid. In such cases, the search is split into two parts with a 2-dimensional search in the horizontal plane and a 1-dimensional search along the vertical axis. In the horizontal plane, if the different columns or rows of the grid are respectively just a longitudinal or latitudinal translation of one same column or row, the search can be con-

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



ducted by performing a 1-dimensional search along each axis which further reduces the compute time.

It is obvious that individual source cells or source points are needed for more than just one target location. In order to minimize the amount of data to be transferred and stored, redundant geographical information (usually of type double precision or real) is removed and an index list is created which points to the required source data for each target location. These lists (data and indices) are transferred to the Transformer process for further processing.

6 PSMILe-transformer interaction and regridding

As outlined further in Sect. 7, the PSMILe supports two different ways of exchanging data, directly between two components or through the Transformer processes when regridding is required. Here we concentrate on the latter case and describe the transfer of information between the PSMILe and the Transformer and the regridding per se done on the coupling fields.

As the final result of the PSMILe search described in the last section, each source process holds 1-dimensional lists about the source neighbour points required for the regridding of the target points encompassed by its local domain; the information relevant for the intersection of the local source grid domain with each target process grid domain is stored in a particular list. Along with the raw geographical information, index lists are created to guarantee a well defined and correct access to the data. In order to allow the scattering onto the final n-dimensional destination array during the data exchange, index lists are transmitted to the target process.

The lists containing the geographical information and the access information are equally distributed over the number of Transformer processes available, resulting in an effective parallelisation of the Transformer over these lists. The different Transformer processes can be assimilated to automates that loop over the reception of predefined arrays of integers sent by the PSMILe processes and react accordingly following a

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



well defined sequence of actions matching the actions on the PSMILe side. During the initialisation phase, the Transformer processes thereby receives and stores the different sets of lists containing the source and target information required to perform the regridding. During the exchange phase, each Transformer process receives from the source PSMILe the grid point field values (transferred from the source component code with a PRISM.Put call) corresponding to its lists, calculates the regridding weights if it is the first exchange, and applies the weights. The resulting target values are made available to the corresponding target PSMILe process. The data are sent upon request from the respective target process (i.e. when a PRISM.Get is called in the target component code).

The calculation of the weights depends on the regridding algorithm chosen by the user in the XML configuration file which can be different for each coupling field. The current version of OASIS4 supports purely 3-dimensional regridding (3-D n-neighbour distance-weighted average or trilinear interpolation) and 2-dimensional regridding in the horizontal plane (2-D n-neighbour distance-weighted average, bilinear, bicubic interpolation or conservative remapping) followed by a linear interpolation in the vertical. As in OASIS3, the 2-D algorithms are taken from the Spherical Coordinate Remapping and Interpolation Package (SCRIP) library (see e.g. Jones, 1999). In particular, for the 2-D conservative remapping, the contribution of each source cell is proportional to the fraction of the target cell it intersects; this ensure local conservation of extensive properties such as fluxes. The 3-D algorithms are 3-D extensions of the 2-D SCRIP algorithms. For a detailed description of the regridding the reader is therefore referred to Jones (1999) and to the SCRIP User's Guide (see Jones, 2001). As the SCRIP bicubic algorithm is based on the function gradient in the i and j directions, it cannot be applied for the Gauss-reduced grids. For those grids, the SCRIP functionality has therefore been extended by a standard bicubic algorithm based on the 16 source neighbours.

Analysing the quality of the regridding algorithm is a delicate issue and no firm number can be given as the results depend on the source and target grids and on the characteristics of the regridded field. As an indication, we show here the error ob-

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



tained when regridding the values of an analytical cosine bell function which presents one minimum and one maximum over the global Earth domain:

$$F_1 = 2 - \cos[\pi * \text{acos}(\cos(\theta)\cos(\phi))]$$

The relative error, i.e. the difference between the regridded value and the analytical value of the function divided by the analytical value, is calculated on each target grid point. The Figs. 5 to 8 present the results with each pixel showing exactly this relative error for each grid point. No projection in the latitude-longitude space has been performed to avoid any additional interpolation performed by the graphic package.

Figure 5 shows the relative error obtained with the bilinear interpolation from the LMDz grid to the ORCA2 grid. LMDZ (Hourdin et al., 2006) has a regular $3.75^\circ \times 2.535^\circ$ grid. ORCA2 (Madec, 2008) is based on a 2° Mercator mesh, (i.e. variation of meridian scale factor as cosine of the latitude); in the Northern Hemisphere the mesh has two poles so that the ratio of anisotropy is nearly one everywhere. The error is not shown for masked target points (grey areas). One can observe a relative error smaller than 0.4% over the whole domain except near the coastline, for example near the west coast of the northern South America, where it can reach 0.8% at maximum. For those points near the coastline, some source points that are identified as bilinear neighbours are masked out. The remaining (non-masked) valid points are used for a less precise weighted-distance interpolation resulting in a larger error.

Figure 6 shows the relative error obtained with the bicubic interpolation from a BT42 Gauss-reduced grid (with 6232 grid points in total distributed over 64 longitudinal circles) to the ORCA2 grid. The relative error in the equatorial and tropical regions of the basins is smaller than 0.4%, and growing slightly but not above 0.8% at higher or lower latitudes which corresponds to regions where the Gaussian grid is effectively reduced. Again, the error is larger near the coastline, reaching even about 4.2% for some points; as for the bilinear interpolation, only the non-masked source neighbours are used for those points near the coastline resulting in a less precise weighted-distance interpolation.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Figure 7 shows the relative error obtained with the conservative remapping from the ORCA2 grid to the LMDz grid. The relative error is everywhere smaller than 0.2% except for the last row of the LMDz grid close to the North pole and near the coastline where it can reach respectively 1% and 2.2%. The SCRIP library assumes that the border of a cell between two cell corners is linear in longitude and latitude: near the North pole, this assumption becomes less valid and this can explain the larger error. Near the coastline, different normalisation options are available in the SCRIP library to calculate the value of the target cells which partially overlap masked source cells (see Jones, 2001). We chose the option to normalize by the area of the target cell intersecting non-masked source cells; this option does not locally conserve extensive properties but results in more realistic target values even if the error is still relatively large. With the alternative option, the normalisation by the full area of the target cell, the relative error observed (not shown) can become as large as 100%.

