Australian Government
Bureau of Meteorology

The Centre for Australian Weather and Climate Research
A partnership between CSIRO and the Bureau of Meteorology

CSIRO

# Australian Climate Ocean Model (AusCOM) Users Guide

Daohua Bi and Simon Marsland

**CAWCR Technical Report No. 027**

August 2010

www.cawcr.gov.au

# Australian Climate Ocean Model (AusCOM) Users Guide

Daohua Bi and Simon Marsland[1]

[1]*The Centre for Australian Weather and Climate Research - a partnership between CSIRO and the Bureau of Meteorology*

**CAWCR Technical Report No. 027**

August 2010

ISSN: 1835-9884

National Library of Australia Cataloguing-in-Publication entry

Author: Daohua Bi and Simon Marsland

Title: Australian Climate Ocean Model (AusCOM) Users Guide / Daohua Bi and Simon Marsland.

ISBN: 978-1-921605-92-5 (PDF)

Series: CAWCR technical report; 27.

Notes: Included bibliography references and index.

Subjects: Ocean-atmosphere interaction--Australia—Simulation methods.

Dewey Number: 551.5246

Enquiries should be addressed to:

Daohua Bi
Centre for Australian Weather & Climate Research
GPO Box 1289 Melbourne
Victoria 3001, Australia

dave.bi@csiro.au

## Copyright and Disclaimer

# CONTENTS

# LIST OF FIGURES



# LIST OF TABLES

# ABSTRACT

A user guide for the Australian Climate Ocean Model (AusCOM) is presented. AusCOM is the ocean and sea ice component of the Australian Community Climate and Earth System Simulator (ACCESS) model. AusCOM is an IPCC-class coupled ocean and sea ice climate model with its current configuration developed primarily within the Ocean and Coupled Modelling Team at the Centre for Australian Weather and Climate Research (CAWCR). It comprises the NOAA/Geophysical Fluid Dynamics Laboratory Modular Ocean Model version 4.1 (GFDL MOM4p1), the Los Alamos National Laboratory Sea Ice Model version 4.0 (LANL CICE4.0), and a data atmospheric model (MATM). Numerical coupling (i.e. data exchange between the sub-models) utilises the Message Passing Interface within the PRISM_2-5 OASIS3 coupler. A standard version is released as AusCOM1.0, an independent, self-contained, and easily setup package.

# 1 INTRODUCTION

The Australian Climate Ocean Model (AusCOM) is an IPCC-class coupled ocean and sea ice model aimed to serve the Australian climate sciences community (including the Bureau of Meteorology, CSIRO and the Australian universities) for ocean climate research and applications. It was conceived in discussions and informal workshops involving wide representation from the Australian ocean climate modelling community during 2003 and 2004, and initial prototype components were implemented at the Tasmanian Partnership for Advanced Computing (TPAC) (Roberts et al. 2007; Heil et al. 2005). The current configuration was developed at the Centre for Australian Weather and Climate Research (CAWCR, a partnership between the Commonwealth Scientific and Industrial Research Organisation and the Australian Bureau of Meteorology), primarily by staff in the Ocean and Coupled Modelling Team of the CAWCR Earth System Modelling Program.

AusCOM comprises the Geophysical Fluid Dynamics Laboratory Modular Ocean Model version 4.1 (GFDL MOM4p1), the Los Alamos National Laboratory Sea Ice Model version 4.0 (LANL CICE4.0), and a data atmospheric model (MATM). Numerical coupling (i.e. data exchange between the sub-models) is strictly constrained by the PRISM_2-5 OASIS3 coupler (hereafter refered to as the OASIS3.25 coupler, or simply OASIS3).

AusCOM is the core component for climate purposes of the Australian Community Climate and Earth System Simulator (ACCESS). Specifically, the ACCESS model is built by coupling the UK Met Office Unified Model (UM) atmosphere, and other sub-models as required, to AusCOM, via the OASIS3 coupling framework (see Fig. 1).

This document describes the AusCOM system framework, the coupling strategy, and the interfaces designed for using the OASIS3.25 coupler. Section 2 introduces the AusCOM framework, along with details of the horizontal and vertical discretisation in the default setup. The coupler, along with each of the component models, are briefly introduced in Section 3. AusCOM users are directed to the user guides provided by the developers of the MOM4 ocean code (Griffies et al. 2004) and the CICE sea ice code (Hunke and Lipscomb 2008) for more detailed scientific and technical descriptions of these ocean and sea ice component codes. Section 4 is for users who are keen to fully understand the coupling logic, re-mapping algorithms, and associated Message Passing Interface (MPI) technology used in the AusCOM system. It should be read in conjunction with the OASIS3.25 coupler user guide (Valcke 2006), the SCRIP package user guide (Jones 1997) and any MPI reference. Section 5 gives technical details of the AusCOM infrastructure, information for compilation of the system components, a description of various configuration files, and notes on the model run script. A quick start guide is provided in Section 6.

The CAWCR ACCESS Coupled and Ocean Modelling Team provides users with the AusCOM1.0 release, which is a compressed tar-file containing full packages of the four components, their associated auxiliary input files, and a few sample experiments ready for the user to start with. This AusCOM1.0 release package has been extensively tested on the National Computational Infrastructure (NCI) Sun Constellation cluster VAYU, the super-computing platform on which AusCOM has been developed. Several test cases are included, that utilise both climatological and interannually varying forcing datasets of Large and Yeager (2004), Large and Yeager (2008). A 500 year spin-up run benchmarks AusCOM against the models

used in the Coordinated Ocean-ice Reference Experiment (CORE) of Griffies et al. (2009), and is documented in our second technical report on the AusCOM development (Bi et al. 2010).

Users who want to spend as little time as possible in learning to set-up and run AusCOM1.0 at NCI, can skip directly to Section 5 for the technical guide, or Section 6 for the quick start shortcut. In fact, users who have access to the NCI VAYU disk /short/p66/ only need type in one simple command, wait for less than 25 minutes, then can run this independent, self-contained AusCOM1.0 system on VAYU.

# 2 AUSCOM FRAMEWORK AND THE OCEAN-SEA ICE HORIZONTAL GRID

The AusCOM ocean-sea ice coupled model is a four-executable system that consists of three sub-models (MATM, CICE, and MOM4) and a numerical coupler (OASIS3.25). Figure 1 shows the components of the ACCESS coupled model initial implementation and the coupling framework. The indicated core part AusCOM, when runing alone as AusCOM1.0, actually includes the data atmospheric model MATM in the place of UM, plus different coupling interface for CICE multi-layer sea ice physics configuration.

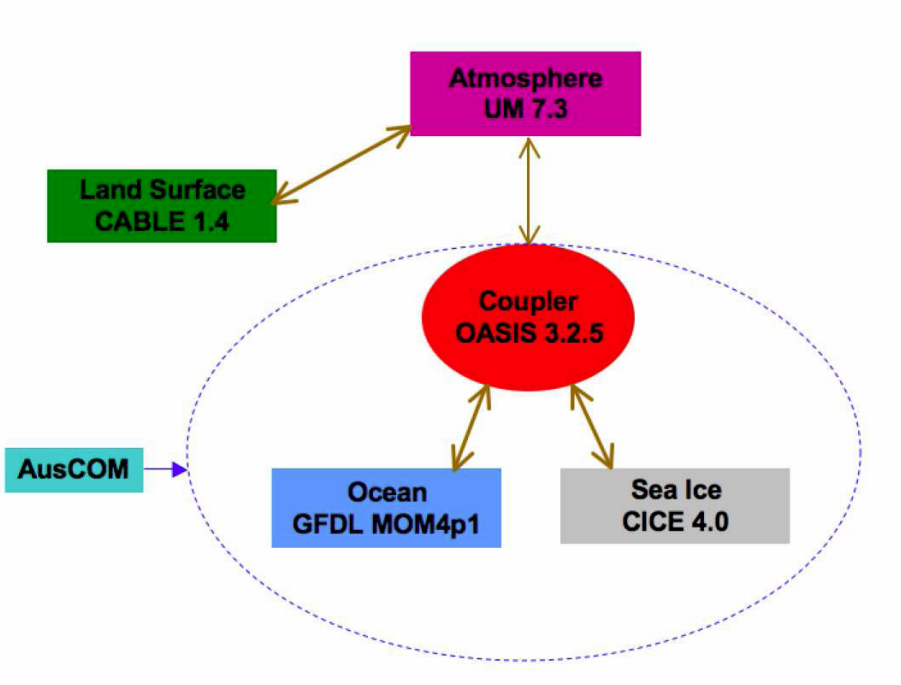


**Fig. 1** ACCESS/AusCOM Coupled Ocean and Sea Ice Model Framework and Components

Horizontally the AusCOM ocean model MOM4p1 is configured with an orthogonal curvilinear grid (Fig. 2). A singularity at the north pole is avoided by using a tripolar grid following Murray (1996). This approach also provides reasonably fine resolution in the Arctic Ocean, which enhances the computational efficiency and accuracy of the model. The AusCOM sea ice model CICE is configured to share this tripolar grid with MOM4. An important advantage of doing so in terms of ice-ocean coupling performance will be addressed below.

The orthogonal curvilinear spatial discretization is shown in Fig. 3. Along the curvilinear 'zonal' direction AusCOM has a regular spaced grid with 1° resolution. In the meridional direction the grid spacing is nominally 1° resolution, with the following three refinements:

- tripolar Arctic north of 65°north;
- equatorial refinement to ⅓° between 10°S and 10°N;
- and a Mercator (cosine dependent) implementation for the Southern Hemisphere, ranging from 0.25° at 78°S to 1° at 30°S.

For the AusCOM1.0 release documented here, MOM4p1 and CICE4.0 are horizontally configured on a global tripolar grid, with a 360x300 logically rectanglar horizontal mesh. This is a typical IPCC-class resolution. In the vertical direction, AusCOM1.0 implements the z* coordinate available in MOM4, with 50 model levels covering 0-6000 meters with a resolution ranging from 10 meters in the upper layers (0-200 meters) to about 333 meters for the abyssal ocean.

It is worth mentioning that the conventional z (height) coordinate models have previously represented free-surface variations by a variable thickness upper layer. The z*coordinate (Adcroft and Campin2004), however, rescales the height coordinate and treats the time-dependent free surface as a coordinate surface. The finite volume method (for discretizing the model) within the z* coordinate framework allows an accurate representation of topography by means of shaped volumes (shaved cells) or variable bottom layer thickness (partial cells) and has been demonstrated to overcome the inadequacies of height coordinates in representing topography. Additionally, the z* coordinate allows for more accurate treatment of sea ice in the model, by removing the problem of disappearing levels when the sea-ice thickness exceeds the thickness of the upper levels.

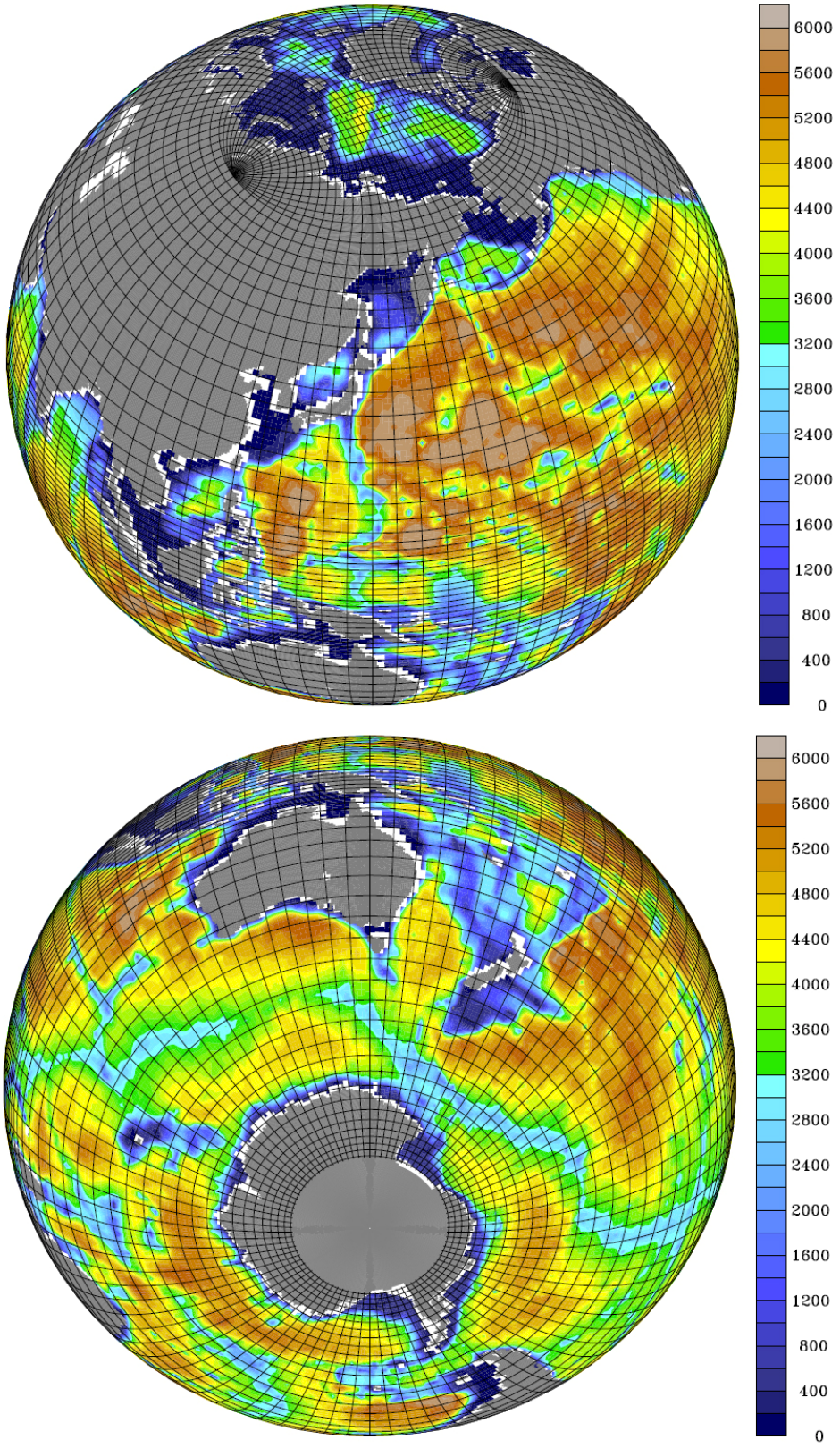


**Fig. 2** Bathymetry (m) for the AusCOM ocean and sea ice model. Grid lines indicate each fourth row in the orthogonal curvilinear 'zonal' and 'meridional' directions. Top: northern hemisphere projection showing the tripolar grid over the Arctic. Bottom: southern hemisphere projection showing the Mercator meridional grid over the Southern Ocean.

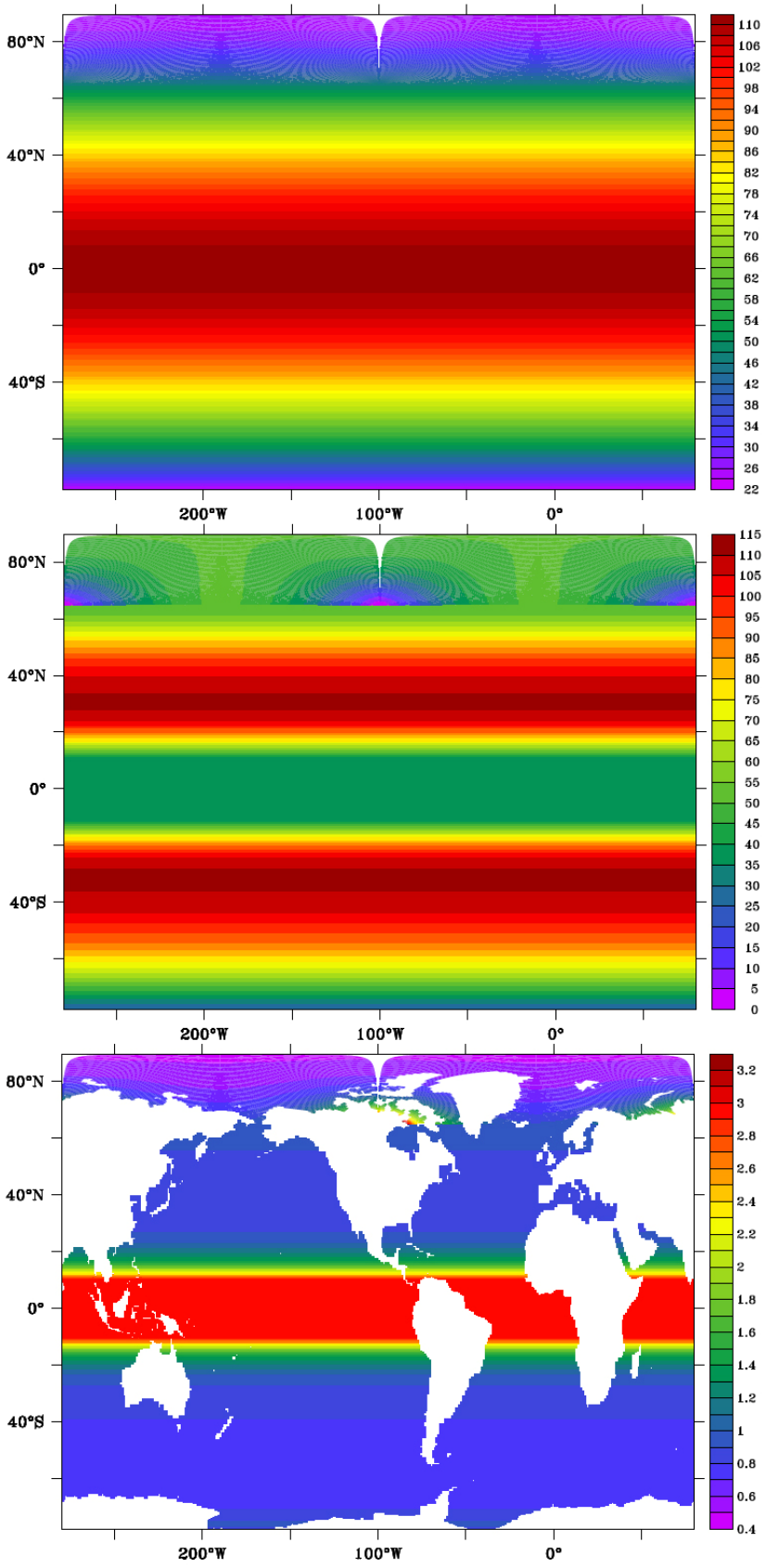


**Fig. 3** AusCOM ocean and sea ice grid orthogonal curvilinear 'zonal' grid spacing dx (top, km); orthogonal curvilinear 'meridional' grid spacing dy (middle, km); and grid aspect ratio dx/dy (bottom, dimensionless).