We note here that as the data structure that is provided to the OASIS4 Transformer differs from the data structure sent to the OASIS3 Transformer, truncation errors may therefore lead to slight differences when comparing the results obtained with OASIS3 and with OASIS4 even though both software use the same regridding algorithms.

Finally, as an illustration of the benefits of the global search (see Sect. 5.2), Fig. 8 shows the relative error obtained with the bilinear interpolation from the LMDz grid to the ORCA2 grid when each component is partitioned over 3 processes (in the longitudinal direction) and when the global search is not activated. Comparing with Fig. 5, we observe greater relative error (of the order of 1% comparable to the maximum error obtained near the coastline) near the borders of the source partitions. This can be expected as the local search cannot, for those target points find the usual 4 surrounding neighbours needed for the bilinear interpolation (as some of those 4 are in fact located on a neighbour process domain) and the remaining neighbours are used for a less precise weighted-distance interpolation resulting in a larger error. When the interpolation is performed with the global search described in Sect. 5.2, this greater relative error vanishes and regridded results (not shown) are exactly the same in the partitioned and

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



not partitioned cases, which validates the global search.

7 Data exchange

The exchange of coupling data is typically happening inside a loop over time or iterations and is invoked many times during the execution of the coupled model. Therefore the handling of the data exchange is a time-critical task. One of the major targets is to minimise the communication overhead and to avoid load-balancing problems induced by the implementation of the exchange.

The data exchange is invoked by the component models using the PRISM_Put and PRISM_Get interface routines. As described in Sect. 4.2, the routines can be used to read from and write to NetCDF files as well as sending data to and receiving data from another model component. The arguments of the subroutine calls are identical for IO and coupling operations, and the concrete action(s) are only specified in the component XML configuration file.

Figure 9 schematically illustrates possible communication pathways in a coupled application including the access of data from files. Depending on the respective grid configuration, the exchange of coupling data is handled in two different ways. Data points that reside on identical geographical locations on the source and target side are exchanged directly between the component processes (involving repartitioning if necessary). Data points for which a regridding is required are sent through the Transformer. Matching and non-matching grid points are identified by the PSMILe solely based on the geographical description of the grid. In order to reduce the amount of messages to be exchanged, the current implementation assumes that the number of matching points must at least sum up to 10% of the total number of grid points. For any number below this threshold, matching data are exchanged together with non-matching data through the Transformer processes. To further optimise the exchange of data and to reduce the work load of the Transformer processes, only non-masked source and target points are transferred. On the receiver side, the non-masked data are scattered

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



onto the destination array behind the PRISM_Get call and provided to the API in the appropriate array shape leaving masked target indices untouched. The calculation of regridding weights in the Transformer is delayed until the first exchange for which a particular set of weights is required. The weights are then stored in the Transformer for subsequent exchanges.

It noteworthy to mention that the data are buffered internally on the sender side and transferred with a so-called non-blocking MPI function (MPI_Isend). This guarantees that the PRISM_Put will immediately return when the data are stored locally even if the target component (be it the Transformer or another component in case of direct communication) is not yet ready to receive data. On the sender side the user code outside the PSMILe can safely progress and reuse its own local memory. The PRISM_Get on the other hand will only return when the data are received.

The IO layer is based on the mpp_io package as described by Balaji (2001). For the OASIS4 PSMILe IO, various modes are supported which become especially relevant if components are parallelised. The default behaviour of the PSMILe is, for output, to collect all data on the component root process and write out data into a single file or, for input, to read data on the component root process and distribute it internally. While this approach is still applicable when the domain partitioning is changing from job to job (especially when thinking about coupling restart files) it may not be the most performant way to deal with the IO. As an alternative approach, all IO processes read data from and write data to individual files. For further analysis these files have to be glued together in a postprocess. In order to support a full parallel IO, the mpp_io package has been extended to be compliant with the pNetCDF library (Li et al., 2003). When linked against pNetCDF, data are written into and read from a single file via MPI-IO with no pre- or postprocessing required. From a user perspective, this approach is invariant to changes in domain partitioning and should also provide a reasonable IO performance provided that the respective MPI-IO implementation efficiently uses the underlying system architecture (memory layout and file system).

During the data exchange and IO, the PSMILe is able to invoke some local trans-

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



formation routines like time-accumulation, time-averaging, gathering and scattering of physical fields and performs the required transformation locally on the source side before the exchange with other components of the coupled system or IO; only the end product is then communicated to the target.

8 Performance

Regarding the general performance of the coupling software from a user point of view it is tempting to ask for an exact number concerning the cost of the coupling and the overhead the user has to pay in terms of CPU seconds or a percentage of time. For full ESMs, it is difficult to predict the additional time needed for the coupling. To some degree, this depends on the search that has to be performed on a variety of grids for a larger number of exchange fields and different regriddings. Another important aspect is the run-time behaviour of the different ESM components and the load-(im)balancing between the processes involved. This may strongly influence the time needed to perform a data exchange. ESMs differ from each other in the above mentioned aspects, and providing a number for a particular ESM is of little help and not in the scope of this article. In addition, the picture may again change completely when an ESM job is running in batch mode and eventually has to share some resource with competing applications. In the following, our aim is to give a first idea about the potential cost for coupling and to provide information about the efficiency of the coupling software as such. Therefore we try to exclude the aforementioned side effects as much as possible. We will revisit this subject in our discussion in Sect. 9. We now use a simplified test case application to demonstrate the performance of the OASIS4 search algorithm and the data exchange.