# 3 COMPONENTS OF THE AUSCOM SYSTEM

## 3.1 Numerical Coupler OASIS3

The Ocean Atmosphere Sea Ice Soil version 3 (OASIS3) is a numerical coupler developed at CERFACS, France (Valcke 2006) under the concepts of portability and flexibility. At run time, OASIS3 acts as a separate mono-process executable. It receives, interpolates, and sends coupling fields (2D only) between the sub-models. It hooks the 'stand-alone' sub-models via a few specific OASIS3 PRISM Model Interface Library (PSMILe) calls implemented in sub-models. OASIS3 is in charge of all data flows (coupling) between sub-models and controls model synchronization via MPI functions.

Currently OASIS3 has been adopted by around 20 climate modelling groups in the world as a coupling framework for their coupled models, and run on various computing platforms. Earlier versions of OASIS have been used in Australia for some years. When the Australian climate sciences community started building its own new generation coupled climate model (at first the coupled ocean-sea ice AusCOM, then the fully coupled model ACCESS) which 'combines' component models from different providers around the world, OASIS3 was chosen to be the numerical coupler because of the following advantages:

- Modular architecture: it is relatively easy for alternative sub-model replacement;
- Individual executables for sub-models: easy code management and compilation;
- Grid independence: conservative remapping between different grids at the 2D interface allows each sub-model to use its own preferred grid. (In practice we make a choice to simplify the coupling computation, as discussed in sections 3.3 and 3.4);
- System expandability: additional sub-models can be relatively easily added into the system.

## 3.2 Data Atmospheric Model MATM

The AusCOM atmospheric component MATM is a mono-processor data model designed for the purpose of handling various atmospheric forcing datasets. It is equiped with a set of data reading subroutines and modules used for different datasets (such as NCEP2, ERA40, CORE and UM AMIP run outputs etc.) which may be of different spatial and temporal resolutions, and often use different variable names for the same fields. In conjunction with an external data table and a datafile selection script (both are preprocessed for a specific dataset), MATM reads in all the required atmospheric forcing fields, with proper scaling and offsetting treatment whenever applicable, and passes them into CICE.

Currently the MATM model resolution is dataset-dependent and must be determined at compile time (i.e. static array definition). Therefore, datasets of different horizontal resolution have to be handled with different MATM executables. In the next release, MATM will be upgraded to support dynamic array definition, and therefore all datasets will be handled using a single MATM executable.

Another feature of MATM worthy of mention is that, for providing the ocean and sea ice system with better 'timing' diurnal foring, MATM is required to send forcing fields of one coupling interval (e.g. 6 hours) ahead of the coupling time. For example, at a coupling time point say, 00:00AM, MATM sends fields of 06:00AM instead of 00:00AM into CICE via OASIS. CICE already has the 00:00AM data (by reading in preprocessed data at the very beginning of an experiment, or passed from MATM at the last coupling time point in the middle of a run). CICE interpolates these two sets of forcing data to obtain the 'right time' forcing for every time-step of the sea ice model within this coupling interval (00:00 to 06:00AM).

## 3.3 Sea Ice Model CICE

The Los Alamos Sea Ice Model CICE (Hunke and Lipscomb 2008; Hunke 2001; Hunke and Dukowicz 1997) was initially designed as a computationally efficient sea ice component for a fully coupled atmosphere-ice-ocean-land global climate model, especially compatible with the Los Alamos Parallel Ocean Program (POP) ocean circulation model POP which is the ocean component of the National Centre for Atmospheric Research (NCAR) Community Climate System Model (CCSM) fully coupled model. In recent years, through collaboration with other institutions such as NCAR and UK Met Office, the CICE4.0 release has been largely enhanced in terms of its technical and physical compatibilities with different models. In particular, CICE now supports tripolar grids, and a 'Zero-Layer' thermodynamic configuration to allow the overlaying atmospheric model to calculate the ice surface temperature. These developments have made it possible (or at least easier) for other users to adopt CICE into their own coupled systems.

CICE consists of several interacting componments: a thermodynamics module computing local growth rates of snow and ice due to vertical conductive, radiative and turbulent fluxes; an ice dynamics module predicting the ice pack velocity field based on a model of the ice material strength; a transport module describing advection of ice concentration, volume and other properties; and a ridging module transfering ice among thickness categories (default 5 categories) based on energetic balances and rates of strain.

The original motivation for using CICE as the sea ice component of AusCOM/ACCESS was on the basis of it being a recognised leading sea ice code. However, as the Met Office Unified Model (UM) has been chosen to be the atmospheric component of the ACCESS coupled model, and UM is specifically configured to be coupled to the NEMO-CICE ocean-sea ice model, our using CICE becomes a natural choice.

The CICE code can be run in a standlone mode with an active boundary module handling the required atmospheric and oceanic forcing, or directly coupled to an ocean model via a supported external "flux coupler" (i.e. the NCAR CCSM coupler). It can also function as a sea ice module/subroutine in an ocean model, such as the ocean component NEMO of the Met Office HadGEM3 coupled system.

In AusCOM, CICE is coupled to MOM4 and MATM via the OASIS3 coupler, allowing for (relatively) flexible data exchange and coupling frequency control. For doing so, the original data exchange functions in CICE supporting the NCAR CCSM coupler have been dropped and a new coupling interface, which consists of a set of subroutines dealing with PSMILe libraries,

has been developed and implemented in the code to facilitate connections between CICE and MATM and between CICE and MOM4 via OASIS3.

As shown in Section 2, AusCOM configures CICE and MOM4 on an identical global tripolar grid. By doing so, coupling CICE to MOM4 becomes relatively straightforward because of grid compatibility. All coupling fields can be exchanged directly between the two models with no need of transformation (e.g. vector rotation, grid remapping interpolation, etc.) by the coupler main process, which to some extent enhances the model computational performance by reducing the workload of the mono-cpu OASIS3 coupler. Further, MOM4 and CICE both being Arakawa B-grid models makes the coupling between them more efficient: exchanged fields need no additional treatment such as grid-point shifting before data sending in the source model or after data receiving in the target model.

CICE plays a very special role in the AusCOM coupled system. As to be revealed in section 4, it functions as a "coupling buffer or media", where all coupling fields are gathered, processed if required, and then delivered to their receivers.

## 3.4 Ocean Model MOM4

The MOM4 ocean model is developed at and supported by the NOAA/Geophysical Fluid Dynamics Laboratory (GFDL). MOM4 is a community code, with contributions by many scientists from collaborating instituations around the world, including researchers at CSIRO Marine and Atmospheric Research. A description of the MOM4 framework and elements can be found in the technical report (Griffies et al. 2004. A detailed introduction to the theoretical aspects of the oceanic physics, dynamics and various parameterizations realised in the model numerics can be found in (Griffies 2004).

We have adopted the MOM4 model as the AusCOM/ACCESS ocean component for these reasons:

- MOM is a recognised leading world-class ocean models code;
- there is a long history of MOM usage in Australia with associated development of in-house expertise;
- synergy is maintained with the other major CAWCR ocean modelling theme, involving Bluelink and associate projects, which focus on eddy-permitting regional to global scale modelling using MOM4;
- there is a long and fruitful history of collaboration between CSIRO and GFDL scientists on ocean modelling, which we wish to continue and enhance.

MOM4p1 is originally configured within the GFDL Flexible Modelling System (FMS, please see www.gfdl.noaa.gov), allowing a wide range of convenience in modelling management such as pre- and post-processing, time control, coupler and data over-riding, diagnostic managing, and so on. It is already coupled to the SIS sea ice model (Winton 2001) via the FMS coupler in the standard MOM4p1 release package.

For connecting MOM4 to CICE via the OASIS3 coupler, we drop the FMS coupling framework at compile time and replace it with a new coupling interface in the model code that links MOM4 to OASIS3. Controls over modelling time and coupling frequency etc. are also handed over to the OASIS3 coupler. However, large parts of the functions provided by the FMS package can still be used by AusCOM MOM4 in an offline manner. These include the pre- and post-processing tools, including grid generator, initialisation, mpp-combining functions etc.

# 4 COUPLING STRATEGY

## 4.1 Coupling fields

AusCOM has a total of 31 2-dimensional coupling fields exchanged between CICE and MATM and between CICE and MOM4 via OASIS (see Table 1). All coupling fields are delivered by OASIS into or from CICE (the "coupling media"), and there is no direct coupling between MATM and MOM4. Note the atmospheric (near-surface) variables 2m air tempterature and 10m velocity are received by the CICE model and converted there into heat and momentum fluxes (i.e. latent and sensible heat fluxes, windstress) by the standard NCAR boundary layer bulk formulae (Large and Yeager 2004, 2009).

## 4.2 Coupling Interfaces

One of the key tasks of building a coupled model under the OASIS3 framework is to develop for each sub-model a coupling interface that connects to the coupler. These interfaces are coded under the OASIS3 philosophy, using the PRISM CLIM communication technique (based on MPI) and associated functions provided by the PSMILe libraries. These subroutines control data transition between the sub-models and the coupler. They include a series of PSMILe subroutine calls which can either be scattered in the model code, mainly within the time loop, or be grouped as a set of functions for better code readablity and the convenience in code maintainance.

In the AusCOM sub-models, all major coupling functions are coded in a coupling module (e.g. cpl_interface). For each of the sub-models that link to OASIS, the coupling module code can be found in the associated driver directory:

- CICE: AusCOM1.0/submodels/cice4.0/drivers/auscom
- MOM4: AusCOM1.0/submodels/mom4p1/src/auscom_cpl
- MATM: AusCOM1.0/submodels/matm/source

Each module consists of a group of 'interfacing' subroutines which are called by the model at different stages to accomplish various coupling tasks. Taking the "coupling media" CICE as the example, the subroutine calls are as follows.

**Table 1:** The 31 coupling fields of the AusCOM system occur in four classes: there are 10 fields passed from the atmosphere model to the sea ice model **(a2i)**; seven fields passed from ocean to sea ice **(o2i)**; one field passed from sea ice to atmosphere **(i2a)**; and 13 from sea ice to ocean **(i2o)**. Where appropriate, 'zonal' and 'meridional' denote local direction on the tripolar orthogonal curvilinear grid. The source and target entries refer to actual variable names in the broadcasting and receiving models, respectively.

| No. | Coupling field | units | source | target |
|---|---|---|---|---|
| | | | | |
| **a2i: Atmosphere to sea ice coupling fields** | | | | |
| 1 | 2M AIR TEMP | $^{\circ}$K | TAIR | TAIR1 |
| 2 | 10M EASTWARD WIND SPEED | m s$^{-1}$ | UWND | UWND1 |
| 3 | 10M NORTHWARD WIND SPEED | m s$^{-1}$ | VWND | VWND1 |
| 4 | DOWNWARD SHORTWAVE RADIATION | W m$^{-2}$ | SWFLD | SWFLX1 |
| 5 | DOWNWARD LONGWAVE RADIATION | W m$^{-2}$ | LWFLD | LWFLX1 |
| 6 | 2M AIR SPECIFIC HUMIDITY | kg kg$^{-1}$ | QAIR | QAIR1 |
| 7 | RAINFALL RATE | kg m$^{-2}$ s$^{-1}$ | RAIN | RAIN1 |
| 8 | SNOWFALL RATE | kg m$^{-2}$ s$^{-1}$ | SNOW | SNOW1 |
| 9 | PRESSURE | Pa | PRESS | PRESS1 |
| 10 | RIVER RUNOFF | kg m$^{-2}$ s$^{-1}$ | RUNOF | RUNOF1 |
| | | | | |
| **o2i: Ocean to sea ice coupling fields** | | | | |
| 11 | SEA SURFACE TEMPERATURE | $^{\circ}$K | SST | SSTO |
| 12 | SEA SURFACE SALINITY | psu | SSS | SSSO |
| 13 | 'ZONAL' WATER SPEED | m s$^{-1}$ | SSU | SSUO |
| 14 | 'MERIDIONAL' WATER SPEED | m s$^{-1}$ | SSV | SSVO |
| 15 | 'ZONAL' SEA SURFACE GRADIENT | m m$^{-1}$ | SSLX | SSLX |
| 16 | 'MERIDIONAL' SEA SURFACE GRADIENT | m m$^{-1}$ | SSLY | SSLY |
| 17 | POTENTIAL ICE FREEZE/MELT HEATFLUX | W m$^{-2}$ | FRZMLT | PFMICE |
| | | | | |
| **i2a: Sea ice to atmosphere coupling fields** | | | | |
| 18 | SEA ICE SURFACE TEMPERATURE | $^{\circ}$C | ISST | SST |
| | | | | |
| **i2o: Sea ice to ocean coupling fields** | | | | |
| 19 | 'ZONAL' ICE-OCEAN STRESS | kg m$^{-1}$ s$^{-2}$ | IOSTRSU | IOSTRSU |
| 20 | 'MERIDIONAL' ICE-OCEAN STRESS | kg m$^{-1}$ s$^{-2}$ | IOSTRSV | IOSTRSV |
| 21 | RAINFALL RATE | kg m$^{-2}$ s$^{-1}$ | IORAIN | IORAIN |
| 22 | SNOWFALL RATE | kg m$^{-2}$ s$^{-1}$ | IOSNOW | IOSNOW |
| 23 | SALT FLUX | kg m$^{-2}$ s$^{-1}$ | IOSTFLX | IOSTFLX |
| 24 | ICE MELTING HEAT FLUX | W m$^{-2}$ | IOHTFLX | IOHTFLX |
| 25 | SHORTWAVE PENETRATING TO OCEAN | W m$^{-2}$ | IOSWFLX | IOSWFLX |
| 26 | LATENT HEAT FLUX DOWN | W m$^{-2}$ | IOQFLUX | IOQFLUX |
| 27 | SENSIBLE HEAT FLUX DOWN | W m$^{-2}$ | IOSHFLX | IOSHFLX |
| 28 | LONGWAVE HEAT FLUX DOWN | W m$^{-2}$ | IOLWFLX | IOLWFLX |
| 29 | RIVER RUNOFF | kg m$^{-2}$ s$^{-1}$ | IORUNOF | IORUNOF |
| 30 | PRESSURE (ANOMALY) | Pa | IOPRESS | IOPRESS |
| 31 | SEA ICE CONCENTRATION | normalised | IOAICE | IOAICE |

### 4.2.1 prism_init

This subroutine is called by CICE on start-up to initialise PSMILe and MPI. This call opens a channel for the model to communicate with the coupler. Major MPI and PSMILe library calls include:

- **MPI_init**: initialise MPI if required.
- **prism_init_comp_proto**: called by all processors to initialise the coupling.
- **MPI_Buffer_Attach**: determine the model MPI buffer size.
- **prism_get_localcomm_proto**: called by all processors to obtain the value of the local communicator to be used by the model for its internal parallelisation. In a mono-coupling case, this local communicator is also in charge of data exchange between model and coupler.

### 4.2.2 cpl_init

This subroutine is also called at the model initilisation stage to determine coupling strategy (mono- or multi-processor coupling) and define coupling field names etc. in the context of the OASIS coupling control namelist in the file **namcouple** (see Section 5 for details). Major PSMILe library calls include:

- **prism_def_partition_proto**: called by all coupling processors to obtain the model MPI partition information. OASIS3 supports 4 types of partition: Serial (no partition, which is the current choice for AusCOM coupling), Apple, Box, and Orange.
- **prism_def_var_proto**: called by all coupling processors to declare each coupling field to be sent or received in the course of a simulation. It passes to the coupler all the required information about a coupling field, including:

    a. symbolic name as defined in the **namcouple** file;
    b. partition identification;
    c. array dimensions;
    d. 'stamp' for sending or receiving;
    e. data type etc.

This subroutine also allocates all coupling field arrays and the associated temporary arrays, which are all defined in a separate module (cpl_arrays).

### 4.2.3 from_atm

This routine integrates all atmospheric data receiving operations. Only one PSMILe routine is involved:

- **prism_get_proto**: it is called repeatedly to receive each atmospheric field declared in cpl_init and listed in the OASIS coupling namelist file **namcouple**.

Routine **from_atm** also supports a 'post-processing' operation, namely, rotating the received vector (wind) from the regular atmospheric latitude-longitude mesh grid onto the CICE tripolar grid, in case the two wind components are passed in as scalar arrays (defined in the **namcouple** file). Doing this and other kinds of transformations in a sub-model instead of in OASIS3 reduces the workload of the mono-processor coupler and enhances the coupling efficiency.