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



8.1 Benchmark design

Here we investigate the so-called “strong scaling” properties of OASIS4. By keeping the global problem size constant, we expect a decrease in CPU time for individual processes with the problem distributed to larger number of processes in proportion to the local problem size. While this is in general reflecting the typical situation the user is confronted with, strong scaling is considered to be more difficult to achieve (in contrast to weak scaling where the local problem size is kept constant and the global problem to be solved is increased with increasing numbers of processes used). The test case we are considering now is a bidirectional exchange of 2-dimensional data between a global T255 “atmospheric” grid for component A and an “ocean” grid with $0.3^\circ \times \cos \phi$ horizontal resolution for component B, ranging from 0° to 360° in longitude and latitudes ϕ between 70°S and 70°N . This corresponds to 1202×665 grid points for component B and 768×385 grid points for component A. Both grids are described as horizontally irregular grids in order to bypass the optimised routines that are available for completely regular grids and to mimic the large class of use cases, the 2-dimensional data exchange between any two physical components at the Earth surface. Some of the ocean and atmosphere grid points have been masked out in order to bypass additional optimisation steps of the search algorithm which can be applied if complete non-masked grids are treated. The search is performed for bilinear interpolation without requesting an extra nearest-neighbour search for target points that only have masked “bilinear” source points (see Sect. 5.1). The search is performed for one exchange field for each direction. In an additional test, we have expanded the search problem into the vertical dimension towards a full 3-D-dimensional search and data exchange with 40 levels for component A and 45 levels for component B.

For this particular benchmark calculations, the OASIS4 sources have been compiled with the Intel Fortran compiler version 10.1 and the GNU C compiler version 4.1.2, both using default compiler switches without any further optimisation. The code has been run on a local PC cluster. Each node of the cluster is equipped with 2.0 Gi-

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



gahertz 2x single core AMD Opteron processor 246 and 4 Gigabyte of memory per node connected via a 2 Gigabit Myrinet. For the message passing, we use the Message Passing Interface Chameleon Glenn's Messages (GM) proprietary communication layer (MPICH-GM) provided by Myricom. For all tests, we have used the MPI1 method where all processes have to be started at once (see Sect. 3).

8.2 2-dimensional search

The numbers presented in Table 1 show the wall-clock time needed to perform the search for the bilinear interpolation, the time needed to perform the first exchange of data (which includes the calculation of the weights for the bilinear interpolation in the Transformer), and the time needed for subsequent data exchanges (averaged over 100 data exchanges). Although the benchmark has been run on a dedicated system, the numbers in the table only represent snapshots and are subject to slight variance caused by conflicts with system processes. Nevertheless we have not performed any ensemble means with appropriate statistics as this will not change the general picture we are going to discuss.

The time needed for the search can be represented by the time needed for the PRISM_Enddef routine. The search is performed on the two sets of source component processes as the component A target points are searched for on the component B processes and vice versa. As the PRISM_Enddef and subsequent calls therein contain blocking MPI calls, the total time needed by the PRISM_Enddef is to a large extent controlled by the source process which has to solve the largest problem (mainly determined by the number of target points to be processed). The time spent in PRISM_Enddef is almost identical for all processes due to this blocking nature of the MPI function calls used inside. Therefore only one number, the time needed by the component process with the largest data load is shown in the table for each setting. For the first and subsequent data exchanges, the time is provided for the slowest component process, separately for A and B .

In general, the wall clock time needed for the search decreases with the number of

The OASIS4 software
for coupled climate
modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



processes dedicated to the components: in this case, the speed-up is roughly a factor of 7 when going from 1 to 16 processes. When going from one to two component processes, the time for the search is slightly increased. The main reason can be attributed to the different algorithms used, as in the one-processor case certain subroutines do not have to be called simply because no boundary exchange is required on the source grid. When the number of processes is increased further the required wall-clock time is reduced significantly to values well below 1 s for the 2-D data exchange on 16 processors. At this stage, the number of Transformer processes is not of critical importance as they are not involved in the search as it is only at the very end of the search cycle that results are delivered to the Transformer process(es) (see Sect. 6).

8.3 2-dimensional data exchange

In this example, a data exchange can be understood as one pair of PRISM.Put and PRISM.Get operations in each component (referred to as ping-pong). The time needed for a complete ping-pong usually is different on each process. Load-balancing problems provide only one answer to explain this behaviour, and is beyond the control of the PSMILe. Furthermore, it may happen that one instance of the coupled system may have to receive more than just one message at a time. This is e.g. the case if only one Transformer process is available and two source components are trying to send or receive data at the same time. In this case, the messages can only be processed one after the other by the Transformer causing other component processes to wait. As only non-masked data are exchanged between the components, data are subsampled prior to the send operation and have to be scattered on the receiver side before they are delivered to the user space. This requires additional operations which are also captured by our time measurements for the ping-pong. In order to provide only a single number for the ping-pong, we show the time needed by the slowest component processor to complete the exchange.

The numbers in Table 1 show a general decrease of the wall-clock time for the exchange with an increasing number of Transformer processes available. One reason

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



is that the likelihood of the Transformer process being ready to receive, process and send data increases. The other reason is that the amount of data to be treated is reduced for each process. The only case when increasing the number of Transformer processes does not result in a decrease of the exchange wall-clock time is when the components themselves are not parallel and when the number of processes for the Transformer goes from 2 to 4. In this case, as the components are not parallel, there are only two distinct lists of source neighbour points required for the regridding of target points, i.e. one when considering component A as the target and one when considering component B as the target. As the Transformer is parallelised over these lists (see Sect. 6), the 2 additional processes (when going from 2 to 4 processes) do not get any workload and do not participate in the data exchange.