### 4.2.4 from_ocn

This subroutine integrates all oceanic data receiving operations. As with from_atm, the PSMILe routine prism_get_proto is called repeatedly to receive each oceanic field declared in cpl_init and listed in the **namcouple** file.

### 4.2.5 into_ocn

This subroutine handles all operations that send forcing fields into the ocean. The involved PSMILe routine is:

- **prism_put_proto**: called in an ice-to-ocean loop to send each forcing field required by the ocean model, which is declared in cpl_init and listed in **namcouple**.

The above three data transfering routines are the core of the coupling interface connecting CICE to OASIS3 in AusCOM.

### 4.2.6 into_atm

This routine contains only one call to the PSMILe routine **prism_put_proto**, sending one ice-ocean variable into the MATM data model, although not needed there. It is included, as mentioned above, to mimic a complete cycle of data exchange required in a fully coupled model such as ACCESS.

In AusCOM and ACCESS, ice-ocean coupling is allowed to occur more frequently than ice-atmosphere coupling (depending on experimental design), and therefore the subroutines **from_ocn** and **into_ocn** can be called more frequently than **from_atm** and **into_atm** in the course of a simulation. In fact, it is highly recommended that, for AusCOM runs, ice-ocean coupling occur once per time-step (common to both MOM4 and CICE), and that ice receive data from atmosphere once per 6-hours, being the highest temporal resolution of the standard forcing of LargeYeager (2009)).

### 4.2.7 coupler_termination

This subroutine calls the PSMILe routine **prism_def_partition_proto** to terminate connection between model and coupler. It also deallocates all coupling fields.

For the other two models (MATM and MOM4), coupling interfaces are coded and called under the same philosophy, and their data exchange operations are a lot simpler than that in CICE because they both are coupled to CICE only. Therefore, they both have only **into_ice** and **from_ice** for the data exchange operations.

## 4.3 Coupling Approach

### 4.3.1 Code Execution

Under the OASIS3 coupling framework, all sub-models are recoded, mainly in their initialisation and time-stepping loop, to implement calls to the above-discussed coupling interface routines for data exchange and model execution synchronisation. As a result, all of the sub-models are running concurrently and exchanging data with OASIS3 at certain coupling time points in an order determined by the chosen coupling algorithm.

We again take the "coupling media" CICE model as an example to explain the AusCOM coupling approach and demonstrate how it is realised during the model code execution. Following is a symbolic flow of CICE code execution coordinated by the coupler via the data exchange operations:

```
(initialisation)
(read in a2i and i2o fields saved at the end of last run)
(read in o2i fields saved by ocean model at the end of last run)
time_sec = 0
!begin atmosphere-ice coupling iterations:
DO icpl_ai = 1, num_cpl_ai
    call from_atm(time_sec)
    !begin ice-ocean coupling iterations:
    Do icpl_io = 1, num_cpl_io
        call into_ocn(time_sec)
        !cice time loop within each ice-ocean coupling interval:
        do itap = 1, num_ice_io
           (update atmospheric data using time interpolation)
           (calculate forcing fields required for ocean and ice)
           (ice time-stepping)
           time_sec = time_sec + dt_ice
        enddo
        call from_ocn(time_sec)
     End Do
     call into_atm(time_sec - dt) !offset "lag"
     (update i2a fields for the beginning of next a2i coupling)
  END DO
 (save the last calculated i2o fields, which have not been sent into ocean)
 (save the last received a2i fields for use at the beginning of next run)
 (save other fields required for next run)
 (finalisation)
```

The CICE integration is completed by three time loops which advance the ice time-stepping, the ice-ocean coupling, and the ice-atmosphere coupling, respectively. The number of ice steps within an Ice-Ocean Coupling Period (IOCP), number of IOCP within an Ice-Atmosphere Coupling Period (IACP), and number of IACP for the job are determined by the ice time-step, the ice-ocean coupling frequency, the ice-atmosphere coupling frequency, and the integration length of the job. These four time intervals are defined in the AusCOM1.0 runscript (see sub-section 5.5).

Under the concepts of OASIS3 data exchange, nature of air-ice-ocean coupling, and the atmospheric forcing temporal resolution, AusCOM is configured to perform coupling between sub-models and progress simulations in the way shown by the above CICE code execution flow, which, although quite self-explaining, is illustrated by Fig. 4, and further discussed below for users to better understand the AusCOM coupling stragegy.

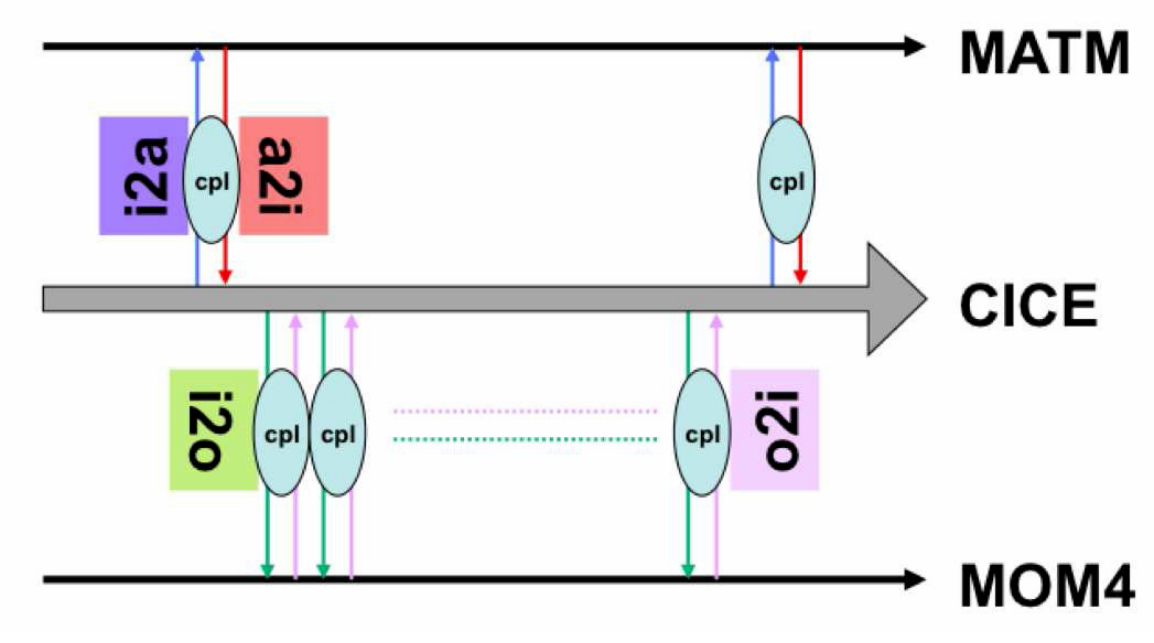


**Fig. 4** AusCOM/ACCESS coupling approach-CICE acts as the “coupling media”.

Firstly, AusCOM allows for different frequencies of coupling atmosphere to sea ice and sea ice to ocean.

- MATM-CICE coupling occurs once per 6-hours, i.e. 4 times daily, which is the most frequent temporal resolution of atmospheric fields in the available datasets (such as NCEP, ERA40 and CORE).

- CICE-MOM4 coupling interval can be flexible, depending on model time-steps and experiment design, usually 1, 2, 3 or 6 hours. In practice, we choose 1 hour as the ice-ocean coupling interval, which is usually the ice and ocean model time-step, for the purpose of minimising coupling lag.

Secondly, CICE functions as a “coupling buffer or media”, where all coupling fields are gathered, processed if required, and then delivered to their receivers.

- At the beginning of a certain ice-atmosphere coupling period (IACP), CICE receives from MATM the ‘raw’ atmospheric fields, then starts the ice-ocean coupling cycles (IOCP) within this IACP.

- At the beginning of an IOCP, CICE first sends the prepared i2o fields to MOM4, then starts its own time-stepping loop.

- At every time-step in an ice internal time loop, CICE updates the atmospheric fields by time interpolation, using them and some oceanic fields (most recently received from MOM4) to calculate surface fluxes required by sea ice and ocean. This calculation is performed by either a built-in boundary layer module or an 'external' module adopted from the GFDL FSM system based on standard NCAR bulk formula of Large and Yeager (2004). Those i2o fields are time-averaged, and weighted by ice coverage if applicable, for the next IOCP.

- After finishing the ice time loop, CICE receives o2i fields from MOM4, and then starts the next IOCP ... and so on, until reaching the end of the curent IACP.

- Then CICE transfers the i2a field(s) to MATM, and starts the next ice-atmosphere coupling cycle.

Note the ice-to-atmosphere sending operation is not really necessary for AusCOM. In practice we let CICE send one field (ocean/ice surface temperature) into MATM only for mimicking a complete cycle of data exchange in a fully coupled model such as ACCESS.

The above is basically the same coupling approach used for the ACCESS fully coupled model, except ACCESS requires a lot more ice-atmosphere data exchange and obviously does not allow for atmospheric data update within an ice-atmosphere coupling interval.

For MATM and MOM4, the code execution occurs in a similar manner but with only 2-level time loops, and the data exchange operations occuring there must be in accordance with their CICE counterparts in every aspect, i.e. coupling fields and coupling time, otherwise the system would enter a deadlock.

### 4.3.2 Coupling Algorithm: Mono-processor vs Parallel Coupling

AusCOM and ACCESS use a mono-prorcessor coupling algorithm for all sub-models, although multi-processor (parallel) coupling is seemingly more efficient. Our arguement is that, under the mono-processor OASIS3 coupling framework, efficiency of data exchange between sub-models via the coupler largely depends on workload of the coupler. At a coupling time, letting the coupler communicate with all processors of each sub-model (i.e. do all the global gathering and scattering work for all coupling fields) may take more time than the time saved by each sub-model's avoiding global gathering and scattering operations.

Figure 5 illustrates the fate of a certain coupling field F when it is passed from the source model A to a target model B via the OASIS3 coupler, using a mono-processor coupling algorithm. Let's assume F is partitioned into 2 x 3 and 2 x 2 sub-domains in A and B, respectively. At coupling time, model A's coupling processor (local communicator) gathers from all slave processors the partitioned information of F, places F onto the global domain of the source grid, and then sends this global field to the coupler. If required, which is the case for most coupling operations but not for the AusCOM ice-ocean coupling, the coupler re-maps this global field onto the global domain of the target model grid by suitable interpolation. Then the coupler sends F into model B where the local communicator receives the global field F and scatters it onto all slave processors, including itself.

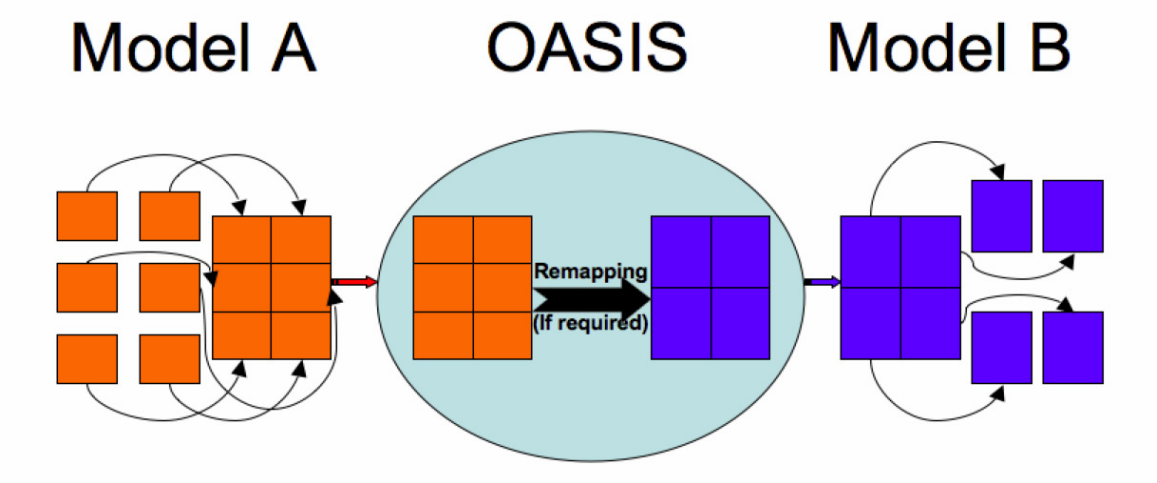


**Fig. 5** Schematic diagram of the mono-processor coupling algorithm in AusCOM/ACCESS.

In this coupling process, MPI global gathering and scattering operations are needed in the source and target models, respectively, to some extent slowing down the sub-models execution. This drawback may become significant if large number of processors are used for the sub-models.

Figure 6 demonstrates how the multi-processor coupling works for the coupling field F. At coupling time, each of the coupling processors in model A, usually all the processors allocated for this model, sends their own partitioned information of F to the coupler. The coupler receives this partitioned field and gathers it 'automatically' onto the global domain of the source grid (following the partitioning pattern defined in model A's coupling interface). Then the coupler re-maps this global field F onto the global domain of the target grid by interpolation, divides it into partitions (following the partitioning pattern defined in model B's coupling interface), and sends each partition of field F directly to model B's corresponding coupling processors.

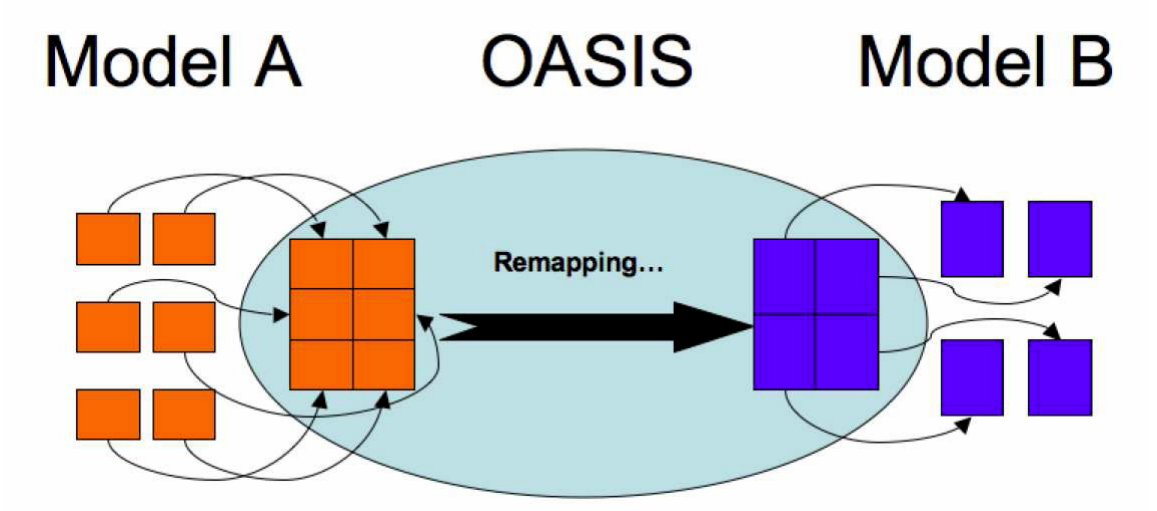


**Fig. 6** Schematic diagram of the multi-processor coupling algorithm in AusCOM/ACCESS.

Obviously this ‘parallel’ coupling does not involve any MPI global gathering and scattering in the sub-models, and therefore does not slow down the sub-models. However, in ‘parallel’ coupling, all processors of all sub-models communicating with the coupler does not necessarily enhance the computing efficiency of the coupled system. In contrast, because the coupler takes over all the gathering and scattering duties which would otherwise be undertaken by sub-model local communicators in the mono-processor coupling case, the system coupling efficiency actually becomes lower, highlighting the coupling bottleneck. This is especailly true when the system, such as the ACCESS fully coupled model, has a large number of coupling fields and requires higher coupling frequency. This is the reason why we have chosen the mono-processor approach over the ‘parallel’ coupling approach for our AusCOM and ACCESS coupled systems.

# 5 A TECHNICAL GUIDE TO THE AUSCOM SYSTEM

This section is a full technical manual for users to install the AusCOM system, design experiments and set up runs for their own applications.

## 5.1 Platforms

A majority of the AusCOM development was undertaken on the National Computational Infrastructure National Facility (NCI-NF) SGI cluster XE and Sun Constellation cluster VAYU platforms. The AusCOM1.0 model with IPCC-class configuration has been running on VAYU, the current peak system at NCI, for many multi-century simulations, and can achieve the computational performance of at least 20 model years per day on 128 processors (i.e. 16 nodes).

This manual provides guidance for AusCOM users who have access to the XE and VAYU machines. We assume users know about the machine configuration and disk system, and also have access to data under project p66. For users who have no access to /short/p66/, we may provide the required forcing data for them to store in their own /short space. For users who wish to set up AusCOM1.0 on another computing platform, the model IO management, compiling and run control etc. introduced here may all need to be changed according to the specific infrastructure of their platforms.

## 5.2 AusCOM1.0 System Infrastructure

Ideally, but not necessarily, the AusCOM1.0 'base' directory should be set up under the user's home directory. It contains the following sub-directories:

**AusCOM1.0/bin**
**/exp**
**/input**
**/submodels**
**/forcing**
**/output**

In practice, the first four sub-directories can be phyically located here, whilst the other two may be just symbolic links to their physical locations on other disks that allow for large data storage, such as the /short system on XE and VAYU. This arrangement is based on the fact that the $HOME space quota is always very limited and should be used for files that need permanent storage.