During the first exchange of data in addition to the processing of the exchange fields, the Transformer has to calculate the weights which are then stored and reused for subsequent data exchanges. This additional workload is responsible for the longer time needed to perform the first exchange when comparing with the time needed for subsequent exchanges.

It can also be observed that for a fixed number of Transformer processes the exchange time is fairly independent of the number of application processes. As the data are exchanged through the Transformer processes, it can act as a bottleneck in the exchange if it is not sufficiently parallel. In general, it is the number of available Transformer processes that dictates the speed of the exchange and not the parallelisation of the components themselves.

Finally, one can also observe that the scaling behaviour is in general not fully predictable; this can mainly be attributed to the fact that a particular Transformer process may still be busy with other tasks, that data are sometimes not available from the source component or delivered only with a delay due to punctual load imbalancing. On the other hand, cache effects can cause super-scalability when going in our case from 2 to 4 processes for the components.

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



8.4 3-dimensional data exchange

The general picture does not change much with an increasing amount of data to be exchanged, e.g. going from the 2-dimensional to a 3-dimensional coupling, i.e. when exchanging data between 40 levels for component A and 45 levels for component B (see Table 2). As in the 2-dimensional case, the efficiency of the search mainly depends on the number of component processes available. The wall-clock time decreases by roughly a factor of 4 to 5 when going from 1 to 16 component processes. The time needed for the search decreases to only a few seconds when employing four or more application processes – which is likely the case with applications running with the grid sizes similar to our example. With an increasing number of Transformer processes the exchange time is reduced as in the 2-dimensional case.

8.5 Discussion

As we tried to outline above, it is difficult to provide an exact prediction for the time needed by the search algorithm. Apart from the quality of the hardware involved, results depend on a variety of other parameters defined by the application or the user. The run-time behaviour (or load-balancing) of the physical components involved is one important aspect. Although the coupling software can try to hide as much as possible communication with computation, it cannot completely cancel out load-balancing problems. The fact that component A needs data in order to progress with its calculation and that application B has not yet delivered the data is totally independent of the coupling software functionality or optimisation. An external observer will notice that component A or the Transformer is eventually spending CPU time in some MPI function call waiting for data. The additional time needed for the coupling when compared against the run time of a stand-alone component run cannot necessarily be attributed to an inefficient coupling software (or MPI implementation) but may solely be attributed to a inefficient load-balancing between the physical components involved.

Apart from those problems mentioned above, there are some more predictable pa-

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



rameters which influence the additional time needed for the coupling. Clearly, the time will depend on the number of exchange fields to be processed. If each field is defined on a separate grid a complete new search has to be performed which will consequently lead to a higher demand of CPU time. If different exchange fields are defined on an identical grid, the PSMILe is able to reuse previous results and the required time will only slightly increase when compared against the single field search. If no masks are applied at different vertical levels, the PSMILe is able to perform some optimisation which will again decrease the work load.

9 Conclusions and future perspectives

In this article, we have described the new OASIS4 coupling software in its technical details. The new parallel multi-grid search algorithm has been introduced. By using first simple coupled test configurations, the applicability of the software has been demonstrated. A remarkable feature of the new software is the rather efficient search algorithm. The compute time measured to perform the search provides confidence in the efficiency of the search algorithm. Apart from using OASIS4 for coupled models, it is also well suited to serve as a pre- or post-processing software to perform less time-critical regridding tasks. For coupled applications with reasonable process-local problem sizes of the order of 10.000 grid points in the horizontal, the time needed for the search is only up to the order of a second which is considered to be negligible when compared with the total runtime for a typical ESM job of several hours.

Up to now the focus of the development activities has been to provide the functionality which is required to make OASIS4 a useful tool for real applications. This first goal is now achieved, an indication for that being that OASIS4 is currently actively used for atmosphere dynamics to atmospheric chemistry coupling by the Global and regional Earth-system (Atmosphere) Monitoring using Satellite and in-situ data (GEMS) project (Hollingsworth et al., 2008), and for the Regional Coupled Atmosphere and Ocean (RCAO) model at the Swedish Meteorological and Hydrological Institute (SMHI)

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Rosby Centre (Döscher et al., 2009). Also more and more groups start thinking about using the OASIS4 software within new projects, e.g. within the ScalES project recently funded by the Federal Ministry of Education and Research in Germany.

The current version of OASIS4 provides support for block-structured grids and Gauss-reduced grids. A next step forward is to support unstructured grids which we now see arriving at the horizon; this will be done in collaboration with the Alfred Wegener Institute for Polar and Marine Research (AWI) in the framework of the ScalES project. Other important developments are being or will be realised within the framework of funded projects, such as the InfraStructure for the European Network for Earth System Modelling (IS-ENES) supported by the 7th Framework Programme of the European Commission, in particular : (1) the realisation of a Graphical User Interface to help the user build the XML configuration files (see Sect. 3), (2) the possibility of using a pre-defined set of weights and addresses for the interpolation, and (3) general optimisation of the software. On a longer timescale, the support of adaptive grids or grids changing e.g. their land-sea mask is targeted; as we have outlined above, the efficiency of the search algorithm provides the potential to handle this aspect. Finally, work will continue to establish comprehensive services around OASIS4 through a portal offering documentation, user guides, tutorial, FAQs, user forum and tips for best practices, as such services are considered essential to fully support an active community of users.