### 5.2.1 bin/

It stores model and coupler executables, and some auxiliary tools used for various purposes. For running an AusCOM experiment, the following minimum number of files are located here:

- oasis3_MPI1.exe: the OASIS3 coupler executable.
- matm_MPI1.exe: the data atmospheric model MATM executable.
- cice_MPI1.exe_np: the sea ice model CICE executable compiled for using "n" processors.
- mom4_MPI1.exe: the ocean model MOM4p1 executable.
- calendar.exe: a calendar tool used in the runscript for calendar management.
- mppnccombine.exe: a post-processing tool used to combine the domain processor based MOM4 history files.
- do_mppncombine.ksh: a script calling mppnccombine.
- environs.vayu.nci.org.au: loads modules for the AusCOM compilation environment.

### 5.2.2 exp/

All AusCOM experiments, such as test-01/, ciaf2-99/ should be set under this directory. Jobs are sumbmitted under their own run directories through a preprocessed runscript located there. for example, in one of the sample runs,

**$ cd ciaf2-xx**
**$ ls**
**run_auscom.VAYU**

Once the job is submitted, a new directory "Running_dir" will be created here, which is a symbolic link to the directory where the job is actually running (see section 6).

### 5.2.3 input/

**$ ls input/**
**cice matm mom4 oasis3**

These directories store the preprocessed data files, namelist files etc. required by each sub-model and the coupler for system initialisation, configuration, and experiment control. The following lists show the files that are essential for running an AusCOM experiment.

under input/matm/:

- core2_fields_4_matm.table: COREv2 forcing data table.
- get_core2_IA.ksh: script called by runscript to set COREv2 Inter-Annual forcing.

- get_core2_NY.ksh: script called by runscript to set COREv2 Normal Year forcing.

under input/cice/:

- cice_grid.nc: CICE model grid information file.
- cice_kmtu.nc: CICE model grid land-sea masks.
- cice4.0_in.nml: CICE model namelist file for experiment design and control.
- SSTS.nc: SST and SSS data used to initialise the sea ice properties, if required.
- A2I_time0_10fields.nc: air-to-ice forcing data at the beginning of an experiment (to be read in by CICE for 'two-time-level' forcing purpose).
- core_runoff_regrid.nc: CORE runoff data regrided on the CICE grid, used as replacement of the runoff received from coupler, if required.
- u_star_t0.nc: $u^*$ data required by CICE if the GFDL module is used to calculate surface fluxes.

under input/mom4/:

- grid_spec.nc: MOM4 grid information data file.
- mom4_in.nml: namelist file for MOM4 physical configuration and experiment control.
- diag_table: MOM4 diagnostic output table.
- field_table: MOM4 field table.
- data_table: MOM4 data table (not used in AusCOM but existence is required).
- ocean_temp_salt.nc: MOM4 ocean initial condition for temperature and salinity.
- salt_sfc_restore.nc: 12-monthly SSS data for surface salinity relaxation.
- temp_sfc_restore.nc: 12-monthly SST data for surface temperature relaxation.

under input/oasis3/:

- grids.nc: grid lat-lon and rotation angle information of all sub-model grids.
- masks.nc: land-sea masks for all sub-model grids.
- areas.nc: grid cell areas for all sub-model grids.
- namcouple: OASIS3 configuration file, containing all coupling fields.
- cf_name_table.txt: a text file for OASIS3 to obtain coupling fields description.
- a2i.nc, o2i.nc, i2o.nc, i2a.nc: preprocessed restart files for OASIS3.

### 5.2.4 submodels/

It contains full packages of codes of the sub-models and coupler.

**submodels/matm**
**/cice4.0**
**/mom4p1**
**/oasis3_prism_2-5**

Each sub-model has its own sub-directory trees (some of which are very deep):

**matm/source**
**/bld**
**/compile**

**cice4.0/csm_share**

**/doc**
**/drivers/.....**
**/auscom**
**/input_templates**
**/mpi**
**/rundir**
**/serial**
**/source**
**/compile**

**mom4p1/bin**
**/exp**
**/src/.....**
**/auscom_cpl**
**/compile**

**oasis3_prism_2-5/prism/Linux**
**/data**
**/src**
**/util**
**/compile**

For compatibility, the three 'imported' sub-systems CICE, MOM4 and OASIS3 retain their own original infrastructure and contents, as determined from the public release package, which will allow for more straightforward system upgrading using subversion (svn) in the future. Note however, the addition of the extra "driver" directories where the bulk of the coupling code can be found.

- CICE: AusCOM1.0/submodels/cice4.0/drivers/auscom
- MOM4: AusCOM1.0/submodels/mom4p1/src/auscom_cpl

There are additional places in the modelling code where various modifications are made for coupling purposes. These are indicated by "#ifdef AusCOM" precompiler statements. Also, for each component we set a new subdirectory compile/, as shown above, in which scripts are provided for compiling each sub-model and also the coupler.

### 5.2.5 forcing/

Under this directory, symoblic links should be set to the physical locations where the available atmospheric forcing datasets are stored. For example, we need:

**$ ls -l forcing/**
**..... CIAF -> /short/p66/sjm599/COREv2_26JAN2010/CIAF**
**..... CNYF_v2 -> /short/p66/sjm599/COREv2_15JUNE2009/CNYF_v2**

The scripts input/matm/get_core2_IA.ksh etc. mentioned above will pick up atmospheric forcing data from these directories. Users should note that the CORE forcing is updated from time to time. As such, the get_core2*.ksh scripts may also be updated.

### 5.2.6 output/

Under this directory we can find symbolic links to outputs of experiments. For example,

**$ ls -l output/ ......**
**ciaf2-xx -> /short/p??/(userid)/OUTPUTS/AusCOM1.0/ciaf2-xx**

**$ ls output/ciaf2-xx/***
**ciaf2-01/history:**
**cice mom4**
**ciaf2-01/restart:**
**cice mom4 oasis3**

These links are established by the runscript when the initial run of an experiment is conducted.

## 5.3 AusCOM Compilation

To build the AusCOM system under the OASIS3 framework, all sub-models need to use PSMILe libraries. Therefore we should first compile the PRISM_2-5 libraries and generate the OASIS3 executable. The three sub-models can then be built in any order.

As already mentioned, a building directory compile/ has been set up for each of the sub-models, and each sub-model can be built using the provided script under compile/, in a very simple way

### 5.3.1 PSMILe libraries and OASIS3

**$ cd AusCOM1.0/submodels/oasis3_prism_2-5/prism/compile/**
**$ comp_oasis325.VAYU**

Note the compiled PRISM_2-5 librares and object files are stored at:

**../Linux/lib**
**/build**

The OASIS3 executable "oasis3.MPI1.exe" is moved to AusCOM1.0/bin/ immediately after being generated, where the sub-model executables are also stored.

### 5.3.2 MATM

**$ cd AusCOM1.0/submodels/matm/compile/**
**$ comp_auscom_matm.VAYU**

Note, as mentioned earlier, MATM resolution is dataset-dependent and must be determined at compile time. For example, to handle the CORE forcing fields, which are on the "nt62" grid on a 192x94 mesh, MATM must be compiled specifically for this grid. This can be achieved by setting two variables GRID and RES in the compiling script comp_auscom_matm.VAYU:

**# Grid resolution**
**setenv GRID nt62**
**setenv RES 192x94**

Consequently, the generated MATM executable is labelled with the grid name nt62 to distinguish it from other MATM executables compiled for different grids such as the um96 UKMO Unified Model grid used in the ACCESS fully coupled model.

The actual building location is

**compile/build_MPI1_nt62/**

and the generated executable matm_MPI1.exe_nt62 is also moved to AusCOM1.0/bin/.

### 5.3.3 CICE

**$ cd AusCOM1.0/submodels/cice4.0/compile/**
**$ comp_auscom_cice.VAYU.nP N**

Note the compiling script must be provided with an integer "N" such as 1, 2, 3, 6, 12 etc, specifying how many processors are to be used to run CICE. The processor usage is determined at compile time because the CICE code does not support dynamic MPI partitioning. Consequently, the generated CICE executable must be labeled with this number to distinguish it from other CICE executables compiled for using a different number of processors.

Following the convention, the building directory is compile/build_MPI1_Np/, and the executable cice_MPI1.exe_NP is moved to AusCOM1.0/bin/, where one may soon find more than one CICE executable, having been built for different numbers of processors. This happens because for different applications AusCOM may need to use different computing resources. Also, processors allocated for CICE and MOM4 should be adjusted to achieve a reasonable load balance for the best computing efficiency. Note, however, because of the different scalabilites of CICE and MOM4 on different platforms, processor allocations for these two models may vary largely for achieving a suitable load balance. From experience, we find that MOM4 needs to use 10-20 times more processors than CICE.

In addition, CICE4.0 code does not support dynamic array allocation. Grid resolution must be specified in the compiling script by setting variable GRID and RES as following,

**# Grid resolution**
**setenv GRID tp1**
**setenv RES 360x300**

and, because of its dependence on the MOM4 grid, whenever changes are made to the MOM4 grid horizontal resolution, the above two lines must be changed accordingly, and CICE be re-compiled.

### 5.3.4 MOM4

On VAYU, AusCOM MOM4 can be compiled straight away:

**$ cd AusCOM1.0/submodels/mom4p1/compile/**
**$ comp_auscom_mom4p1.VAYU**

On XE, however, MOM4 compilation is somewhat more complicated:

```
$ cd AusCOM1.0/submodels/mom4p1/compile/
$ qsub -q express -lwalltime=01:00:00,vmem=2Gb,jobfs=4Gb,ncpus=1 -wd -I \
        -lsoftware=intel-fc:intel-cc
$ comp_auscom_mom4p1.XE
```

The complication on the XE comes about because we need to request more memory usage for the mono-processor compiling operation, which would otherwise fail in generating the ocean boundary condition module ocean_obc_mod that requires large memory.

It is worth mentioning that XE and VAYU are compatible in their operating systems. Executables compiled on one platform can be readily executed on the other platform. Therefore, users who have accounts on both XE and VAYU may compile the AusCOM system on only one of these two machines and set up runs on both of them. The only inconvenience is that XE and VAYU do not share the disk systems, and users need to make sure the forcing data are available for both machines.

## 5.4 OASIS3 Configuration File

For users who are keen to configure their own version of the AusCOM system or run their own designed experiments using the standard AusCOM1.0 release, some of the provided auxliary files may need to be modified, or even regenerated. Please check the relevent documentation of the sub-models and coupler, or consult the AusCOM developers for advice to make sensible changes.

There are two key namelist files required for AusCOM physical configuration. They are cice4.0_in.nml for CICE and mom4_in.nml for MOM4, adapted from the CICE4.0 and MOM4p1 release. Some ‘minor’ changes have been made, especially to mom4_in.nml, for flexible model management and tuning. Users are encouraged to tackle physics issues in MOM4 and CICE by switching the supported parameterization schemes and changing phyical parameters (within a reasonable range) in these two namelist files for achieving better modelling performance of the coupled ocean and sea ice system.

Here we focus on the core namelist file for AusCOM coupling control, ‘namcouple’.

### 5.4.1 Introduction

namcouple is the key OASIS3 namelist file configuring and controling all coupling operations of the system in the course of modelling for a particular set-up. It provides the coupler with all required information about coupling such as the associated component models and their processor usages, run time and calendar option, full list of coupling fields and associated prerequisites including the souce and target grids, coupling algorithm, coupling frequency, remapping interpolation options etc.

The namcouple file released with AusCOM1.0 is designed for the sample experiments based on the example namcouple used for the OASIS3 ToyClim model. It was prepared with extreme care after the coupling approach was determined, and the coupling interface in each sub-model

was set up. All items listed in this file, some of which are to be discussed below, must be consistent with their coresponding part defined in each of the sub-models, or in other preprocessed auxiliary data files such as the OASIS3 grid information files.

As required by OASIS3, namcouple predefines 11 groups of parameters for coupling control. Each group contains one or more parameters, starts with a $KEYWORD line (e.g. $RUNTIME), and ends with a $END line. As shown in Appendix A, which is the complete namcouple file used for the example run ciaf2-xx, the first 10 groups are general parameters, and are all well described inside the file. Therefore, the following introduction focuses only on the last, large group of parameters, which configures all the coupling fields, one by one.

### 5.4.2 Configuring Coupling Fields

This part is found in namcouple between lines $STRING and $END. It contains multi-line configuring information for each of the coupling fields, with a particular format which depends on the 'field status' given by the last entry on the first line of each field's configuring block. Note in namcouple all comment lines start with "#" and empty lines are not allowed.

AusCOM's 31 coupling fields are divided into 4 groups, namely, i2a, i2o, o2i, and a2i. We choose one field from each group to demonstrate how they are configured under the OASIS3 coupling philosophy. See Appendix A for a complete list of the coupling fields.

The atmosphere to ice (a2i) fields are formulated as follows.

```
#------------------
# Atmosphere ==> Ice
#------------------ #
# Field 01: downward short wave radiation
swfld_ai swfld_i 367 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0
CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
```

This coupling field has 8 configuring lines with different numbers of entries.

1. The first line includes 7 entries:

- swfld_ai: symbolic name of this field as defined in the source model MATM (character*8).

- swfld_i: symbolic name of this field as defined in the target model CICE (character*8).

- 367: index in auxiliary file cf_name_table.txt used by OASIS3 and PSMILe to identify proper description for this coupling field just for printing purpose, i.e. 'surface net downward shortwave flux (W $m^{-2}$)'. Note this number and the associated description have no modelling significance and thus may not necessarily reflect the real definition

of the coupling field. In fact, filed swfld_ai is actually defined and used as '(total) surface downward shortwave flux (W $m^{-2}$)' in the AusCOM model code.

- 21600: coupling interval (in seconds) for this field.

- 4: number of transformations to be performed by coupler on this field. These 4 transformations are specified below (line 4 here), followed by their own configurations (one line each, in order).

- a2i.nc: coupling restart filename (character*8). In AusCOM, coupling fields in the same group share one restart file.

- EXPORTED: field status, indicating that this field is to be exchanged between component models and transformed by OASIS3 main processor. OASIS3 supports a number of field status, all named in capital: AUXLIARY, EXPORTED, EXPOUT, IGNORED, IGNOUT, INPUT, and OUTPUT, standing for different treatments for the coupling field. Please see the OASIS3 User Guide (Valcke 2006) for detailed descriptions.

2. The second line has 4 entries:

- nt62: 4-character string, source grid name in grid datafiles where all grid information variables must use their grid name as the prefix of the variable names (e.g. nt62.lon, cice.ang etc.).

- cice: 4-character string, target grid name ("cice" is the AusCOM MOM4/CICE tripolar grid).

- LAG=+3600: lag index for this field. 3600 is the MATM time-step (in seconds)

- SEQ=+1: sequence index. LAG and SEQ must be set properly in the context of the coupling strategy to avoid coupling deadlock. See Valcke (2006) for detailed discussions.

3. The third line contains 4 entries:

- P: source grid first dimension characteristic (P for periodical and R for regional).

- 0: source grid first dimension number of overlapping points (none for nt62).

- P: target grid first dimension characteristic.

- 0: target grid first dimension number of overlapping points (none for cice).

4. Four transformations for this field are listed in line 4:

- LOCTRANS: time transformation.

- CHECKIN: preprocessing, namely, calculating the field mean and extreme values on the source grid and printing them to the coupler log file cplout. The other available and useful preprocessing transformations in OASIS3 include MASK, EXTRAP, CORRECT.

- SCRIPR: interpolation transformation. OASIS3 also supports BLASOLD, INTERP, MOZAIC, NOINTERP, and FILLING for different applications.

- CHECKOUT: postprocessing, namely, calculating the field mean and extreme values on the target grid and printing to cplout.

5. Line 5 configures the time transformation LOCTRANS:

- INSTANT: the instantaneous field is transfered (i.e. no time transformation). The supported time transformations include INSTANT, ACCUMUL, ACERAGE, T_MIN, T_MAX, and ONCE. See Valcke (2006) for full descriptions. Note, for all the AusCOM coupling fields, the only time transformation needed is time average, and it is done in sub-models. Communication between a sub-model and OASIS3 only occurs at actual coupling time points, avoiding "unnecessary" data exchanges and time transformations in OASIS3 and therefore reduing its workload.

6. Line 6 configures the preprocessing transformation CHECKIN:

- INT=0: 1 or 0, whether or not calculating and printing the source field integral.

7. Line 7 configures the interpolation transformation SCRIPR: SCRIPR in PSMILe adopts the LANL CSRIP1.4 interpolation software (Jones 1997) and offers the following types of interpolation: DISIWGT, GAUSWGT, BILINEAR, BICUBIC, and CONSERV. Each of them has advantages and disadvantages, and may be used for different circumstances. For AusCOM, the CONSERV algorithm is chosen to perform interpolations for all fields, although it is only necessary for those involving mass and energy fluxes.

- CONSERV: performs first or second order conservative remapping. The following entries further configure this interpolation algorithm:

- LR: the source grid type (Logically Rectangular).

- SCALAR: field type.

- LATLON: search restriction type.

- 10: number of restriction bins.

- FRACNNEI: normalisation option for the SCRIPR CONSERV interpolation algorithm. The other two alternative options are FRACAREA and DESTAREA. Extra caution needs to be taken for choosing the proper one, otherwise the model might not behave due to "abnormal" products of the interpolation. FRACNNEI is our choice for AusCOM after numerious tests, mainly because it results in a reasonable flux value, although local flux conservation is not guaranteed.

- FIRST: the conservation order.