Acknowledgements. We highly appreciated Reiner Vogelsang's (SGI, Germany) collaboration during the early times of OASIS4, and thank Damien Declat (now at Bull) and Thomas Schoenemeyer (NEC-HPCE) for contributing to the initial phase of the OASIS4 development. Part of this work has been funded within the EU FP5 PRISM project (Contract EVR1-CT2001-40012), the CICLE project supported by the French Agence Nationale de la Recherche (ANR, Contract ANR-05-GIGC-04), and the EU FP7 IS-ENES project (Contract GA No: 228203). The support by CERFACS, the NEC High Performance Marketing Division (Tokyo, Japan) and CNRS is gratefully acknowledged. Special thanks (in alphabetical order) go to Mats Bentsen (NERSC), Joseph Charles (UK MetOffice), Laure Coquard (CNRS), Ralf Döscher (SMHI), Johannes Flemming (ECMWF), Josephine Ghattas (now at IPSL), Ulf Hanson (SMHI), Ian Henderson (University Reading), Jean Latour, Kristian Mogensen (ECMWF), Marie-Pierre Moine (CER-

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



FACS) and Andreas Ripke (NEC) for using and testing earlier versions of the software and for their very helpful comments that allowed us to improve the coding.

References

- Balaji, V.: Parallel Numerical Kernels for Climate Models, in: ECMWF TeraComputing Workshop 2001, ECMWF, World Scientific Press, 2001. 807, 821
- Buja, L. and Craig, T.: Community Climate System Model CCSM2.0.1 User Guide, National Center of Atmospheric Research, Boulder CO, 2002. 801
- CLIVAR Scientific Steering Group: CLIVAR Initial Implementation Plan, vol. ICPO 14, International CLIVAR Project Office, SOC, Empress Dock, Southampton, UK, 1998. 799
- Déqué, M., Dreveton, C., Braun, A., and Cariolle, D.: The ARPEGE/IFS Atmosphere Model: A Contribution to the French Community Climate Modelling, *Clim. Dynam.*, 10, 1994. 813
- Döscher, R., Willén, U., Jones, C., Rutgersson, A., Meier, H. E. M., Hansson, U., and Graham, L. P.: The development of the Coupled Regional Ocean-Atmosphere Model RCAO, *Boreal Environ. Res.*, 7, 183–192, 2002. 800
- Döscher, R., Wyser, K., Meier, H., Qian, M., and Redler, R.: Quantifying Arctic Contributions to Climate Predictability in a Regional Coupled Ocean-Ice-Atmosphere Model, *Clim. Dynam.*, in press, doi:10.1007/s00382-009-0567-y, 2009. 829
- Gropp, W., Huss-Lederman, S., Lumsdaine, A., Lusk, E., Nitzberg, B., Saphir, W., and Snir, M.: MPI – The Complete Reference, 2, The MPI Extensions, MIT Press, 1998. 803
- Hill, C., DeLuca, C., Balaji, V., Suarez, M., and da Silva, A.: Architecture of the Earth System Modeling Framework, *Compute Sci. Eng.*, 6, 2004. 801
- Hollingsworth, A., Engelen, R., Textor, C., Benedetti, A., Boucher, O., Chevallier, F., Dethof, A., Elbern, H., Eskes, H., Flemming, J., Granier, C., Kaiser, J., Morcette, J.-J., Rayner, P., Peuch, V.-H., Rouil, L., Schultz, M. G., Simmons, A., and The GEMS Cpnsortium: Toward a Monitoring and Forecasting System for Atmospheric Composition, *B. Am. Meteor. Soc.*, 89, 1147–1164, 2008. 828
- Hourdin, F., Musat, I., Bony, S., Braconnot, P., Codron, F., Dufresne, J.-L., Fairhead, L., Filiberti, M.-A., Friedlingstein, P., Grandpeix, J.-Y., Krinner, G., LeVan, P., Li, Z.-X., and Lott, F.: The LMDZ4 general circulation model: climate performance and sensitivity to parametrized physics with emphasis on tropical convection, <http://web.lmd.jussieu.fr/~lmdz>, 2006. 818

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



- Jones, P.: Conservative remapping: First- and second-order conservative remapping, *Mon. Weather Rev.*, 127, 2204–2210, 1999. 817
- Jones, P. W.: A User's Guide for SCRIP: A Spherical Coordinate Remapping and Interpolation Package, <http://climate.lanl.gov/Software/SCRIP/SCRIPUsers.pdf>, 2001. 817, 819
- 5 Joppich, W. and Kürschner, M.: MpCCI – a tool for the simulation of coupled applications, *Concurrency and Computation: Practice & Experience*, 18, 183–192, 2006. 801
- Li, J., Liao, W., Choudhary, A., Ross, R., Thakur, R., Gropp, W., Latham, R., Siegel, A., Gallagher, B., and Zingale, M.: Parallel netCDF: A Scientific High-Performance I/O Interface, in: SC2003 Proceedings, Supercomputing, 2003. 803, 821
- 10 Madec, G.: NEMO ocean engine, Note du Pole de modélisation 27, Institut Pierre-Simon Laplace, France, <http://www.nemo-ocean.eu/About-NEMO/Reference-manuals>, 2008. 818
- Madec, G. and Imbard, M.: A Global Ocean Mesh to Overcome the North Pole Singularity, *Clim. Dynam.*, 12, 381–388, 1996. 815
- Manabe, S. and Bryan, K.: Climate Calculations with a Combined Ocean-Atmosphere Model, *J. Atmos. Sci.*, 26, 786–789, 1969. 798
- 15 Rew, R. K. and Davis, G. P.: Unidata's netCDF Interface for Data Access: Status and Plans, in: Thirteenth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, Anaheim, California, American Meteorology Society, 1997. 803
- 20 Snir, M., Otto, S., Huss-Lederman, S., Walker, D., and Dongarra, J.: MPI – The Complete Reference, 1, The MPI Core, MIT Press, second edn., 1998. 803
- Valcke, S. and Redler, R.: OASIS4 User Guide, PRISM Technical report No.4, 64 pp., <http://www.prism.enes.org/Publications/Reports/OASIS4.User.Guide.T4.pdf>, 2006. 805
- Valcke, S., Budich, R., Carter, M., Guilyardi, E., Foujols, M.-A., Lautenschlager, M., Redler, R., Steenman-Clark, L., and Wedi, N.: A European Network for Earth System Modelling, *EOS*, 88, 143, 2007. 801
- 25

GMDD

2, 797–843, 2009

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Table 1. Performance of OASIS4 tasks obtained on a PC cluster for 2-dimensional search, 2× Single Core AMD Opteron and MPICH-GM message passing with a Myrinet-2000 network. For the different tasks the maximum time of the respective subset of processes is provided.

number of processes for			time in seconds for				
Driver	atmos	ocean	search	atm 1st exchn	ocn 1st exchn	atm nth exchn	ocn nth exchn
1	1	1	6.049	0.224	0.248	0.077	0.101
2	1	1	6.050	0.087	0.167	0.046	0.064
4	1	1	6.062	0.089	0.167	0.046	0.064
1	2	2	6.487	0.241	0.237	0.104	0.097
2	2	2	6.484	0.122	0.124	0.052	0.054
4	2	2	6.468	0.114	0.120	0.043	0.049
8	2	2	6.486	0.049	0.101	0.041	0.032
1	4	4	2.425	0.254	0.257	0.075	0.083
2	4	4	2.385	0.195	0.183	0.073	0.078
4	4	4	2.412	0.093	0.115	0.033	0.043
1	8	8	1.174	0.234	0.238	0.092	0.090
2	8	8	1.172	0.147	0.152	0.045	0.049
4	8	8	1.179	0.097	0.097	0.037	0.035
1	16	16	0.826	0.226	0.225	0.113	0.114
2	16	16	0.779	0.137	0.136	0.061	0.065
4	16	16	0.791	0.092	0.091	0.041	0.042
8	16	16	0.771	0.042	0.045	0.018	0.025

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



The OASIS4 software for coupled climate modelling

R. Redler et al.

Table 2. As in Table 1 but for 3-dimensional search and exchange.

number of processes for			time in seconds for				
Driver	atmos	ocean	search	atm 1st exchn	ocn 1st exchn	atm nth exchn	ocn nth exchn
1	1	1	16.671	2.605	3.091	0.715	1.191
2	1	1	16.526	0.353	2.797	0.186	1.059
4	1	1	16.313	0.357	2.789	0.189	1.050
1	2	2	18.767	2.994	3.050	1.092	1.134
2	2	2	18.753	2.629	2.686	0.912	0.979
4	2	2	18.825	1.119	2.722	0.340	1.001
8	2	2	18.856	0.226	1.452	0.131	0.570
1	4	4	7.685	3.027	3.052	1.074	1.097
2	4	4	7.276	1.626	1.824	0.500	0.649
4	4	4	7.344	1.194	1.277	0.398	0.463
1	8	8	4.382	3.011	3.033	1.039	1.061
2	8	8	4.669	1.669	1.687	0.558	0.585
4	8	8	4.756	0.933	1.102	0.320	0.405
1	16	16	3.310	2.928	2.927	0.999	1.007
2	16	16	2.739	1.765	1.793	0.563	0.590
4	16	16	2.560	1.052	1.150	0.343	0.394
8	16	16	3.035	0.557	0.627	0.183	0.250

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Table A1. List of OASIS4 PSMILe function calls. Description of the OASIS4 PSMILe Function Calls.

Name	Description
Initialisation	
PRISM_Init	Initialisation of the application
PRISM_Init_comp	Initialisation of a component within an application
PRISM_Def_grid	Abstract definition of a compute grid
PRISM_Def_partition	Definition of the process-local grid partition relative to the global compute domain
PRISM_Reducedgrid_map	Additional information needed for Gauss-reduced grids
PRISM_Set_corners	Definition of geographical positions of vertices of grid cells
PRISM_Set_mask	Definition of masked points or cells
PRISM_Set_points	Definition of geographical positions of points
PRISM_Def_var	Definition of exchange and IO fields
PRISM_Enddef	End of the definition phase
Data Exchange	
PRISM_Get	To receive data from a remote component and/or read in from a file
PRISM_Put	To send data to a remote component and/or write into a file
PRISM_Put_restart	To write additional data sets into an OASIS4 coupling restart file
Termination	
PRISM_Terminate	Termination of the prism part
Auxiliary and Query Functions	
PRISM_Calc_newdate	Add and Subtract operations on the OASIS4 Fortran90 Time structure
PRISM_Get_localcomm	To retrieve a MPI communicator for component local communication
PRISM_Initialized	To check whether PRISM_Init has already been called
PRISM_Put_inquire	A query function to obtain information whether a PRISM_Put has to be posted
PRISM_Terminated	Query function to check whether PRISM_Terminate has already been called
PRISM_Abort	To terminate the coupled application in a well defined to support user-defined error handling