8. The last line configures the postprocessing transformation CHECKOUT:

- INT=0: Specifies to not calculate or print the target field integral.

The other 9 i2o fields are configured the same way. This means that even the two wind components are treated as scalars. By doing so we in effect reduce the workload of the coupler which would otherwise be required to perform vector rotation transformations. As already expressed, we let the sub-models take care of the vector rotation before or after coupling, whenever required. For users who do want the vector rotation to be done in OASIS3, the wind components (fields No. 9 and 10) may be configured as below:

```
# Field 09 : 10m wind (u)
uwnd_ai uwnd_i 56 21600 3 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P  0  P  0
CHECKIN SCRIPR CHECKOUT
INT=1
DISTWGT LR VECTOR_I LATLON 10 4 vwnd_ai
INT=1
#
# Field 10 : 10m wind (v)
vwnd_ai vwnd_i  56 21600 3 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P  0  P  0
CHECKIN SCRIPR CHECKOUT
INT=1
DISTWGT LR VECTOR_J LATLON 10 4 uwnd_ai
INT=1
```

Our tests show that this approach and the recommended give very close model results.

The ice to ocean (i2o) fields are formulated as follows.

```
#-------------------
# Ice ==> Ocean
#-------------------
#
# Field 11: ice-ocean interface stress (x-direction)
strsu_io strsu_o 170 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
LOCTRANS
INSTANT
```

Coupling field No. 11 is configured by only four lines of parameters.

1. The first line has 7 entries:

   - strsu_io: symbolic name of this field as defined in the source model CICE (character*8).

   - strsu_o: symbolic name of this field as defined in the target model MOM4 (character*8).

   - 170: index in cf_name_table.txt used by OASIS3 to obtain description for this field.

   - 3600: interval (in seconds) of ice-ocean coupling.

- 1: only 1 time transformation to be performed by coupler.
- i2o.nc: coupling restart filename (character*8) for all ice-to-ocean fields.
- IGNORED: field status, indicating that this field is to be exchanged directly from CICE to MOM4 without being transformed by OASIS3 main process. This is the advantage of CICE and MOM4 sharing the same tripolar grid.

2. The second line has 4 parameters:
   - cice: source (CICE) grid.
   - cice: target (MOM4) grid.
   - LAG=0: no lag for ice-ocean coupling.
   - SEQ=+1: sequence number.
3. Line 3 and Line 4 configure the only time transformation, the same as that for field 1.

The other 12 i2o fields are configured the same way.

The ocean to ice fields (o2i) are formulated as follows.

```
#------------------
# Ocean ==> Ice
#------------------
#
# Field 24: sea surface temperature
sst_oi sst_i 1 3600 1 o2i.nc IGNORED
cice cice LAG=0 SEQ=+1
LOCTRANS
INSTANT
```

The 7 o2i fields are all configured the way shown above. It's basically the same as that for the i2o fields. Those explanations for all parameters apply here, except for the source and target names being swapped.

Finally, the ice to atmosphere (i2a) field is configured as follows.

```
#------------------
# Ice ==> Atmosphere
#------------------
#
# Field 31: ice/ocean surface temperature
isst_ia isst_a 1 21600 4 i2a.nc EXPORTED
cice nt62 LAG=+3600 SEQ=+1
P 0 P 0
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0
```

**CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST**
**INT=0**

This only i2a field, which is also the last coupling field listed in the namcouple file, is configured the same way as that for the a2i fields discussed earlier, except for the source and target names being swapped.

It must be stressed again that, due to the complexity of the OASIS3 coupling philosophy and a wide range of choices for coupling algorithms, creating a functional coupling configuration file, especially the coupling fields configuration part, is not an easy task. This sample namcouple file works fine, but is not necessarily a very reasonable and efficient design for the AusCOM system's high standard coupling performance. AusCOM users with good experience in OASIS3 are strongely encouraged to explore deeper into the interpolation algorithms (and other aspects) and test their own preferred schemes.

With all the auxiliary files preprocessed, the design of an AusCOM experiment is finalised and the model is, well, almost ready to run.

## 5.5 The Runscript run_auscom.VAYU

### 5.5.1 Introduction

A single runscript named run_auscom.VAYU is used to set up and run AusCOM on the NCI VAYU platform. This UNIX script is written in the Korn shell (ksh) programming language, and contains sections performing various management tasks, including system initialisation, run time calendar control, IO management, job re-submission and so on. In addition, it supports run time modifications to some key configuration files for flexible experiment and run setup (which of course unavoidably increases the file length). This is proven very helpful in many circumstances, especially for test runs that need a multitude of tuning work.

A sample runscript file is provided in this document as an important part (Appendix B) for users to examine before they conduct any AusCOM experiments. In the AusCOM1.0 release package, it is located under the three sample run directories.

Although it is quite self-descriptive, some key parts of this multi-hundred line script need more detailed explanation. This introduction is aimed to help users to set up their own experiments. Therefore any parts associated with a specific run setup will be highlighted for users attention. Those parts that are required for general or common use may just be mentioned or simply skipped. In addition, here we follow the section sequence in the script for reader's convenience of target-locating, but prefix the section numbers with characters "R-" to distinguish them from the section sequence of this document.

### 5.5.2 Steps for Setting up an AusCOM Run

#### R-0. Prologue

This part requests computing resouces through the PBS system and sets up some environment variables associated with MPI and Fortran IO control. Frequently changed are the following key PBS lines:

**#PBS -P p66**
**#PBS -q normal**
**#PBS -l ncpus=128**
**#PBS -l vmem=128GB**
**#PBS -l walltime=1:30:00**
**#PBS -N ciaf2-xx**

They define project (p66, user-dependent), queue type (normal), number of processors to be used (128), memory request (128GB), wall-time needed (1.5 hours), and the job name (ciaf2-xx).

# R-1. Primary Setups

### R-1.1 Experiment ID and Forcing

**project=p66**

This defines the project that the user belongs to on the NCI system, so that the user can use the short/$project disk to run the model and store model outputs. It's not necessarily the same as that in the "#PBS -P p??" line if the user belongs to more than one project.

**expid=ciaf2-xx**
**atmdata=core2**
**datatype=IA**
**year_data_end=2007**

These lines define experiment ID (run name, same as that defined by the "#PBS -N ciaf2-xx" line), the atmospheric forcing dataset, forcing type (IA for interannual, NY for normal year), and up to which year the data is available (for a NY forcing run, can be set up to 9999). Although MATM also supports other forcing such as NCEP2, ERA40, and UM96 (UM AMIP run outputs), the CORE2 dataset is the recommended, standard atmospheric forcing for AusCOM.

### R-1.2 Run Path Definition

This part sets all paths associated with this experiment or segment run, specifically, paths to the preprocessed input files, model outputs, and the working directory, which on VAYU should all be located on the /short disk system. For example,

The actual run directory of the sample run ciaf2-xx is:

**$rundir=/short/$project/$user/RUNNING/AusCOM1.0/ciaf2-xx/**

and its outputs are stored at

**$outputdir=/short/$project/$user/OUTPUT/AusCOM1.0/ciaf2-xx/**

Users are not supposed to make any change here. All required directories will be created by this script when the experiment is started (i.e. at the beginning of the initial run).

Useful tips:

- When a job is running, a symbolic link to the physical location of the working directory ($rundir) will be established under the submitting directory ($jobdir), named as "Running.dir", which helps the user get direct access to $rundir (without caring about where it is actually located) to monitor simulation progress.

- Also symbolically linked is the model archive directory, which can be accessed directly from say, AusCOM1.0/output/ciaf2-xx. This helps the user locate the model outputs very efficiently.

## R-2. Exp/Run Time Control

This is one of the most frequently visited parts.

### R-2.1 Runtime Control for the Whole Experiment and a Segment Run

**iniyear=1948**
**finalyear=1950**
**inimonth=1**
**finalmonth=12**
**iniday=1**
**finalday=31**

These lines set the simulation period for the experiment ciaf2-xx: from the first day of year 1948 to the last day of year 1950, i.e. 3 complete years.

For a datatype=NY (CORE Normal Year) experiment, the iniyear should be set to "1" and finalyear can be any sensible number (1, 10, 500, up to 9999).

**nyear=0**
**nmonth=12**
**nday=0**

The above lines set the simulation period for a segment run of this experiment, namely, 12 months. Note "nyear" must always be "0" (for some "unreasonable" reason in the calendar management program code curently used, not essential for its function though), and for a multi-year segment say, 2-year run, we can set nmonth=24. Theoretically, any combination of "nmonth" and "nday" are acceptable, such as "nmonth=7; nday=13", but in practice we always set whole months, and in short test cases, such as a 3-day test, one may set "nyear=0; nmonth=0; nday=3".

The following 'self-meaning' variables set time-step for all sub-models, atm-ice and ice-ocean coupling intervals. As is shown, atm-ice coupling occurs once per 6-hour, whilst ice couples with ocean every time-step, i.e. once per hour.

**dt_cpl_ai=21600**
**dt_cpl_io=3600**

**dt_atm=3600**
**dt_ice=3600**
**dt_oce=3600**

A useful tip:

- In some circumstances users may encounter a (MOM4) model crash due to "free surface penetrating into rock", a sign of instability caused by a number of reasons. Reducing "dt_oce" to 1800 (half an hour), only for the crashing run, often helps to get around this problem.

**R-2.2 Processor usage**

**nproc_cpl=1**
**nproc_atm=1;  ncplproc_atm=1**
**nproc_ice=6;  ncplproc_ice=1**
**nproc_oce=120;  ncplproc_oce=1**
**(( ntproc = nproc_atm + nproc_ice + nproc_oce + nproc_cpl ))**

These variables declare how many processors should be allocated for each model and the coupler. Note the coupler and MATM are both hardwired to use 1 processor each, but CICE and MOM4 processor usage can vary. Also declared here is the number of coupling processors for each sub-model. As already mentioned, AusCOM 1.0 currently only supports mono-cpu coupling.

**R-2.4 Calendar Date Control**

Attention needs be paid to these two lines, they are forcing dataset dependent.

**caltype=1  #0, 1 or n (eg, 30 for 30 day months)**
**cal_type="'julian'"  #for caltype=1; "'thirty_day'" for caltype=30.**

The following if block

```
if [ ! -f ${exp}.date ]
  ......
fi
```

determines status of the current job: initial run (jobnum = 1) or continue run (jobnum larger than 1), by existence of file (exp).date (e.g. ciaf2-xx.date) and sets the required date and time control information for this job accordingly.

This (exp).date file has an important role in controlling the experiment:

A newly set up experiment should only have the runscript in position. After the initial run is successfully completed, file (exp).date is created, containing start time and date information for the second run. When the second run is finished, the previously used (exp).date is renamed to (exp).date_2 and a new (exp).date is created for the third run; and so on. Therefore, as an experiment progresses, more and more (exp).date files will be created, labelled with job numbers and saved for possible future use.

A useful Tip:

- If users wants to restart a run (from a certain time point in the run history) for whatever reason, they can simply rename the associated (exp).date_?? file to (exp).date and submit the job, under one condition though: the associated restart files for all sub-models and the coupler have to be there.

## R-3. Getting All the Files into the Run Directory

This section puts in place all preprocessed auxiliary files, initial condition data files, and sub-model and OASIS executables. Some configuration files that control run time execution will be edited to reflect particular features of the experiment, and control over the to-be-performed run segment. Run time configuration of 'namelist' files are also created for each of the sub-models.

### R-3.1 Grids, Initial Condition, Executables and some Preprocessed Auxiliary Files

```
boundary_layer=gfdl      #"gfdl" or "non-gfdl"
runoff_data=core         #"core" or "non-core"
cold_start=1             #1/0: cold/warm start
```

In this section, probably only these top lines need close attention. They decide some features of an experiment:

- using the 'external' GFDL module for surface flux calculations (otherwise using the CICE 'built-in' boundary module).

- using the preprocessed CORE runoff (regrided, currently the annual mean climatology only) data instead of that read in by MATM and interpolated by OASIS3. Doing so ensures reasonable re-distribution (spreading) of the river runoff into ocean, avoiding potential model crashes due to unrealistic freshwater 'accumulation' and assocaied abnormal heating at some river outlets. The penalty is that model loses all temporal variablities of the runoff forcing.

- ocean and sea ice start from the observed temperature and salinity, namely, a "cold start". Otherwise the ocean-sea ice system will use a "warm start", as described below.

A "warm start" means an experiment is initialised with the full set of restart data from an existing AusCOM spin-up run. So in the case of "cold_start=0", the user must set a path to an initial condition source, for example:

```
owner=dhb599
expname=cnyf2-01
ic_year=500
access=~$owner                  #if run under owner's home directory
#access=/short/p66/$owner       #if run under /short/p66 disk
```

These example lines indicate that restart files of (end of) year 500 from user dhb599's spin-up run cnyf2-01 will be used to start this experiment.

Following the above pre-definition, all data files will be copied to the working directory (i.e. $rundir).

### R-3.2 Adapting or Creating Configuration Files

This sub-section handles all configuration files. They are either edited from their own 'template' file by replacing some adjustable (i.e. experiment/run dependent) parameters, or created at run time. These files are:

- namcouple: OASIS3 coupling configuration (namelist) file (edited).
- input_atm.nml: MATM run and coupling control file (created).
- data_4_matm.table: definition of names of forcing variables and files (created).
- cice_in.nml: CICE configuration (namelist) file (edited).
- input_ice_gfdl.nml, ocean_rough_nml, and input_ice_monin.nml: namelists for the GFDL module used by CICE to calculate surface fluxes (optionally created).
- input_ice.nml: CICE run and coupling control file (created).
- mom4_in.nml: MOM4 configuration (namelist) file (edited).
- MOM4_input/diag_table: MOM4 namelist specifying the diagnostic ocean outputs (edited).
- input_oce.nml: MOM4 run and coupling control file (created).

Useful tips:

- The sub-model configuration template files, especially mom4_in.nml provided in this AusCOM1.0 release, contain a number of parameters editable at run time by the runscript, namely, those strings prefixed with "#" need be replaced with sensible values when job is submitted. This flexible modelling approach encourages the tuning of physical parameterisations, in particular, that required by MOM4. Users who have similar intentions can create their own template to allow for more (or less) run time tuning freedom.
- In the sub-model coupling namelist files, users can find a number of parameters associated with run time coupling field checks. In input_ice.nml, for example, we have the default setting "chk_i2o_fields=.false.". If it is set to .true., CICE will record all the ice-to-ocean fields, at every coupling interval, in a NetCDF file named fields_i2o_in_ice.nc. This file, and similarly others if also written, can be very useful for checking if the coupling fields cause abnormal behaviour of the models, which happens quite often when a new experiment is started using new model code, new coupling algorithms, and/or new forcing datasets etc. These checks add computational expense and should be turned off (i.e. set to '.false.') once the AusCOM system is shown to be robust.

Now the preparation is completed, and the model is ready to go. Users thus do not need touch the remaining sections R-5 and R-6, but might like to familiarise themselves with these nevertheless.

## R-4. Launch/Execute the AusCOM Coupled Model on VAYU

```
mpirun  -n $nproc_cpl ./$oa3_exe : -n $nproc_ice ./$ice_exe : \
        -n $nproc_atm ./$atm_exe : -n $nproc_oce ./$oce_exe
```

## R-5. Postprocessing: Saving the Output Data

Upon the successful completion of a segment run, output data (history and restart) is ready to be written to the storage disk space. This section handles all output files: moving them to the destinition space and labelling them with proper model date information (end date of this run). Note, a PBS (copyq) queue job "do_mppncombine.ksh" is submitted here, requesting for 1 processor to combine the MOM4 history (NetCDF) files which are processor-dependent when written out by MOM4. Doing so can obviously speed up the post-processing operation, avoiding possible long-time holding of all the processors (e.g. 128 processors allocated for the current job) while only one of them is actually working. This is especially true for the XE machine, on which the MOM4 post-processing "mppnccombine" operation takes minutes.

## R-6. Submission of the next job

This section updates the exp.date file and submits the next job (if the simulation has not reached the final date as defined in Section R-1.)

That's all for the runscript. Users can try AusCOM out by submitting one of the sample jobs say, ciaf2-xx:

**$ qsub run_ausCOM.VAYU**

One last useful tip:

- When this runscript is adapted for a new experiment, which usually involves use of different input files, please run it first in interactive mode to check if all input files are in place, namely,

  **$ run_auscom.VAYU**

- If it reaches the model launching line "mpirun ", then go ahead and qsub it. Otherwise, check what's missing and do it again, and again... until everything is in place. The beneifit of doing so is obvious: it avoids wasting time waiting for the job to get on the PBS queue only to find (frustratingly again!) a few missing files.

# 6 AUSCOM QUICK-START ON NCI VAYU

For the quickest start possible, a short script, Setup_AusCOM1.0.ksh, is provided for users to set up their own independent, self-contained AusCOM1.0 system on the NCI VAYU platform. Under your home directory (recommended) or other places where you want to install the AusCOM1.0 system, run the script:

**$ /home/599/dhb599/Setup_AusCOM1.0.ksh**

It should only take 20 to 25 minutes to instal and compile the component models. The fully compiled AusCOM1.0 system takes about 427 MB of disk space. Now everything is in place, and the model is ready to run.

The following Sections (6.1-3) give a more detailed description of the steps taken in the installation script. The remainder of Section 6 gives advice on running the sample experiments provided with the package (6.4), monitoring the progress of runs (6.5), and finding the model outputs at the end of a run (6.6).

## 6.1 Getting the AusCOM1.0 Release Package

The whole package of the AusCOM version 1.0 release is a compressed tar-file named **AusCOM1.0.tar.gz**, about 92 MB, located at **VAYU:/home/599/dhb599/**. It contains the whole AusCOM system, except for the supported CORE forcing datasets (due to size limitations).

Copy it to your home directory (recommended), or somewhere on the /short disk, and unpack it:

**$ cp /home/599/dhb599/AusCOM1.0.tar.gz .**
**$ gunzip AusCOM1.0.tar.gz**
**$ tar xvf AusCOM1.0.tar**
**$ \rm AusCOM1.0.tar**
**$ cd AusCOM1.0/**
**$ ls**
**Doc bin exp forcing input output submodels**

- Doc/: you can find a file named AusCOM1.0_usersguide.pdf, which is this document.
- bin/: a few files are located here: environs.vayu.nci.org.au, do_mppncombine.ksh, calendar.F90.
- exp/: three sample experiments are prepared for you, i.e. ciaf2-xx, ciaf2-xy, and cnyf2-xx, with a README file describing these experiments.
- forcing/: only one small script file link_forcing.ksh is located here.
- input/: four sub-directories are here, containing the preprocessing auxiliary files for each of the components, some of which are “templates” for run-time adaption.

- output/: nothing is here until your experiment proceeds.

- submodels/: the full source code packages of the 4 components are located here.

## 6.2 Establishing Links to the Available Forcing Datasets

**$ cd AusCOM1.0/forcing/**
**$ ./link_forcing.ksh**
**$ ls -la**
**CIAF -> /short/p66/sjm599/COREv2_26JAN2010/CIAF**
**CNYF_v2 -> /short/p66/sjm599/COREv2_15JUNE2009/CNYF_v2**
**link_forcing.ksh**

We see the links point to the physical locations of the forcing data.

- CIAF: CORE2 interannual forcing.

- CNYF_v2: CORE2 normal year (i.e. climatology) forcing.

## 6.3 Building the component executables

Each component system tree has a directory named compile/, where the compilation script is located. All successfully generated executables will be automatically moved to AusCOM1.0/bin/.

- PSMILe libraries and OASIS3 executable

**$ cd AusCOM1.0/submodels/oasis3_prism_2-5/prism/compile/**
**$ comp_oasis325.VAYU**

This script compiles all the PSMILe libaries first and then the OASIS3 executable. Also generated is the calendar tool executable, i.e. calendar.VAYU.

- MATM executable

**$ cd AusCOM1.0/submodels/matm/compile/**
**$ comp_auscom_matm.VAYU**

- CICE executable

**$ cd AusCOM1.0/submodels/cice4.0/compile/**
**$ comp_auscom_cice.VAYU.np 6**

Note this builds the CICE executable running on 6 processors.

- MOM4 executable

**$ cd AusCOM1.0/submodels/mom4p1/compile/**
**$ comp_auscom_mom4.VAYU**

It also compiles the MOM4 history file combining tool mppnccombine.exe.

Congratulations.

You have just successfully set up the standard AusCOM1.0 system. Now you can test the sample runs, and hopefully soon design your own experiments for various applications.

## 6.4 Running A Sample Experiment

**$ cd AusCOM1.0/exp/**
**$ ls**
**ciaf2-xx ciaf2-xy ciaf2-xx**

Three sample experiments are available here for you to test:

- ciaf2-xx: CORE2 interannual forcing experiment with "cold" start, i.e. ocean and ice are initialised with the observed temperature and salinity only. It is set to start from date 19480101 and end on 19501231, with 1 year duration for each run segment.

- ciaf2-xy: CORE2 interannual forcing experiment with "warm" start, i.e. ocean and ice are initialised with an existing ocean-ice spinup. Same run time arrangement as in ciaf2-xx.

- cnyf2-xx: CORE2 normal year forcing experiment, cold start.

Please compare the runscripts run_auscom.VAYU under the job directories to see the setup differences. Make sure to replace project "p66" in the script with your own project name. If you are on p66, this script should work immediately for you. You can choose any one of them to start your very first AusCOM test run. For example,

First test-run the job interactively to test all necessary files are available (for reasons described in Section R-6 above):

**$ cd AusCOM1.0/exp/ciaf2-xx/**
**$ ls**
**run_auscom.VAYU**

Kill the interactive run when it reaches the model launching line "mpirun  "

Now submit the job to the PBS queue:

**$ qsub run_auscom.VAYU**

## 6.5 Monitoring the Run

**$ nqstat | grep $user**
**12345 R userid p66 ciaf2-xx .....**

When you see “R” in the above job status line, a symbolic link to the working directory ($rundir) has already been established here, called **Running.dir**. Also created is a log file **ciaf2-xx.log** which will record the run history of this experiment:

**$ ls**
**Running.dir  ciaf2-xx.log  run_auscom.VAYU**

You can monitor the model progress:

**$ less Running.dir/cplout**

or

**$ tail -f Running.dir/cplout**

Note the first couple of minutes see little progress because OASIS3 is calculating the huge interpolation coefficient matrices. After that the model will advance quickly. Normally it takes slightly more than 1 hour to complete a 1-year run.

Useful tip:

- The runscript will re-submit itself until the whole integration period of the experiment is completed, unless you wish to stop it by doing something like:

**$ mv run_auscom.VAYU run_auscom.VAYU_stop**

To resume the run, simply change the runscript name back.

## 6.6 Verifying the Outputs

A symobolic link to the phyical location of the model output storge has now been built under the AusCOM home:

**$ ls AusCOM1.0/output/**
**ciaf2-xx**

When the first run is completed, you can check (and/or process) the outputs of the ice and ocean models:

**$ cd AusCOM1.0/output/ciaf2-xx/history/mom4/**
**$ ls**
**ocean_layer.nc-19481231  ocean_param.nc-19481231  ocean_surf.nc-19481231**
**ocean_month.nc-19481231  ocean_snap.nc-19481231  ocean_trans.nc-19481231**

**$ cd AusCOM1.0/output/ciaf2-xx/history/cice/**
**$ ls**
**iceh.1948-01.nc  iceh.1948-04.nc  iceh.1948-07.nc  iceh.1948-10.nc**
**iceh.1948-02.nc  iceh.1948-05.nc  iceh.1948-08.nc  iceh.1948-11.nc**
**iceh.1948-03.nc  iceh.1948-06.nc  iceh.1948-09.nc  iceh.1948-12.nc**

# 7 ACKNOWLEDGMENTS

This work has been undertaken as part of the Australian Climate Change Science Program, funded jointly by the Department of Climate Change and Energy Efficiency, the Bureau of Meteorology and CSIRO. We are grateful for super-computing support from the National Computing Infrastructure (NCI) under projects p66 and p73. AusCOM development greatly benefitted from many helpful discussions with international collaborators and scientists at NOAA/GFDL, LANL, CERFACS, and the UKMO. We acknowledge contributions to the development of AusCOM made by: the CAWCR ACCESS Coupled and Ocean Modelling Team; Xiaobing Zhou at BoM; Michael Bates at UNSW; Nathan Bindoff, Jason Roberts, Petra Heil, and Roger Stevens at the University of Tasmania and Antarctic Climate and Ecosystems Cooperative Research Centre; collaborators in the CAWCR/BoM Seasonal Prediction Group; Russ Fiedler, Richard Matear, Matthew Chamberlain and Andrew Lenton at CSIRO Marine and Atmospheric Research, Hobart; and the many others involved in ACCESS model development.

# Notice of usage conditions:

New users are requested to contact Dave Bi or Simon Marsland prior to usage of the model. This is in order to maintain a user data base, by which users will be notified of upgrades, bug fixes, etc. Users are requested to cite this User Guide in publications which include results from AusCOM1.0. This standard reference for AusCOM1.0 may be updated in the future, and users on the database will be advised.



# 8 REFERENCES

Adcroft, A. and Campin, J.-M. 2004. Rescaled height coordinates for accurate representation of free-surface flows in ocean circulation models. Ocean Modelling, 7, 269–84.

Bi, D., Marsland, S.J., Collier, M. and Griffies, S.M. 2010. Benchmarking the Australian Climate Ocean Model (AusCOM) with a coordinated ocean-ice reference experiment. CAWCR CSIRO Marine and Atmospheric Research, CAWCR Technical Report No.xx. in press.

Griffies, S.M. 2004. Fundamentals of ocean climate models. Princeton University Press, Princeton, USA, 496pp.

Griffies, S.M., Biastoch, A., Boening, C.W., Bryan, F., Chassignet, E., England, M., Gerdes, R., Haak, H., Hallberg, R.W., Hazeleger, W., Jungclaus, J., Large, W.G., Madec, G., Samuels, B.L., Scheinert, M., Gupta, A.S., Severijns, C.A., Simmons, H.L., Treguier, A.M., Winton, M., Yeager, S. and Yin, J. 2009. Coordinated ocean-ice reference experiments (COREs). Ocean Modelling, 26, 1–46, doi:10.1016/j.ocemod.2008.08.007.

Griffies, S.M., Harrison, M.J., Pacanowski, R.C. and Rosati, A.2004. A Technical guide to MOM4. NOAA/Geophysical Fluid Dynamics Laboratory, GFDL Ocean Group Technical Report No. 5, 337pp.

Heil, P., Phipps, S.J. and Roberts, J.L. 2005. User guide for the TPAC coupled ocean-sea ice model. TPAC Technical Report , 45pp.

Hunke, E.C., 2001. Viscous-plastic sea ice dynamics with the EVP model: Linearization issues. J. Comput. Phys., 170, 19–38.

Hunke, E.C. and Dukowicz, J.K. 1997. An elastic-viscous-plastic model for sea ice dynamics. Journal of Physical Oceanography, 27, 1849–67.

Hunke, E.C. and Lipscomb, W.H. 2008. CICE: the Los Alamos sea ice model documentation and software user's manual version 4.0. LA-CC-06-012, 72pp.

Jones, P.W. 1997. A User's Guide for SCRIP: A spherical coordinate remapping and interpolation package. LA-CC Number 98-45, 27pp.

Large, W. and Yeager, S. 2004. Diurnal to decadal global forcing for ocean and sea-ice models: the data sets and flux climatologies. CGD Division of the National Center for Atmospheric Research, NCAR Technical Note: NCAR/TN-460+STR.

Large, W.G. and Yeager, S. 2009. The global climatology of an interannually varying air-sea flux data set. Climate Dynamics, 33, doi:10.1007/s00382-008-0441-3.

Murray, R.J. 1996. Explicit generation of orthogonal grids for ocean models. Journal of Computational Physics, 126, 251–73.

Roberts, J.L., Heil, P., Phipps, S.J. and Bindoff, N.L. 2007. AusCOM: The Australian community ocean model. Journal of Research and Practice in Information Technology, 39(2), 137–150.

Valcke, S. 2006. OASIS3 User Guide. PRISM-Support Initiative, Report No. 3, 64pp.

Winton, M. 2001. FMS sea ice simulator. NOAA/Geophysical Fluid Dynamics Laboratory, 11pp.

# APPENDIX

# A A SAMPLE OASIS3 COUPLING CONFIGURATION FILE NAMCOUPLE

```
# This is a typical input file for OASIS 3.0, using netCDF format
# for restart input files. Oasis reads this file at run-time.
#
# Any line beginning with # is ignored. Blank lines are not allowed.
#
  $SEQMODE
# This keyword concerns the coupling algorithm. Put here the maximum number
# of fields that have to be, at one particular coupling timestep,
# necessarily exchanged sequentially in a given order.
         1
  $END
#############################################################################
  $CHANNEL
# The communication technique you want to use.
# Choices are MPI1 or MPI2, NONE, SIPC, GMEM.
# - if you use MPI1 or MPI2 message passing (CLIM library based on
# MPI1 or MPI2), you must write MPI1 or MPI2 on one line
# + one line per model giving for the model the total number of procs,
# the number of procs implied in the coupling and, for MPI2 only, an
# optional launching argument
#
MPI1 NOBSEND
6 1
1 1
120 1
$END ########################################################################
  $NFIELDS
# This is the total number of fields being exchanged.
### 10 fields atm -> ice
### 13 fields ice -> ocn
### 7 fields ocn -> ice
### 1 fields ice -> atm
        31
  $END ##############################################################################
  $JOBNAME
# This is an acronym for this run.
# (3 characters)
     OIA
  $END
#############################################################################
  $NBMODEL
# This gives you the number of models running in this experiment +
# their names (character*6, exactly!) + , in option, the maximum Fortran unit
# number used by each model; 1024 will be used if none are given.
#
```

```
  3    cicexx    matmxx    mom4xx    99    99    99
  $END
#############################################################################
  $RUNTIME
# This gives you the total simulated time for this run in seconds
#
# 86400 ! 1 day
# 2678400 ! 31 days
 31536000
 $END
#############################################################################
  $INIDATE
# This is the initial date of the run. This is important only if
# FILLING analysis is used for a coupling field in the run.
# The format is YYYYMMDD such as 19910101
  19500101
  $END
#############################################################################
  $MODINFO
# Indicates if a header is encapsulated within the field brick
# in binary restart files for all communication techniques, and
# for coupling field exchanges for PIPE, SIPC and GMEM.
# (YES or NOT)
    NOT
  $END
#############################################################################
  $NLOGPRT
# Index of printing level in output file cplout: 0 = no printing
# 1 = main routines and field names when treated, 2 = complete output
 1
 $END
#############################################################################
$CALTYPE
# Calendar type :     0 = 365 day calendar (no leap years)
#                          1 = 365 day, or 366 days for leap years, calendar
#                          n (>1) = n day month calendar
# This is important only if FILLING analysis is used for a coupling
# field in the run.
#
   0
   $END
#############################################################################
   $STRINGS
#
# The above variables are the general parameters for the experiment.
# Everything below has to do with the fields being exchanged.
#
#####################################################################
# --- note file cf_name_table.txt does not include all the coupling flds ---
# --- listed below. so the fld number (367 for swfld etc) may not point ---
# --- to the realistic 'longname' of the variable. (may modify this file ---
# --- to add more fields in the future.)
#####################################################################
```

```
#
#                          ATMOSPHERE    --->>>    ICE
#                          ------------------------------------------
##########
# Field 01 : swflx down
########## swfld_ai swfld_i 367 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
#
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0
CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
#########
# Field 02 : lwflx down
##########
lwfld_ai lwfld_i 366 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
#
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0
CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
##########
# Field 03 : rainfall
##########
rain_ai rain_i 26 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
#
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0 CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
##########
# Field 04 : snowfall
########## snow_ai snow_i 26 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
#
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0 CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
##########
# Field 05 : surface pressure
##########
press_ai press_i 33 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
```

```
#
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0
CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
##########
# Field 06 : runoff
##########
runof_ai runof_i 297 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
 #
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0
CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
##########
# Field 07 : near surface (2m) air temp
##########
tair_ai tair_i 110 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
#
LOCTRANS CHECKIN SCRIPR CHECKOUT INSTANT
INT=0
CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
##########
# Field 08 : 2m air humidity
##########
qair_ai qair_i 339 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
#
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0
CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
##########
# Field 09 : 10m wind (u)
##########
#uwnd_ai uwnd_i 56 21600 3 a2i.nc EXPORTED
#nt62 cice LAG=+3600 SEQ=+1
#P 0 P 0
##
#CHECKIN SCRIPR CHECKOUT
##
# INT=1
# DISTWGT LR VECTOR_I LATLON 10 4 vwnd_ai
# INT=1
```

```
#------------------------------------------------
uwnd_ai uwnd_i 56 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
#
LOCTRANS CHECKIN INTERP CHECKOUT
INSTANT
INT=0
BICUBIC    G    SCALAR
INT=0
##########
# Field 10 : 10m wind (v)
##########
#vwnd_ai vwnd_i 56 21600 3 a2i.nc EXPORTED
#nt62 cice LAG=+3600 SEQ=+1
#P 0 P 0
##
#CHECKIN SCRIPR CHECKOUT
##
# INT=1
# DISTWGT LR VECTOR_J LATLON 10 4 uwnd_ai
# INT=1
#------------------------------------------------
vwnd_ai vwnd_i 56 21600 4 a2i.nc EXPORTED
nt62 cice LAG=+3600 SEQ=+1
P 0 P 0
#
LOCTRANS CHECKIN INTERP CHECKOUT
INSTANT
INT=0
BICUBIC G SCALAR
INT=0 ###############################################################################
#
#                                  ICE --->>> OCEAN
#                                  --------------------------
##########
# Field 11 : ice-ocean interface stress (x-direction)
##########
strsu_io strsu_o 170 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 12 : ice-ocean interface stress (y-direction)
##########
strsv_io strsv_o 175 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 13 : freshwater flux
```

```
########## rain_io rain_o 27 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 14 : freshwater flux
##########
snow_io snow_o 28 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 15 : salt flux (no ref no for saltflux yet!)
##########
stflx_io stflx_o 454 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 16 : next heat flux (heatflux into ocean. '42' not right.)
##########
htflx_io htflx_o 42 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 17 : swflux penetrating through ice into ocean
#########
swflx_io swflx_o 367 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 18 : latent heat flux
##########
qflux_io qflux_o 452 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
#########
# Field 19 : Sensible heat flux
##########
shflx_io shflx_o 362 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
```

```
##########
# Field 20 : LW radiation flux down
##########
lwflx_io lwflx_o 366 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 21 : runoff
##########
runof_io runof_o 297 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 23 : surface pressure
##########
press_io press_o 33 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS INSTANT
##########
# Field 23 : ice concentration
##########
aice_io aice_o 44 3600 1 i2o.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
############################################################################ #
#                                    OCEAN --->>> ICE
#                                    ------------------------
##########
# Field 24 : Sea surface temperature (Celsius in MOM4)
##########
sst_oi sst_i 1 3600 1 o2i.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 25 : Sea surface salinity (psu)
########## sss_oi sss_i 312 3600 1 o2i.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 26 : 'eastward' sea surface water velocity
##########
ssu_oi ssu_i 181 3600 1 o2i.nc IGNORED
```

```
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 27 : 'northward' sea surface water velocity
##########
ssv_oi ssv_i 261 3600 1 o2i.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 28 :         frazil ice fromation energy (J/m^2)
#                    ('441' is "upward_sea_ice_basal_heat_flux' 'W m-2')
##########
#pfmice_oi pfmice_i 441 3600 1 o2i.nc IGNORED
ptice_oi pfmice_i 441 3600 1 o2i.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 29 :           sea surface slope _x_ (m/m)
#                      ('203' is "height" ...)
##########
sslx_oi sslx_i 203 3600 1 o2i.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
##########
# Field 30 :           sea surface slope _y_ (m/m)
#                      ('310' is "sea_surface_elevation")
##########
ssly_oi ssly_i 310 3600 1 o2i.nc IGNORED
cice cice LAG=0 SEQ=+1
#
LOCTRANS
INSTANT
############################################################################ #
#                              ICE --->>> ATMOSPHERE
#                              ----------------------------------
##########
# Field 31 : ice/ocean surface temp (no ref no! '331' is snow temp!)
##########
isst_ia isst_a 331 21600 4 i2a.nc EXPORTED
cice nt62 LAG=+3600 SEQ=+1
P 0 P 0
#
LOCTRANS CHECKIN SCRIPR CHECKOUT
INSTANT
INT=0
```

```
CONSERV LR SCALAR LATLON 10 FRACNNEI FIRST
INT=0
##############################################################################
  $END
```

# B A SAMPLE AUSCOM RUNSCRIPT

```
#!/bin/ksh
#################################################################################     #
run_auscom.VAYU                                                                       #
#################################################################################     #
*** Set up the running environment and run the auscom coupled model        *** #
#
#     'AusCOM' is a coupled ocean and sea ice model consisting of 3 components
#          1. matm (a data atmospheric model, providing atmospheric forcing)
#          2. cice4.0 (LANL sea ice model)
#          3. mom4p1 (GFDL ocean model)
#          built under the OASIS3 PRISM_2-5 framework
#
#     This sample run is set up on the NCI VAYU platform for 128 processes:
#     1 for cpl, 1 for matm, 6 for cice, and 120 for mom4 (mono-cpu cpling)
################################################################################# #
# 0. Prologue
#       ###############################################################################
#PBS -P p66
#PBS -q normal
#PBS -l walltime=1:40:00
#PBS -l vmem=128GB
#PBS -l ncpus=128
#PBS -l software=vampir
#PBS -l other=rms
###PBS -M dave.bi@csiro.au
#PBS -N ciaf2-xx
#PBS -wd

date
set -e
set -xv ulimit -s unlimited
ulimit -a

#
#-- Export System depending variables
#
export MPIPROGINF=DETAIL;
export F_PROGINF=detail;
export F_FTRACE=YES;
export MPLARGS=" " ;
export F_SETBUF06=50000
export F_SETBUF07=50000
export F_SETBUF08=50000
export ATM_COMPARAL=1
export F_SYSLEN=300
export F_SETBUF00=50000
MPIEXPORT="F_PROGINF F_SYSLEN"
MPIEXPORT="${MPIEXPORT} MPIPROGINF F_FTRACE MPISUSPEND"
MPIEXPORT="${MPIEXPORT} F_SETBUF00 F_SETBUF06 F_SETBUF07 F_SETBUF08"
export MPI_MULTITASKMIX="ON"
```

```
export MPIEXPORT="${MPIEXPORT} MPI_MULTITASKMIX"
export MPI_BUFFER_MAX=5000000
############################################################################ #
# 1. Primary Setups
# ############################################################################ #
## 1.1 Define experiment ID etc.
#
project=p66 # /short disk 'owned' by project (e.g. p66)
jobid=$PBS_JOBID # job-id assigned by PBS (the queue sys)
job=$PBS_JOBNAME # name of this script
chan=MPI1 # Message Passage (MPI1/MPI2)

expid=ciaf2-xx # change expid for each new experiment
atmdata=core2 # choose the atmospheric forcing dataset
atm_forcing="'${atmdata}'" # (ncep2, era40, core2, or um96 etc.)
datatype=IA # NY/IA: Normal Year/Interannual Annual
year_data_end=2007 # data NOT available after this year

#
## 1.2 Define all associated paths
#

# Location where jobs are submitted (and this script is located):
cd `pwd`/../..
AusCOMHOME=`pwd`
model=${AusCOMHOME##*/} #the model name, i.e. AusCOM1.0
jobdir=$AusCOMHOME/exp/$expid

# Location of preprocessed input files for the coupled model:
inputdir=$AusCOMHOME/input

# Location where the model executables are stored:
bindir=$AusCOMHOME/bin

# Location where outputs are to be stored:
datahome=/short/$project/$USER/OUTPUT
outputdir=$datahome/$model/$expid
restdir=${outputdir}/restart
histdir=${outputdir}/history

# Location where the sub-models and coupler are actually runing:
workhome=/short/$project/$USER
rundir=$workhome/RUNNING/$model/$expid

############################################################################ #
# 2. Exp/Run Time Control etc.
# ############################################################################ #
## 2.1 Runtime control for the whole exp and this segment run
#

# Initial and final date of the experiment
iniyear=1948; finalyear=1950; typeset -Z4 iniyear finalyear
inimonth=1; finalmonth=12; typeset -Z2 inimonth finalmonth
```

```
iniday=1; finalday=31; typeset -Z2 iniday finalday

# Duration of this run (maybe the most often visited place for test/short runs):
nyear=0    # number of years (ALWAYS 0 ! change nmonth etc...)
nmonth=12 # number of months
nday=0 # number of days
#nmonth=0 # if not a whole month. set length "nday=xx" below
#nday=5 # days of this run. important for quick tests

# Time steps
dt_cpl_ai=21600 #air-ice coupling interval in seconds
dt_cpl_io=3600 #ice-ocn coupling interval in seconds
dt_oce=3600 #oce model timestep
dt_atm=3600 #atm model timestep
dt_ice=3600 #ice model timestep

#
## 2.2 Processor usage for this run #

# Processor for each executable:
nproc_cpl=1     #always 1
nproc_atm=1    #           1
nproc_ice=6      #changable
nproc_oce=120  #changable

# Total number of procs for this job (must <= requested in the #PSB line):
(( ntproc = nproc_atm + nproc_ice + nproc_oce + nproc_cpl ))

# Currently AusCOM is hardwired for mono-cpu coupling:
ncplproc_atm=1
ncplproc_ice=1
ncplproc_oce=1

# Decide ocean domain MPI partitioning pattern:
oce_nx=8; oce_ny=15 #oce_nx x oce_ny = nproc_oce

#
## 2.3 Names of the 4 executables
#

oa3_exe=oasis3
atm_exe=matmxx #These 3 sub-model exe names must be same as
ice_exe=cicexx #defined in namcouple and model code
oce_exe=mom4xx #(character*6)

#
## 2.4 Calendat date control
#
#-- Calendar type: available calendar options:
#           0 : No leap year (365 days per year)
#           1 : Gregorian (365/366 days per year)
#           n : Equal months of "n" days (30 for 30 day months)
```

```
# Default set as below (for era40 and ncep2 forcing)
caltype=1                    #0, 1 or n (eg, 30 for 30 day months)
cal_type="'julian'"          #for caltype=1; "'thirty_day'" for caltype=30.

# For core and core2 forcing we must use below:
if [[ $atm_forcing = "'core'" || $atm_forcing = "'core2'" ]]; then
    caltype=0
    cal_type="'NOLEAP'"
fi

# Dates in format YYYYMMDD:
inidate=${iniyear}${inimonth}${iniday}
finaldate=${finalyear}${finalmonth}${finalday}

cd $jobdir

typeset -Z4 year; typeset -Z2 month day
if [ ! -f ${expid}.date ]; then
    year=${iniyear} #
    month=${inimonth} #
   day=${iniday} #
   jobnum=1 # 1 for initial run, >1 for continue runs
   truntime0=0 # total accumulated runtime by the end of last run
   if [ -f ${expid}.log ]; then
   rm ${expid}.log
   fi
   echo "`date` : Beginning of Experiment ${expid}" > ${expid}.log
else
   read year month day jobnum truntime0 < ${expid}.date
fi

date=${year}${month}${day}
echo " " >> ${expid}.log
echo "`date` : ${jobnum} ${date} - starting pre-processing" >> ${expid}.log

# use an external tool to work out the run date inforamtion.
# there must be a easier way for this. but it works ok anyway.

cat > calendar.in << EOF
    ${inidate} ${date} ${nyear} ${nmonth} ${nday} ${caltype}
EOF

$AusCOMHOME/bin/calendar.exe < calendar.in > calendar.out

prevdate=`cat calendar.out | cut -c '2-9'`
enddate=`cat calendar.out | cut -c '11-18'`
nextdate=`cat calendar.out | cut -c '20-27'`
previnidate=`cat calendar.out | cut -c '29-36'`
days_in_run=`cat calendar.out | cut -c '38-45'`
days_since_start=`cat calendar.out | cut -c '47-54'`
date_in_days=`cat calendar.out | cut -c '56-63'`
days_this_year=`cat calendar.out | cut -c '65-67'`
rm calendar.in calendar.out
```

```
prevyear=‘ echo "$prevdate" | cut -c ’1-4’‘
prevmonth=‘ echo "$prevdate" | cut -c ’5-6’‘
prevday=‘ echo "$prevdate" | cut -c ’7-8’‘
endyear=‘ echo "$enddate" | cut -c ’1-4’‘
endmonth=‘ echo "$enddate" | cut -c ’5-6’‘
endday=‘ echo "$enddate" | cut -c ’7-8’‘
nextyear=‘ echo "$nextdate" | cut -c ’1-4’‘
nextmonth=‘ echo "$nextdate" | cut -c ’5-6’‘
nextday=‘ echo "$nextdate" | cut -c ’7-8’‘
previniyear=‘ echo "$previnidate" | cut -c ’1-4’‘
previnimonth=‘ echo "$previnidate" | cut -c ’5-6’‘
previniday=‘ echo "$previnidate" | cut -c ’7-8’‘

echo ""
echo "first day of this run: ${date}"
echo "last day of this run: ${enddate}"
echo "initial date of the experiment: ${inidate}"
echo "final date of the experiment: ${finaldate}"
echo "day before the initial date: ${previnidate}"
echo "last day of the previous run: ${prevdate}"
echo "first day of the next run: ${nextdate}"
echo ""
echo "number of days in this run: ${days_in_run}"
echo "number of days since beginning of the experiment: ${days_since_start}"
echo ""
(( runtime = ${days_in_run} * 86400 )) #duration of this run in seconds
echo "duration of this run in seconds: ${runtime}"
 ############################################################################ #
# 3. Getting All Files into the Run Directory
# ############################################################################ #
#
# 3.1 Grids, IC, forcing, executables and some preprocessed auxiliary files
#

# following setup needs two things be decided first:

boundary_layer=gfdl # <==how to calculte surface fluxes decided here
runoff_data=core # <==regrided core rundoff instead of that interpolated by oasis
cold_start=1 # 1/0, this experiment starts from ’scratch’/spinup
if [ $cold_start = 0 ]; then #ie, warm start
    # use existing AusCOM run restart to initialise this experiment (for jobnum=1)
    # * next 4-5 lines specify the restart location ----------#
    owner=dhb599
    expname=cnyf2-01
    ic_year=500
    access=~$owner #if run under $HOME
    #access=/short/p66/$owner #if run under /short/p66 disk
    #----------------------------------------------------------#
    typeset -Z4 ic_year
    (( ic_yearp1 = $ic_year + 1 ))
    typeset -Z4 ic_yearp1
    rest_date_oasis=${ic_year}1231
    rest_date_mom4=${ic_year}1231
```

```
    rest_date_cice=${ic_yearp1}0101
    ic_location=$access/$model/output/$expname/restart
    mom4_ic=$ic_location/mom4
    cice_ic=$ic_location/cice
    oasis_ic=$ic_location/oasis3
fi

if [ $jobnum = 1 ]; then #initial run

    rm -rf $restdir
    rm -rf $histdir
    rm -rf $AusCOMHOME/output/$expid
    mkdir -p $restdir/cice $restdir/mom4 $restdir/oasis3
    mkdir -p $histdir/cice $histdir/mom4
    ln -s $outputdir $AusCOMHOME/output/.

rm -fr $rundir;
mkdir -p $rundir
ln -s $rundir $jobdir/Running.dir
cd $rundir

mkdir MATM_input #subdirs for MATM
mkdir CICE_input CICE_restart CICE_hist #subdirs for CICE
mkdir MOM4_input MOM4_restart MOM4_hist #subdirs for MOM4

# get the executables:
cp -f $bindir/oasis3_$chan.exe      $oa3_exe
cp -f $bindir/mom4_$chan.exe      $oce_exe
cp -f $bindir/cice_$chan.exe_${nproc_ice}p      $ice_exe
cp -f $bindir/matm_$chan.exe_nt62      $atm_exe

# get input files for oasis3:

# a. ref and grids data
cp -f $inputdir/oasis3/cf_name_table.txt .
cp -f $inputdir/oasis3/grids_cice_nt62.nc grids.nc
cp -f $inputdir/oasis3/masks_cice_nt62.nc masks.nc
cp -f $inputdir/oasis3/areas_cice_nt62.nc areas.nc

# b. restart
if [ $cold_start = 1 ]; then          #cold start
    # the pre-processed coupling restart files:
    cp -f $inputdir/oasis3/a2i_10fields.nc a2i.nc
    cp -f $inputdir/oasis3/o2i_7fields.nc o2i.nc
    cp -f $inputdir/oasis3/i2o_13fields.nc i2o.nc
    cp -f $inputdir/oasis3/i2a_1fields.nc i2a.nc
else #warm start
    # rstart from an existing run (spinup)
    cp -f $oasis_ic/a2i.nc-$rest_date_oasis a2i.nc
    cp -f $oasis_ic/o2i.nc-$rest_date_oasis o2i.nc
    cp -f $oasis_ic/i2o.nc-$rest_date_oasis i2o.nc
    cp -f $oasis_ic/i2a.nc-$rest_date_oasis i2a.nc
fi
```

```
# get input files for matm (to be done in section 4)

# input files for cice:
if [ $runoff_data = core ]; then
    cp -f $inputdir/cice/core_runoff_regrid.nc CICE_input/.
fi

# a. grid data and surface
cp -f $inputdir/cice/cice_grid.nc CICE_input/grid.nc
cp -f $inputdir/cice/cice_kmtu.nc CICE_input/kmt.nc

# b. IC/restart
if [ $cold_start = 1 ]; then #cold start
    runtype="'initial'"; Lrestart=.false.; ice_ic="'default'"
    cp -f $inputdir/cice/A2I_time0_10fields.nc CICE_input/A2I_time0.nc
    cp -f $inputdir/cice/SSTS_12Jans.nc CICE_input/monthly_sstsss.nc
    if [ $boundary_layer = gfdl ]; then
        cp -f $inputdir/cice/uu_star_t0.nc CICE_input/u_star.nc
    fi
else #warm start
    runtype="'continue'"; Lrestart=.true.; ice_ic="'default'"
    ice_restart=${cice_ic}/iced.$rest_date_cice
    cp -f $ice_restart CICE_restart/iced
    echo iced > CICE_restart/ice.restart_file
    cp -f ${cice_ic}/A2I_time1.nc-$rest_date_oasis CICE_input/A2I_time0.nc
    if [ $boundary_layer = gfdl ]; then
        cp -f ${cice_ic}/u_star.nc-$rest_date_oasis CICE_input/u_star.nc
    fi
fi

# get input files for mom4:
 cd MOM4_input
cp -f $inputdir/mom4/field_table field_table
cp -f $inputdir/mom4/data_table data_table
cp -f $inputdir/mom4/grid_spec.auscom.nc grid_spec.nc
cp -f $inputdir/mom4/salt_sfc_restore.nc salt_sfc_restore.nc
cp -f $inputdir/mom4/temp_sfc_restore.nc temp_sfc_restore.nc
if [ $cold_start = 1 ]; then
    #get ocean initial condition (only T-S)
    cp -f $inputdir/mom4/ocean_temp_salt.nc ocean_temp_salt.res.nc
else
    for restfile in `ls ${mom4_ic}/ocean_*-${rest_date_mom4}`; do
        newfile=${restfile##*/}
        cp ${restfile} ${newfile
done
    ystart=$ic_yearp1
    if [ $ystart -lt 10 ]; then
        typeset -Z1 ystart
    elif [ $ystart -lt 100 ]; then
        typeset -Z2 ystart
    elif [ $ystart -lt 1000 ]; then
       typeset -Z3 ystart
```

```
    elif [ $ystart -lt 10000 ]; then
        typeset -Z4 ystart
    fi
ed ocean_solo.res <<eof
g/$ystart/s/$ystart/${iniyear}/
w
q
eof
    fi #cold_start

else #for continue runs

    cd $rundir
    rm -f *out* ?weights *.prt* #clean up

#prepare restart files:

# for oasis3:
for resfile in `ls $restdir/oasis3/?2?.nc-${prevdate}`; do
    sresfile=${resfile##*/} #take away the front path name
    cp $resfile ${sresfile %-*} #take away the appendix '-YYYYMMDD'
done

# for cice:
runtype="'continue '"; Lrestart=.true.; ice_ic="'default '"
cp $restdir/cice/A2I_time1.nc-${prevdate} CICE_input/A2I_time0.nc
cp $restdir/cice/ice.restart_file-${prevdate} CICE_restart/ice.restart_file
cp $restdir/cice/`cat CICE_restart/ice.restart_file` CICE_restart/.
cp $restdir/cice/u_star.nc-${prevdate} CICE_input/u_star.nc

# for mom4:
for restfile in `ls $restdir/mom4/ocean*-${prevdate}`; do
    ncfile=${restfile##*/}
    cp $restfile MOM4_input/${ncfile%-*}
done

fi #initial or continue run

# prepare the atm_forcing dataset needed for this run:
cd $rundir/MATM_input
typeset -Z4 y1 y2
y1=$endyear
y2=$endyear
if [ $datatype = NY ]; then
y1=1                    #for 'NY' forcing, y1 should be always '0001' !
fi
$inputdir/matm/get_${atmdata}_${datatype}.ksh $y1 $y2 $AusCOMHOME
y2=`expr $endyear + 1`
y1=$y2
if [ $datatype = NY ]; then
    y1=1
fi
if [ $endyear = $year_data_end ]; then
```

```
    y1=$endyear
fi
$inputdir/matm/get_${atmdata}_${datatype}.ksh $y1 $y2 $AusCOMHOME

#
## 3.2 Adapting or creating configuration files
#

# 3.2.1 namelist for oasis3:

nlogprt=1 #cplout writing control: 0-no, 1-medium, 2-full output
npt1=${nproc_ice}; npc1=${ncplproc_ice}; arg1=$ice_exe; nam1=$ice_exe
npt2=${nproc_atm}; npc2=${ncplproc_atm}; arg2=$atm_exe; nam2=$atm_exe
npt3=${nproc_oce}; npc3=${ncplproc_oce}; arg3=$oce_exe; nam3=$oce_exe

#-- buffered MPI Send for coupling communication
#            yes: buffered send       (for MPI, or MPI2 without 'mailbox')
#             no: simple send         (for MPI2 with big enough 'mailbox')
#-------------------------------------------------------------------------------
#bsend=yes         # needs larger buffer size for MPI_Bsend operation: to make it work,
                   # we've doubled "il_bufsendsize" in oasis3/src/inicmc.F!
bsend=no # this one works fine, and is recommended!

if [ ${bsend} = no ]; then
    nobsend="NOBSEND"
else
    nobsend=""
fi
if [ $chan = 'MPI1' ]; then
    arg1=""; arg2=""; arg3=""
fi

#
# get and adapt file namcouple
#

cd $rundir
cp -f $inputdir/oasis3/namcouple_31fields namcouple
ed namcouple <<eof
g/#Channel/s/#Channel/${chan} ${nobsend}/
g/#Mod1procs/s/#Mod1procs/ $npt1 $npc1 $arg1 /
g/#Mod2procs/s/#Mod2procs/ $npt2 $npc2 $arg2 /
g/#Mod3procs/s/#Mod3procs/ $npt3 $npc3 $arg3 /
g/#Mod1_name/s/#Mod1_name/ $nam1 /
g/#Mod2_name/s/#Mod2_name/ $nam2 /
g/#Mod3_name/s/#Mod3_name/ $nam3 /
g/#Runtime_sec/s/#Runtime_sec/${runtime}/
g/#Inidate/s/#Inidate/${date}/
g/#Caltype/s/#Caltype/${caltype}/
g/#NLOGPRT/s/#NLOGPRT/${nlogprt}/
g/#CPL_intv_ai/s/#CPL_intv_ai/${dt_cpl_ai}/ g/#CPL_intv_io/s/#CPL_intv_io/${dt_cpl_io}/
g/#DT_OCE/s/#DT_OCE/${dt_oce}/
g/#DT_ATM/s/#DT_ATM/${dt_atm}/
```

```
g/#DT_ICE/s/#DT_ICE/${dt_ice}/
w
q
eof

# 3.2.2 namelist for matm coupling

cat > input_atm.nml << eof
&coupling
    init_date=${iniyear}${inimonth}${iniday}
    inidate=$date
    truntime0=$truntime0
    runtime=$runtime
    dt_cpl=$dt_cpl_ai
    dt_atm=$dt_atm
    dataset=$atm_forcing
    runtype='${datatype}'
    caltype=$caltype
    days_per_year=$days_this_year
    chk_a2i_fields=.false.
    chk_i2a_fields=.false.
    &end
    eof

# get and adapt the forcing pointer file:
cp -f $inputdir/matm/${atmdata}_fields_$datatype.table data_4_matm.table
ed data_4_matm.table <<eof
g/#YEAR/s/#YEAR/$endyear/
g/#FORCING/s/#FORCING/MATM_input/
w
q
eof

# 3.2.3 namelists for cice

# a. standalone mode input
#
npt_cice=‘expr $runtime / $dt_ice‘
if [ $nmonth != 0 ]; then           #ie, nmonth=1, a whole month run
    histfreq="'m'"; hist_avg=.true.; dumpfreq="'m'"; dumpfreq_n=$nmonth
else #ie, nmonth=0, an nday run
    histfreq="'d'"; hist_avg=.true.; dumpfreq="'d'"; dumpfreq_n=$nday
fi #hist_avg=.false. would output snapshot hist
mixedocean=.false. #use or not use the mixed ocean layer

cp -f $inputdir/cice/cice4.0_in.nml cice_in.nml

ed cice_in.nml <<eof
g/#DAYS_per_year/s/#DAYS_per_year/${days_this_year}/
g/#YEAR_init/s/#YEAR_init/${iniyear}/
g/#DT_CICE/s/#DT_CICE/${dt_ice}/
g/#NPT/s/#NPT/${npt_cice}/
g/#RUNTYPE/s/#RUNTYPE/${runtype}/
```

```
g/#HISTFREQ/s/#HISTFREQ/${histfreq}/
g/#HIST_AVG/s/#HIST_AVG/${hist_avg}/    g/#DUMPFREQ/s/#DUMPFREQ/${dumpfreq}/
g/#DUMPFR_N/s/#DUMPFR_N/${dumpfreq_n}/ g/#RESTART/s/#RESTART/${Lrestart}/
g/#ICE_IC/s/#ICE_IC/${ice_ic}/
g/#FYEAR_init/s/#FYEAR_init/${iniyear}/  g/#MIXEDOCN/s/#MIXEDOCN/${mixedocean}/
g/#NPROCS/s/#NPROCS/${nproc_ice}/
w
q
eof

# b. namelist for coupling purpose

POP_ICEDIAG=' .true. '#use POP approach for ice formation/melting
GFDL_FLUXES='.false. '
if [ $boundary_layer = gfdl ]; then
    GFDL_FLUXES='.true. ' #use GFDL code for surface flux calculation
    cat > input_ice_gfdl.nml << eof
&surface_flux_nml
  no_neg_q = .false.
  use_virtual_temp = .true.
  alt_gustiness = .false.
  old_dtaudv = .false.
  use_mixing_ratio = .false.
  gust_const = 1.0
  gust_min = 0.0
  ncar_ocean_flux = .true.
  ncar_ocean_flux_orig = .false.
  raoult_sat_vap = .false.
/
&ocean_rough_nml
  roughness_mom = 5.8e-5
  roughness_heat = 5.8e-5
  roughness_moist = 5.8e-5
  roughness_min = 1.0e-6
  charnock = 0.032
  rough_scheme = 'beljaars'
  do_highwind = .false.
  do_cap40 = .false.
  zcoh1 = 0.0
  zcoq1 = 0.0
/
eof
cat > input_ice_monin.nml << eof
&monin_obukhov_nml
neutral=.true.
&end
eof
fi #if boundary_layer=gfdl

cat > input_ice.nml << eof
&coupling_nml
  init_date=${iniyear}${inimonth}${iniday}
  caltype=$caltype
```

```
 jobnum=$jobnum
 inidate=$date
 runtime0=$truntime0
 runtime=$runtime
 dt_cpl_ai=$dt_cpl_ai
 dt_cpl_io=$dt_cpl_io
 dt_cice=$dt_ice
 pop_icediag=$POP_ICEDIAG
 ice_pressure_on=.true.
 ice_fwflux=.true.
 use_ocnslope=.false.
 use_umask=.false.
 rotate_winds=.true.
 limit_icemelt=.false.
 meltlimit=-200.0
 use_core_runoff=.true.
 precip_factor=1.0
 cst_ocn_albedo=.true.
 ocn_albedo=0.1
 gfdl_surface_flux=$GFDL_FLUXES
 chk_gfdl_roughness=.false.
 chk_frzmlt_sst=.false.
 chk_i2o_fields=.false.
 chk_o2i_fields=.false.
 chk_i2a_fields=.false.
 chk_a2i_fields=.false.
&end
eof

# 3.2.4 namelists for mom4p1

# a. standalone mode input namelist file
cp -f $inputdir/mom4/mom4_in.nml mom4_in.nml

alap=1.0e5
truncate_velocity='.true. '
truncate_verbose='.true. '
if [[ $year -gt $iniyear ]]; then
    truncate_velocity='.false. '
    truncate_verbose='.false. '
fi

#temp_restore_tscale=30.0 #sst restoring time scale of 30 days
temp_restore_tscale=-1.0 #NO SST restoration!
#salt_restore_tscale=60.0 #sss restoring time scale of 60 days
salt_restore_tscale=15.0 #strong SSS relaxation as 'recommended'
#salt_restore_tscale=-1 #NO SSS restoration!
use_waterflux='.true. '
layout=$oce_nx,$oce_ny #mpi partitioning pattern
Simple_frazil='.false. '  #simple temp frazil. if '.f. ' use complicated scheme
                          #                       and allow multi-layer frazil.
Accurate_frazil='.true. '  #accurate temp frazil. must be .t. if Simple_frazil=.f.
                          #       vice versa.
```

```
TL_frazil='.false. ' #top layer frazil. if '.f. ' multi-layer frazil

diff_cbt_iw=0.1e-6          #'background diffusion' when BL profile is NOT used.
                            # 1.e-4 m2/s is the default value
visc_cbu_iw=1.0e-4          #'BG' vertical vicocity
convection='.true. '
aredi=600.
agm=100.
ricr=0.3

ed mom4_in.nml <<eof
g/#NMONTH/s/#NMONTH/${nmonth}/
g/#NDAY/s/#NDAY/${nday}/
g/#SYEAR/s/#SYEAR/${iniyear}/
g/#SMON/s/#SMON/${inimonth}/
g/#SDAY/s/#SDAY/${iniday}/
g/#CAL_TYPE/s/#CAL_TYPE/${cal_type}/
g/#DT_CPL/s/#DT_CPL/${dt_cpl_io}/
g/#DT_OCE/s/#DT_OCE/${dt_oce}/
g/#LAYOUT/s/#LAYOUT/${layout}/
g/#VLIMIT/s/#VLIMIT/${truncate_velocity}/   g/#VWARN/s/#VWARN/${truncate_verbose}/
g/#SST_restoring/s/#SST_restoring/${temp_restore_tscale}/
g/#SSS_restoring/s/#SSS_restoring/${salt_restore_tscale}/
g/#Freezing_simple/s/#Freezing_simple/${Simple_frazil}/
g/#Freezing_accurate/s/#Freezing_accurate/${Accurate_frazil}/
g/#TL_frazil_only/s/#TL_frazil_only/${TL_frazil}/
g/#DIFF_CBT_IW/s/#DIFF_CBT_IW/${diff_cbt_iw}/
g/#VISC_CBU_IW/s/#VISC_CBU_IW/${visc_cbu_iw}/
g/#CONVECTION/s/#CONVECTION/${convection}/
g/#AREDI/s/#AREDI/${aredi}/
g/#AGM/s/#AGM/${agm}/
g/#RICR/s/#RICR/${ricr}/
g/#USE_waterflux/s/#USE_waterflux/${use_waterflux}/
w
q
eof

#'base_time' is read in from diag_table, and must NOT be changed during the exp.
if [ $jobnum = 1 ]; then
cp -f $inputdir/mom4/diag_table MOM4_input/diag_table
ed MOM4_input/diag_table <<eof
g/#SYEAR/s/#SYEAR/${year}/
g/#SMON/s/#SMON/${month}/
g/#SDAY/s/#SDAY/${day}/
w
q
eof
fi

# b. namelist for coupling purpose
#

 icemlt_factor=1.0 #if < 1, reduce the potential ice melt
```

```
#only usable when POP_icediag=.f.
frazil_factor=0.5 #mom4 uses two-level frog time stepping but cice
#uses forward time-stepping (see comments in code)
frazil_factor=1.0 #CH: MOM4 and CICE use same (two-timelevel) stepping!

cat > input_oce.nml << eof
&coupling
runtime=$runtime
dt_cpl=$dt_cpl_io
dt_mom4=$dt_oce
pop_icediag=$POP_ICEDIAG
do_ice_once=.false.
kmxice=5
fixmeltT=.false.
Tmelt=-.216
use_ioaice=.true.
aice_cutoff=0.15
chk_i2o_fields=.false.
chk_o2i_fields=.false.
icemlt_factor=$icemlt_factor
frazil_factor=$frazil_factor
iceform_adj_salt=.false.
sign_stflx=1.0
&end
eof




############################################################################ #
# 4. Launch/Execute the AusCOM Coupled Model on VAYU
# ############################################################################

set -e

echo "‘date‘ : ${jobnum} ${date} - starting mpirun/mpiexec" >> $jobdir/${expid}.log

echo
echo "*** mpirun/mpiexec started at: " ‘date‘ "***"
echo

#David Singleton’s solution to the bad perforamce of model:
export PATH=/opt/anumpirun/2.1.16a/bin:$PATH

#mpirun -n $nproc_cpl ./$oa3_exe :   -n $nproc_ice ./$ice_exe : \
# -n $nproc_atm ./$atm_exe :         -n $nproc_oce ./$oce_exe
mpirun --mca mpi_paffinity_alone 1  -n $nproc_cpl ./$oa3_exe : \
                                    -n $nproc_ice ./$ice_exe : \
                                    -n $nproc_atm ./$atm_exe : \
                                    -n $nproc_oce ./$oce_exe
echo
echo "*** job completed at: " ‘date‘ "***"
```

```
echo
echo "‘date‘ : ${jobnum} ${enddate} - done mpirun/mpiexec!" >> $jobdir/${expid}.log
echo ’Error code at end of simulation :’$?
 ########################################################################## #
# 5. Postprocessing: Saving the Output Data
# ##########################################################################

cd $rundir
rm MATM_input/*.$endyear.nc #remove the used forcing data

#
 ## 5.1 Output files of the coupler (OASIS3)
#

# Restart files
for resfile in ‘ls ?2?.nc‘; do
    mv $resfile ${restdir}/oasis3/${resfile}-${enddate}
done

#
## 5.2 Output files of the ocean (mom4)
#

# Restart files
cd $rundir/MOM4_restart/
for restfile in ‘ls ocean_*‘; do
    mv $restfile ${restdir}/mom4/${restfile}-${enddate}
done

# History files
cd $rundir/MOM4_hist
sdir=$rundir/MOM4_hist
tdir=${histdir}/mom4
tool=$bindir/do_mppncombine.ksh
archjob=‘qsub $tool -v bdir=$bindir,sdir=$sdir,tdir=$tdir,idate=${enddate}‘

#
## 5.4 Output files of the ice (cice)
#

cd $rundir

# Restart files
mv CICE_input/A2I_time1.nc      ${restdir}/cice/A2I_time1.nc-${enddate}
mv CICE_restart/ice.restart_file  ${restdir}/cice/ice.restart_file-${enddate}
mv CICE_restart/iced.*          ${restdir}/cice/.
if [ -f u_star.nc ]; then
    mv u_star.nc ${restdir}/cice/u_star.nc-${enddate}
fi

# History files (iceh_*.nc)
mv CICE_hist/* ${histdir}/cice/.
```

```
#
## 5.5 Store coupling fields output (if any), e.g. fields_i2o_in_ice.nc etc.
#
if [[ `ls fields*.nc | wc -w` -gt 0 ]]; then
for tmpfile in `ls fields*.nc`; do
    mv -f ${tmpfile} ${restdir}/oasis3/${tmpfile}_${enddate}
done
fi



############################################################################ #
# 6. Submission of the next job
# ############################################################################

cd ${jobdir}

#
# Number of the next job
#
(( nextjob = ${jobnum} + 1 ))

#
# update .date and .log file
#
if [ -f ${expid}.date ]; then
    mv ${expid}.date ${expid}.date_${jobnum}
fi
truntime0=`expr ${truntime0} + ${runtime}`
echo "${nextyear} ${nextmonth} ${nextday} ${nextjob} ${truntime0}" > ${expid}.date
echo "`date` : ${jobnum} ${enddate} - done post-processing!" >> ${expid}.log

#
# Check whether final date is reached. If not, keep going
#
if [[ $nextdate -gt $finaldate ]]; then
    echo "Experiment over"
    echo "`date` : Experiment over" >> ${expid}.log
else
    next_jobid=`qsub -W depend=after:${archjob} -p 10 run_auscom.VAYU`
    echo "`date` : New run is submitted."
fi

############################################################################ #
#
7. Epilogue
# ############################################################################

date exit
```

The Centre for Australian Weather and Climate Research is a partnership between CSIRO and the Bureau of Meteorology.