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



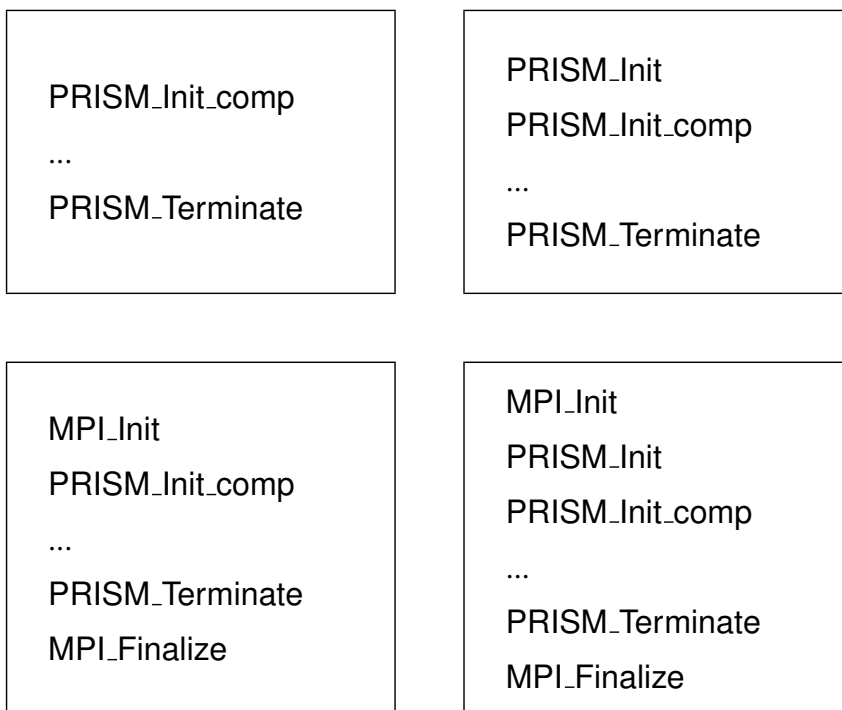


Fig. 1. The four allowed sequences of calls for initializing and terminating an ESM component for coupling with OASIS4 and its relation to MPI.

[Title Page](#)

[Abstract](#)

[Introduction](#)

[Conclusions](#)

[References](#)

[Tables](#)

[Figures](#)

[◀](#)

[▶](#)

[◀](#)

[▶](#)

[Back](#)

[Close](#)

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)

**The OASIS4 software
for coupled climate
modelling**

R. Redler et al.

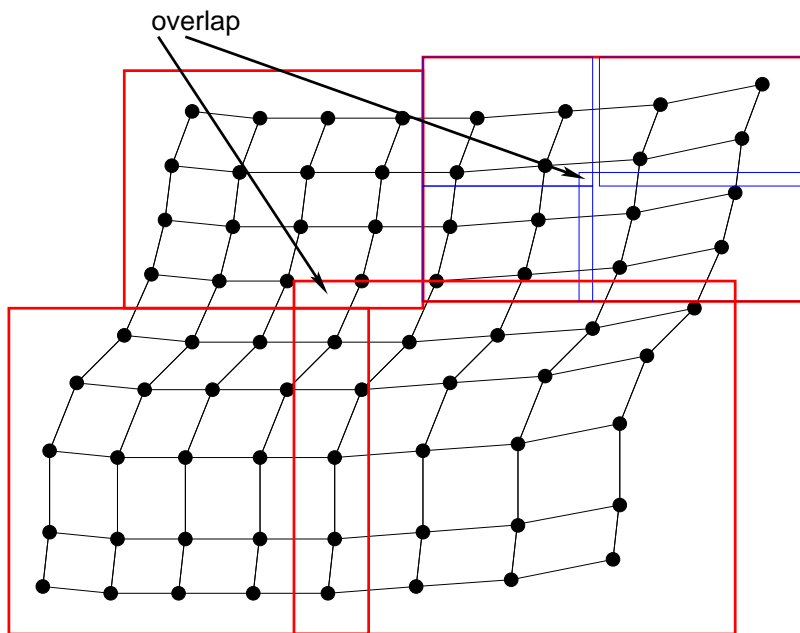


Fig. 2. Sketch of a multi-grid hierarchy for a local domain. Typically the number of cells is much higher and the PSMILe will construct $o(10)$ levels.

[Title Page](#)[Abstract](#)[Introduction](#)[Conclusions](#)[References](#)[Tables](#)[Figures](#)[◀](#)[▶](#)[◀](#)[▶](#)[Back](#)[Close](#)[Full Screen / Esc](#)[Printer-friendly Version](#)[Interactive Discussion](#)

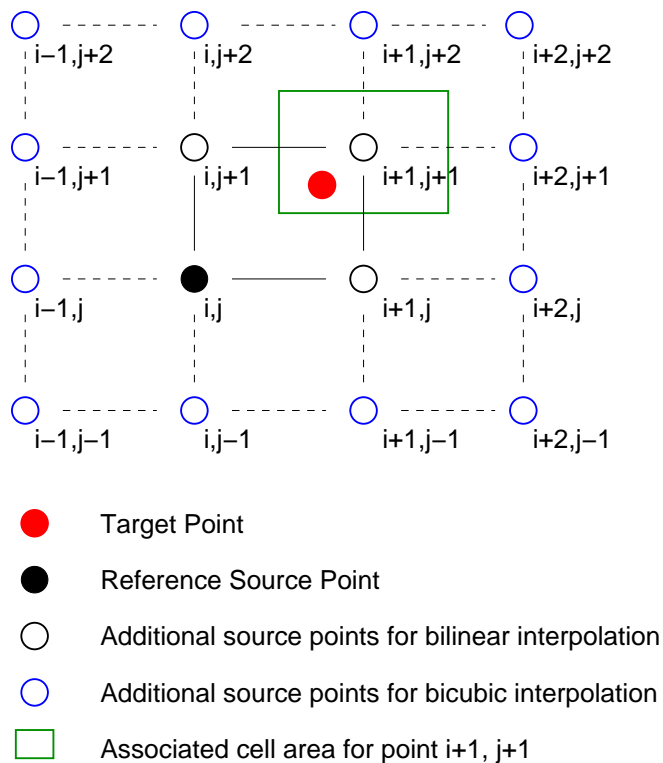


Fig. 3. Building the interpolation star based on the initial location.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

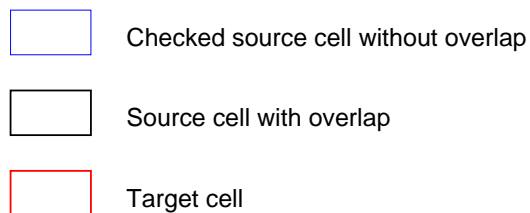
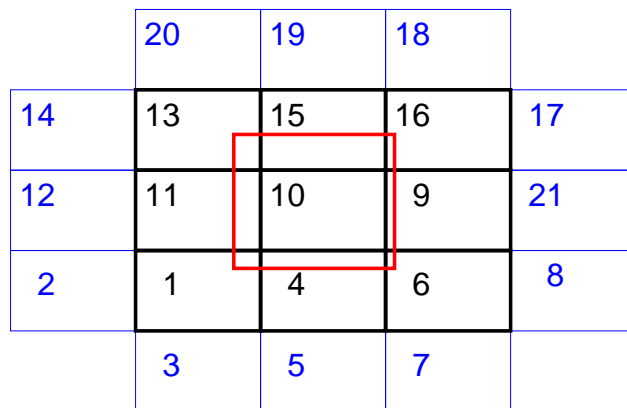


Fig. 4. Search of source cells for conservative remapping. Numbers indicate the sequence in which the source cell are investigated locally.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Fig. 5. Relative error obtained (in %) with the bilinear regridding from the LMDz grid to the ORCA2 grid.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Fig. 6. Relative error obtained (in %) with the bicubic regridding from the Gauss-reduced grid to the ORCA2 grid.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Fig. 7. Relative error obtained (in %) with the conservative remapping from the ORCA2 grid to the LMDz.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

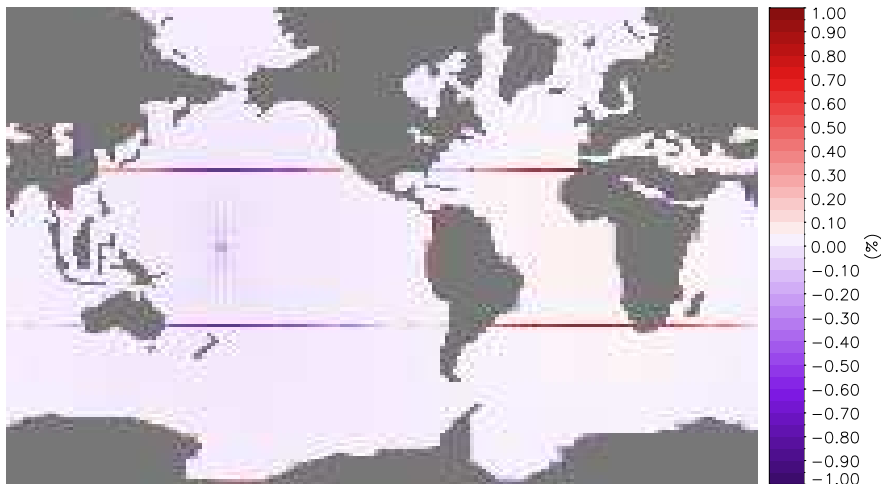


Fig. 8. Relative error obtained (in %) with the bilinear regridding from the LMDz grid to the ORCA2 grid when each component is partitioned over 3 processes and when the global search is not activated.

The OASIS4 software for coupled climate modelling

R. Redler et al.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



**The OASIS4 software
for coupled climate
modelling**

R. Redler et al.

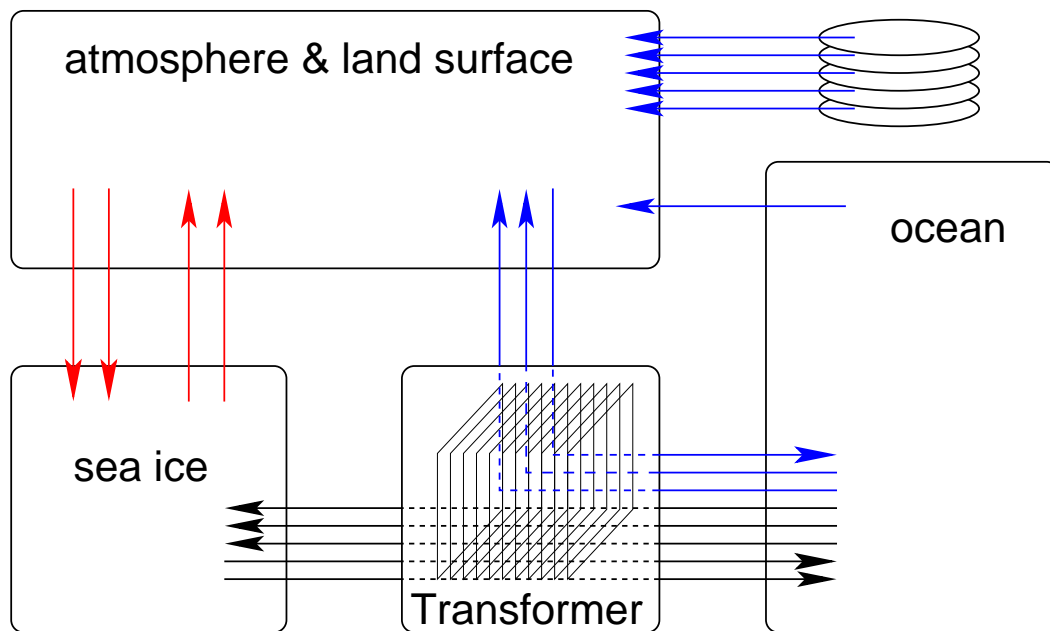


Fig. 9. Example of communication pathways in an OASIS4 coupled application.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion