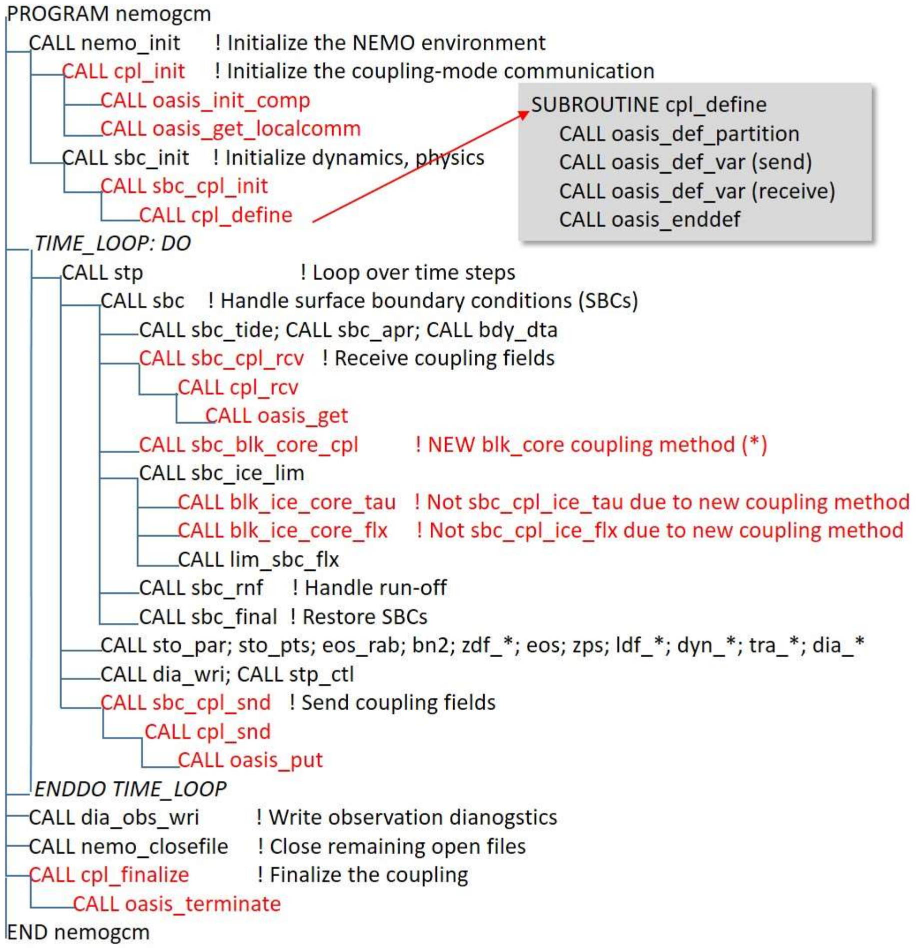
Supplement of Geosci. Model Dev., 17, 7815–7834, 2024
https://doi.org/10.5194/gmd-17-7815-2024-supplement

*Supplement of*

# Coupling the regional climate model ICON-CLM v2.6.6 to the Earth system model GCOAST-AHOI v2.0 using OASIS3-MCT v4.0

**Ha Thi Minh Ho-Hagemann et al.**

*Correspondence to:* Ha Thi Minh Ho-Hagemann (ha.hagemann@hereon.de)

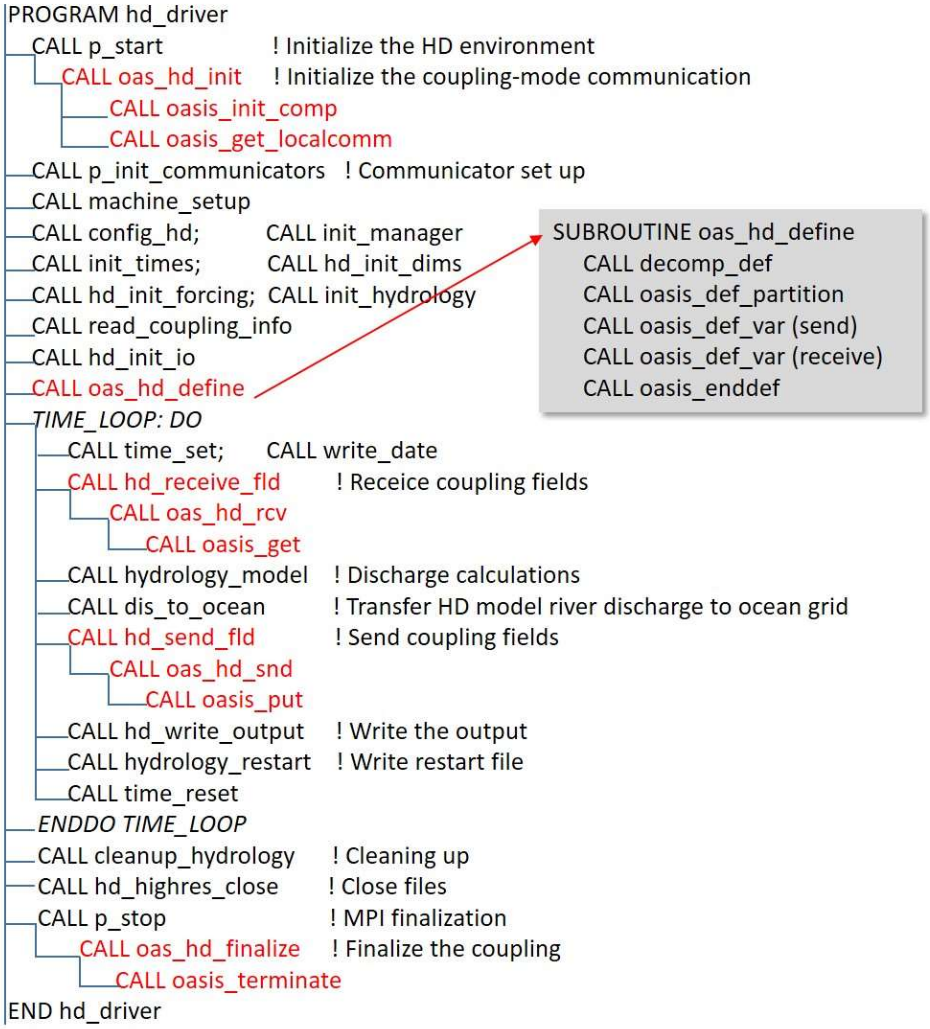


```
PROGRAM nemogcm
  CALL nemo_init      ! Initialize the NEMO environment
    CALL cpl_init     ! Initialize the coupling-mode communication
      CALL oasis_init_comp
      CALL oasis_get_localcomm
    CALL sbc_init    ! Initialize dynamics, physics
      CALL sbc_cpl_init
        CALL cpl_define

SUBROUTINE cpl_define
   CALL oasis_def_partition
   CALL oasis_def_var (send)
   CALL oasis_def_var (receive)
   CALL oasis_enddef

  TIME_LOOP: DO
    CALL stp                          ! Loop over time steps
      CALL sbc    ! Handle surface boundary conditions (SBCs)
        CALL sbc_tide; CALL sbc_apr; CALL bdy_dta
        CALL sbc_cpl_rcv   ! Receive coupling fields
          CALL cpl_rcv
            CALL oasis_get
        CALL sbc_blk_core_cpl         ! NEW blk_core coupling method (*)
        CALL sbc_ice_lim
          CALL blk_ice_core_tau   ! Not sbc_cpl_ice_tau due to new coupling method
          CALL blk_ice_core_flx   ! Not sbc_cpl_ice_flx due to new coupling method
          CALL lim_sbc_flx
        CALL sbc_rnf     ! Handle run-off
        CALL sbc_final  ! Restore SBCs
      CALL sto_par; sto_pts; eos_rab; bn2; zdf_*; eos; zps; ldf_*; dyn_*; tra_*; dia_*
      CALL dia_wri; CALL stp_ctl
      CALL sbc_cpl_snd   ! Send coupling fields
        CALL cpl_snd
          CALL oasis_put
  ENDDO TIME_LOOP
  CALL dia_obs_wri        ! Write observation dianogstics
  CALL nemo_closefile     ! Close remaining open files
  CALL cpl_finalize       ! Finalize the coupling
    CALL oasis_terminate
END nemogcm
```

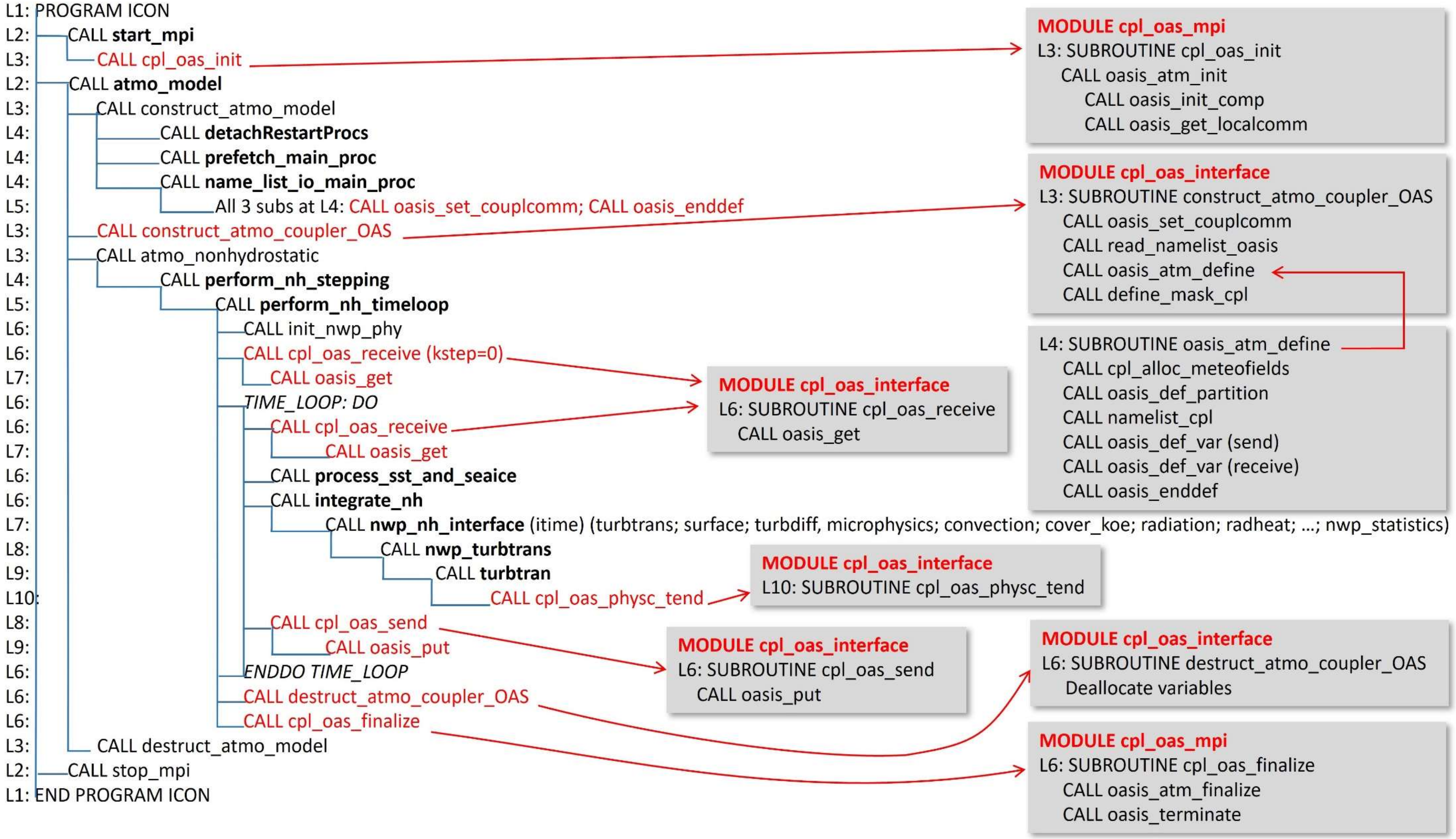
*S1: Flowchart of NEMO v3.6 with the OASIS3-MCT coupling interface (red text). In the default OASIS interface of NEMO v3.6, sensible and latent heat fluxes are passed from the atmospheric model. With the new coupling method (*), NEMO receives state variables (i.e. air temperature, humidity, etc.) to calculate the fluxes using the bulk formula (blk_core) which is available in NEMO v3.6 for the stand-alone mode.*



```
PROGRAM hd_driver
  CALL p_start                 ! Initialize the HD environment
    CALL oas_hd_init           ! Initialize the coupling-mode communication
      CALL oasis_init_comp
      CALL oasis_get_localcomm
  CALL p_init_communicators    ! Communicator set up
  CALL machine_setup
  CALL config_hd;          CALL init_manager
  CALL init_times;         CALL hd_init_dims
  CALL hd_init_forcing;    CALL init_hydrology
  CALL read_coupling_info
  CALL hd_init_io
  CALL oas_hd_define
  TIME_LOOP: DO
    CALL time_set;      CALL write_date
    CALL hd_receive_fld        ! Receice coupling fields
      CALL oas_hd_rcv
        CALL oasis_get
    CALL hydrology_model       ! Discharge calculations
    CALL dis_to_ocean          ! Transfer HD model river discharge to ocean grid
    CALL hd_send_fld           ! Send coupling fields
      CALL oas_hd_snd
        CALL oasis_put
    CALL hd_write_output       ! Write the output
    CALL hydrology_restart     ! Write restart file
    CALL time_reset
  ENDDO TIME_LOOP
  CALL cleanup_hydrology       ! Cleaning up
  CALL hd_highres_close        ! Close files
  CALL p_stop                  ! MPI finalization
    CALL oas_hd_finalize       ! Finalize the coupling
      CALL oasis_terminate
END hd_driver
```

```
SUBROUTINE oas_hd_define
    CALL decomp_def
    CALL oasis_def_partition
    CALL oasis_def_var (send)
    CALL oasis_def_var (receive)
    CALL oasis_enddef
```

*S2: Flowchart of HD v5.1 with the OASIS3-MCT coupling interface (red text).*



3

*S3: Flowchart of ICON-NWP/ICON-CLM with the OASIS3-MCT coupling interface. Red text shows the OASIS interface. Bold text displays the modified subroutines of ICON due to the coupling. "L1" indicates Level 1 – the main ICON program, etc.*

*Table S1: New OASIS coupling files added to ICON-NWP/ICON-CLM.*

| File | Directory | Fortran File | Modules/Subroutines (Sub.) | Description |
|---|---|---|---|---|
| 1 | src/atm_phy_nwp/ | cpl_oas_vardef.f90 | Module cpl_oas_vardef | Coupling variables definition |
| 2 | | cpl_oas_mpi.f90 | Module cpl_oas_mpi | Initialization/ Finalization for OASIS |
| | | | Sub. cpl_oas_init | OASIS initialize |
| | | | Sub. cpl_oas_finalize | OASIS finalize and terminate |
| 3 | | cpl_oas_interface.f90 | Module cpl_oas_interface | Main subroutines of OMCI |
| | | | Sub. construct_atmo_coupler_OAS | Definition: OASIS decomposition, reading coupling namelist, definite coupling mask |
| | | | Sub. cpl_oas_receive | Data exchange: Receive fields from OASIS |
| | | | Sub. cpl_oas_send | Data exchange: Send fields to OASIS |
| | | | Sub. cpl_oas_physc_tend | Update momentum, latent and sensible heat fluxes |
| | | | Sub. destruct_atmo_coupler_OAS | Deallocate coupling variables |

*Table S2: Modified ICON's files due to the OASIS3-MCT coupling interface.*

| File | Directory | Fortran File | Modules/Subroutines (Sub.) | Description |
|---|---|---|---|---|
| 1 | src/parallel_infrastructure/ | mo_mpi.f90 | Module mo_mpi<br>Sub. start_mpi | Initialization of ICON |
| 2 | src/drivers/ | mo_atmo_model.f90 | Module mo_atmo_model<br>Sub. atmo_model | Atmospheric model ICON |
| 3 | src/io/restart/ | mo_restart.f90 | Module mo_restart | |
| | | | Sub. detachRestartProcs | Detach the restart processes |
| 4 | src/io/atmo/ | mo_async_latbc.f90 | Module mo_async_latbc | |
| | | | Sub. prefetch_main_proc | Initialize the prefetch processor |
| 5 | src/io/shared/ | mo_name_list_output.f90 | Module mo_name_list_output | |
| | | | Sub. name_list_io_main_proc | Initialize name list output |
| 6 | src/lnd_phy_nwp/ | mo_nwp_lnd_types.f90 | Module mo_nwp_lnd_types | Declare new variable sea ice albedo “alb_si_ext” to receive from NEMO |
| 7 | src/lnd_phy_nwp/ | mo_nwp_lnd_state.f90 | Module mo_nwp_lnd_state<br>Sub. new_nwp_lnd_diag_list | Declare new variable sea ice albedo “alb_si_ext” to write to output |
| 8 | src/lnd_phy_nwp/ | mo_nwp_sfc_utils.f90 | Module mo_nwp_sfc_utils | Initializes and update surface variables |
| | | | Sub. nwp_surface_init | Initializes sea ice albedo by “alb_si_ext” from NEMO |
| | | | Sub. process_sst_and_seaice | Update sea ice albedo by “alb_si_ext” from NEMO |
| 9 | src/atm_dyn_iconam/ | mo_nh_stepping.f90 | Module mo_nh_stepping | Initializes and controls the time stepping in the nonhydrostatic model |
| | | | Sub. perform_nh_timeloop | Time looping of the nonhydrostatic model |
| 10 | src/atm_phy_nwp/ | mo_nwp_turbtrans_interface.f90 | Module mo_nwp_turbtrans_interface<br>Sub. nwp_turbtrans | Interface between nwp_nh_interface to the turbulence parameterisations |
| 11 | src/atm_phy_schemes/ | turb_transfer.f90 | Module turb_transfer<br>Sub. turbtran | Computing the coefficients for turbulent transfer |

*Table S3: Number of requested nodes/processors for performance tests of GCOAST-AHOI on Levante. NPX and NPY are the processors for NEMO corresponding to x and y dimensions, respectively.*

| Case | Nodes | Total processors | Processors for ICON | Processors for NEMO | Processors for HD |
|---|---|---|---|---|---|
| **A** | 25 | 3200 | 1599 | NPX x NPY = 40 x 40 = 1600 | 1 |
| **B** | 30 | 3840 | 2239 | NPX x NPY = 40 x 40 = 1600 | 1 |
| **C** | 30 | 3840 | 1839 | NPX x NPY = 50 x 40 = 2000 | 1 |
| **D** | 40 | 5120 | 3519 | NPX x NPY = 40 x 40 = 1600 | 1 |
| **E** | 40 | 5120 | 2719 | NPX x NPY = 60 x 40 = 2400 | 1 |

*S4: Interface structure of OMCI*

We can divide the OMCI into four main processes: Initialization, Definition, Data exchange, and Finalization. Box B1 (Fig. 2) belongs to the Initialization phase of OASIS in ICON. In this phase, the ICON file mo_mpi.f90 is modified, and the file cpl_oas_mpi.f90 of OMCI is newly created (see Supplementary Tables S1 and S2). In mo_mpi.f90, the start_mpi subroutine from ICON calls the cpl_oas_init subroutine from OMCI, which in turn calls two subroutines from the OASIS library (oasis_init_comp and oasis_get_localcomm). The subroutine cpl_oas_init (belonging to cpl_oas_mpi.f90) is similar to the subroutine oas_cos_init of the unified OASIS interface in CCLM (Will et al., 2017).

Boxes B2 and B3 belong to the Definition phase to define and allocate all coupling fields. In this phase, three ICON files are slightly modified by calling oasis_set_couplcomm and oasis_enddef from the OASIS library (rows 3-5 in Table S2). In addition, some code lines are added to three ICON modules to declare the new sea ice albedo variable “alb_si_ext” to be sent to NEMO (row 6 Table S2).

Two additional files from OMCI (cpl_oas_vardef.f90 and cpl_oas_interface.f90) are added to the ICON source code. Module cpl_oas_vardef simply contains a definition of all coupling variables. Part of the cpl_oas_interface module is the construct_atmo_coupler_OAS subroutine which is called by the ICON atmo_model subroutine (src/drivers/mo_atmo_model.f90). The subroutine construct_atmo_coupler_OAS also calls oasis_set_couplcomm before calling three other subroutines of OMCI (i.e. read_namelist_oasis, oasis_atm_define and define_mask_cpl) to define the decomposition of ICON and to read in the ocean domain masked on the atmospheric domain of ICON (i.e. the coupling mask, variable mask_cpl) from a netcdf file named atmin.nc.

Calling oasis_set_couplcomm in the Definition phase is a peculiarity of ICON compared to CCLM, NEMO and HD. The reason for this is that ICON devotes one processor out of the total number of processors to reading the lateral boundary conditions (by setting num_prefetch_proc=1 in ICON’s parallel_nml namelist). This single processor should be seen by OASIS, but only in the Initialization phase. The OASIS subroutine oasis_set_couplcomm, called after the Initialization, helps to set a coupling communicator in the case that only a subset of the component processes is involved in the coupling. In this case, the “subset” is all the processors allocated for ICON except the one defined by prefetch_proc. In the ICON-CLM versions prior to 2.6.4, it is possible to set num_prefetch_proc=0, so that the call to oasis_set_couplcomm in the Definition phase would not be necessary. However, since version 2.6.4, num_prefetch_proc=1 is mandatory. Therefore, oasis_set_couplcomm must be called, otherwise the coupled model will hang after the Initialization.

The exchanged variables (see Fig. 1) are listed in the OMCI subroutine oasis_atm_define which are read in from a namelist file namelist_cpl_atm_oce to define which variables are sent and received. The variable names used in ICON, corresponding to the exchanged variables, are similar to the variables listed in Table 1 of Bauer et al. (2021).

Boxes B4, B5, and B7 belong to the Data exchange phase while the coupled system is running. In this phase, subroutines in the OMCI module cpl_oas_interface are used, and five ICON modules are modified (rows 7-9 Table S2). The five ICON modules are highlighted in red in Fig. 2 from level 4 to 9, under the subroutine perform_nh_stepping. Variables (i.e. sea surface temperature, sea ice fraction and albedo) received from NEMO via OMCI by calling the subroutine cpl_oas_receive are updated to the newer values at each ICON time step within the subroutine perform_nh_timeloop through several steps. They are first updated in the subroutine process_sst_and_seaice, and then used to modify the surface roughness in the turbulent scheme via the subroutine turbtran (turb_transfer.f90) of ICON. The subroutine turbtran is called by the module nwp_turbtrans, which in turn is called by the subroutine nwp_nh_interface inside the subroutine integrate_nh. After the subroutine integrate_nh, the subroutine cpl_oas_send is called to pass the defined exchange variables from ICON to NEMO and HD via OMCI.

Box B6 in Fig. 2 indicates the Finalization phase for OASIS. Here, two subroutines destruct_atmo_coupler_OAS and cpl_oas_finalize are called. The subroutine destruct_atmo_coupler_OAS simply deallocates all coupling variables. OMCI's subroutine cpl_oas_finalize calls two OASIS subroutines oasis_atm_finalize and oasis_terminate, as in the Finalization phase in CCLM, NEMO and HD. Alternatively, the Finalization box can be placed at level 3, before destruct_atmo_model of ICON. However, leaving the Finalization box at level 6 is more flexible, e.g. for testing the behavior of ICON when finalizing OASIS at the ktstep=nsteps_total or ktstep=nsteps_total-1.

*S5: Compile ICON with OMCI on Levante.*

a. Environment settings:

```
NETCDFF_DIR=/sw/spack-levante/netcdf-fortran-4.5.3-k6xq5g
NETCDFC_DIR=/sw/spack-levante/netcdf-c-4.8.1-2k3cmu
ECCODES_ROOT=/sw/spack-levante/eccodes-2.21.0-3ehkbb
HDF5_DIR=/sw/spack-levante/hdf5-1.12.1-tvymb5
SZIP_ROOT=/sw/spack-levante/libaec-1.0.5-gij7yv
MKL_ROOT=/sw/spack-levante/intel-oneapi-mkl-2022.0.1-ttdktf/mkl/2022.0.1
MPIINC=/sw/spack-levante/openmpi-4.1.2-yfwe6t/include
MPILIB=/sw/spack-levante/openmpi-4.1.2-yfwe6t/lib
MODULES="'intel-oneapi-compilers/2022.0.1-gcc-11.2.0 openmpi/4.1.2-intel-2021.5.0'"
GCCLIB="/sw/spack-levante/gcc-11.2.0-7jcqrc/lib64"
PYTHON='/sw/spack-levante/mambaforge-4.11.0-0-Linux-x86_64-sobz6z/bin/python3'
```

b. Compiling:

The environment must be the same for the coupler OASIS3-MCT v4.0 as well as for the three model components ICON, NEMO and HD. OASIS3-MCT v4.0 is compiled first and will be used as the library to be linked to the three models.  To compile ICON with OMCI, one must adapt the configure file and icon/config/dkrz/levante.intel-2021.5.0_OASIS (see https://doi.org/10.5281/zenodo.10877618).
The command to compile ICON with OMCI using the setup levante.intel-2021.5.0_OASIS is:

```
icon/config/dkrz/levante.intel-2021.5.0_OASIS --disable-coupling --disable-ocean --disable-jsbach --enable-coupling_OAS --disable-art --enable-ecrad
```

Note that "--disable-coupling --disable-ocean --disable-jsbach" is not to couple with YAC, ICON-O and JSBACH, respectively. Meanwhile "--enable-coupling_OAS --enable-ecrad" is to switch on OMCI and to run ICON with the radiation scheme ecRad. Consequently, a binary file icon is located under the directory icon/bin, like in the case without OASIS. ICON with OMCI has also successfully been compiled on other machines of the same architecture as Levante and on NEC-Aurora at DWD.

*S6: Prepare OASIS input files.*

a. Grid and mask files

Using CDO and NCO libraries is a convenient manner to produce information about grids and masks used by OASIS (i.e. grids.nc and masks.nc), as well as the remapping files requested by OASIS before running the coupled system. The file grids.nc should contain longitude (Lon) and latitude (Lat) of the ICON, NEMO and HD grids. Although three models are considered, there are five grids which are named icon, nemo, nico, nmhd, and hdmd. Lon and Lat of icon and nico have the same dimension of (1, 231660). Lon and Lat of nemo and nmhd have the same dimension of (902, 777). Lon and Lat of hdmd have the dimension of (960, 540). The reason to create five grids is that the masks of them are different. OASIS will do the interpolation/exchange on points which have the mask value of zero and ignore the points with mask of one. File masks.nc contains five masks i.e. icon.msk, nemo.msk, nico.msk, nmhd.msk and hdmd.msk as following:

- The masks icon.msk and hdmd.msk are both zero. They are used for the source grids (see namcouple in S6 below); therefore, OASIS should send results from all points to other grids.
- The nemo.msk has values of zero on the ocean grid points and values of one on the land points. nemo is also a source grid, but results are only available on ocean points.
- The nico.msk has zero values only in the area overlapped between the NEMO domain and the ICON domain, i.e. the dark blue area in Figure 3. The other grid points have a value of one, thus, sea surface

temperature or sea ice fraction from NEMO/LIM3 is updated in ICON only over grid points inside of the dark blue area, also known as the coupling domain.

- The nmhd.msk has values of one everywhere, only on river mouth points the values are zero.

### b. Remapping files

Remapping files are netcdf files containing interpolation matrix, based on that OASIS can exchange data between different model grids. The remapping files can be either generated by OASIS or prepared manually. Applying the first method, OASIS does the interpolation using the *SCRIPR* function as described in the *namcouple* file, GROUP 2 (see S6). Options for the *SCRIPR* function can be DISTWGT, GAUSWGT, BILINEAR or CONSERV. With this method, the grids.nc and masks.nc files will be taken into account, and a remapping file (e.g. **rmp_icon_to_hdmd_DISTWGT.nc**) will be generated. One can conduct one month simulation with the coupled model and wait until the remapping file is generated, which would take about 10-20 minutes. Then one can stop the simulation and rerun the coupled model using the saved remapping file and the *MAPPING* function, as shown in GROUP 1 or GROUP 3 of S6.

Method 2 is to prepare the remapping files using CDO functions outside and before running the coupled model. First, we extracted Lon and Lat information of grids nemo, nico and hdmd from the above-mentioned grids.nc file to obtain the nemogrid.nc, nicogrid.nc, hdmhgrid.nc, respectively. Specifically, for the icogrid.nc, the Lon and Lat of grid icon in grids.nc must be converted to a 1-dimension field of length "ncells" (similar to clon (231660) and clat (231660)), adding vertices information from the ICON grid. We use these netcdf files in the script *remap_ICON_NEMO_HD.sh* to generate several remapping files (rmp_*CONSERV.nc and rmp_*DISTWGT.nc). This script uses "gencon" and "gendis" functions of CDO to produce the remapping files. Note that the HD grid has no corner lon-lat information, therefore only the "cdo gendis" can be used for remapping the ICON to HD grid, while "cdo gencon" is applied for the other two cases. The remapping files are used in the file namcouple as shown in GROUP 1 and 3 of S6 below.

```
script remap_ICON_NEMO_HD.sh:
rm -f rmp_icon_to_nemo_*.nc
rm -f rmp_nemo_to_nico_*.nc
rm -f rmp_icon_to_hdmd_*.nc
CDO gencon,nemogrid.nc icogrid.nc rmp_icon_to_nemo_CONSERV.nc
CDO gencon,nicogrid.nc nemogrid.nc  rmp_nemo_to_nico_CONSERV.nc
CDO gendis,hdmdgrid.nc icogrid.nc rmp_icon_to_hdmd_DISTWGT.nc
```

### c. File namcouple

One field of each exchange group (i.e. atmosphere → ocean; atmosphere → river run-off; ocean → atmosphere; river-runoff → ocean) in the file namcouple is given as an example in S6. In total, 19 fields are exchanged between the three models via OMCI. For all exchanges where ICON is taking part, the *Send_var* and *Receive_var* in the file namcouple must be the same to what is defined in the OMCI/oasis_atm_define as well as in the namelist *&cpl_nml* in the namelist_cpl_atm_oce file (see example in S7). Coupling time step is 3600 seconds. LAG=+0 is set in the GROUP 1 meaning NEMO receives output of ICON at every hour, without any delay. LAG=+100 in GROUP 2 means that HD receives run-off from ICON at every hour plus one running time step (i.e. 100 seconds) of ICON. For any field which is exchanged with a LAG larger than 0, a restart file (i.e. atmin.nc, sstoc.nc or rivin.nc) is needed by OASIS. However, one must prepare the file only once at the first simulation month. These restart files are generated and overwritten by OASIS at the end of each run. One should, therefore, save the restart files right after any run for each month so that they are available in case a later re-running of the simulation is desired for a specific month.

*S7: Example of namcouple file.*

```
################################################################################
 $NFIELDS
         19
 $END
################################################################################
 $STRINGS
################################################################################
#        GROUP 1:        ATMOSPHERE  --->>>  OCEAN
# Field 8: U wind component at 10M [m/s]
# Send_var   Receive_var   Var_number   Coupling_interval(s)  Transformations   Restart_file   Field_Status
U10MtNB       O_WNDI         8                  3600                    2            atmin.nc       EXPORTED
231660    1     902     777     icon      nemo     LAG=+0
R  0  R  0
LOCTRANS      MAPPING
 INSTANT
 rmp_icon_to_nemo_CONSERV.nc src
################################################################################
#         GROUP 2:      ATMOSPHERE  --->>>  RIVER RUN-OFF
# Field 22:  Surface run-off [kg/m2], sum over forecast of 1hr, converted to m/s
# Send_var   Receive_var   Var_number   Coupling_interval(s)  Transformations   Restart_file   Field_Status
RO_StNB       RUNOFF_S       22                 3600                    2            atmin.nc       EXPORTED
231660      1      960     540     icon      hdmd     LAG=+100
R  0  R  0
LOCTRANS       SCRIPR
 INSTANT
 DISTWGT    LR    SCALAR    LATLON    10    4
################################################################################
#         GROUP 3:       OCEAN  --->>>  ATMOSPHERE
# Field 1:  Sea surface temperature [K]
# Send_var   Receive_var   Var_number   Coupling_interval(s)  Transformations   Restart_file   Field_Status
O_TepMix       SSTfNB           1                  3600                    2           sstoc.nc       EXPORTED
902      777      231660     1         nemo    nico       LAG=+90
R  0  R  0
LOCTRANS        MAPPING
 AVERAGE
 rmp_nemo_to_nico_CONSERV.nc src
################################################################################
#        GROUP 4:      RIVER RUN-OFF  --->>>  OCEAN
# Field 19:  River discharge [m3/s]: already on NEMO's grid
# Send_var   Receive_var   Var_number   Coupling_interval(s)  Transformations   Restart_file   Field_Status
RDC2NEMO    O_Runoff           19                 3600                    1           rivin.nc       EXPORTED
nmhd     nmhd       LAG=+3600
R  0  R  0
LOCTRANS
 AVERAGE
END
```

*S8: Example of namelist_cpl_atm_oce*

```
&cpl_nml
!--------------------------------------------------------
! ATMOSPHERE send              !   OCEAN receive
  atm_snd_u10 = 'U10MtNB'     !, 'O_WNDI'
  atm_snd_u10 = 'V10MtNB'     !, 'O_WNDJ'
  atm_snd_swd = 'SWDNtNB'     !, 'O_SWDN'
  atm_snd_lwd = 'LWDNtNB'     !, 'O_LWDN'
  …
!--------------------------------------------------------
! ATMOSPHERE send              !   RIVER receive
  atm_snd_ros = 'RO_StNB'     !, 'RUNOFF_S'
  atm_snd_rog = 'RO_GtNB'     !, 'RUNOFF_G'
!--------------------------------------------------------
! ATMOSPHERE receive           !   OCEAN send
  atm_rcv_sst = 'SSTfNB'       !, 'O_TepMix'
  atm_rcv_ifr = 'FRIfNB'       !, 'OIceFrc'
  atm_rcv_ial = 'ALBIfNB'      !, 'O_AlbIce'
  …
/
```

*S9: Running GCOAST-AHOI.*

The command to conduct the experiment using the job scheduling system SLURM installed on Levante is:

```
    srun -l   --hint=nomultithread   --distribution=block:cyclic   --multi-prog mpmd.lst
```

in which mpmd.lst is a text file listing the number of processors given to each model component. For example, if 25 nodes are used to run the coupled model on Levante, with each node comprising 128 processors, the number of processors given to ICON, NEMO and HD can be 1599, 1600 and 1 processor, respectively. The mpmd.lst file would look like this:

```
    $cat mpmd.lst
    0-1598 icon
    1599-3198 oceanx
    3199-3199 hdmd.x
```

*S10: Using LUCIA to estimate model computing performance.*

To use LUCIA, first step is to compile the LUCIA source code included the OASIS3-MCT released package by

```
    cd ${OASIS_DIR}/util/lucia
    lucia -c
```

to obtain the executable file lucia.exe. ${OASIS_DIR} is the path referring to the OASIS3-MCT directory. Then, in the namcouple file, under the section $NLOGPRT we set:

```
     $NLOGPRT
        1 -1
```

and then the coupled model for one month as normal. Consequently, in the working directory, some files with name of lucia.xx.xxxxxx will be generated. In this working directory, we have to run two commands:

```
     ${OASIS_DIR}/util/lucia/lucia                       # generate oasis_balance.eps
     ps2pdf  oasis_balance.eps  oasis_balance.pdf        # convert to pdf file oasis_balance.pdf
```

to obtain oasis_balance.eps and then oasis_balance.pdf. File oasis_balance.pdf includes a bar-chart showing the calculation time and the coupling exchange duration including time spent to wait for the other model components.

In this study, we conduct five one-month experiments using ICPL266 to find out the most suitable number of nodes used for each model component. The five experiments are carried out with different numbers of nodes (i.e. 25 nodes, 30 nodes and 40 nodes). The number of processors assigned to each model component is listed in Table 3.

Figure S1 shows computation time (green bars) and coupling exchange time, including the time spent while waiting for slower models (red bars) of the model components. In principle, the smaller the red bars, the better the computational performance. Also, the red bars of ICON and NEMO should not be too different. As a simpler model, HD runs on a single processor, so its running time (green bar) is the shortest and the waiting time (red bar) the longest of the three models.

Figure S1 shows that Case C is the most balanced of the five experiments. In this case, 30 nodes were used, the number of processors (in short procs) given to NEMO (2000 procs) and ICON (1839 procs) are similar. The green and red bars of ICON and NEMO are similar. In Case A, 25 nodes were used, the number of processors given to NEMO (1600 procs) is also very similar to that given to ICON (1599 procs), but the green bars of this case are the highest of the five cases. These two cases have a different ratio of processors used for NEMO. The best one (Case C) has NPX x NPY = 50 x 40 while the worst one (Case A) has NPX x NPY = 40 x 40. Case B also uses 30 nodes like Case C, but with NPX x NPY = 40 x 40 = 1600 and ICON uses 2239 processors. With more processors, ICON runs faster than NEMO in this case, so there is no balance. In Case D uses 40 nodes, again 1600 processors for NEMO and an increased number of processors (3519 procs) for ICON. Here, ICON runs as slow as NEMO, even though it uses more than twice as many processors as NEMO. Case E also uses 40 nodes, but the number of processors for ICON and NEMO are not much different (i.e. 2719 and 2400 procs). However, NEMO runs faster than ICON and the system takes a longer time to run than in Case C. ICON with more processors in Case D and Case E is slower than on Case B and Case C with less processors, which indicates that too many processors were used. The common recommendation for ICON is to have at least 100 grid cells per processor, which would be about 2000 processors at maximum for the EURO-CORDEX domain. These results indicate that not only the number of the nodes used, but also the ratio of processors between ICON and NEMO, and the ratios of NPX and NPY for NEMO should be chosen carefully. The optimal setup may be different on other computer systems. A more thorough analysis is planned to be done with the new OASIS-MCT_5.0 version of LUCIA.

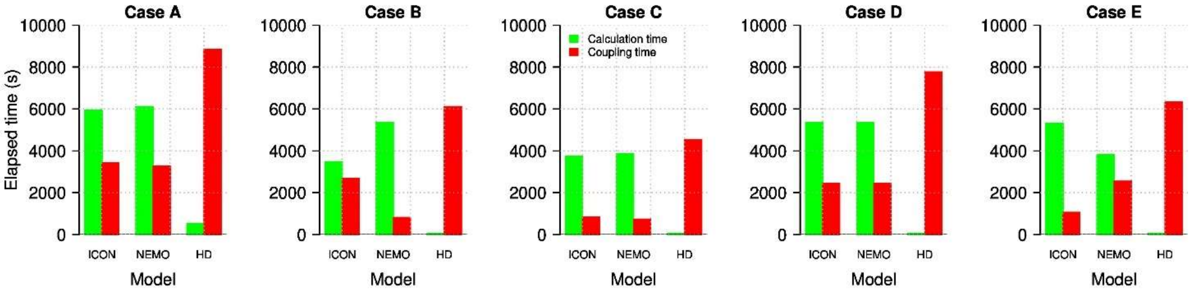
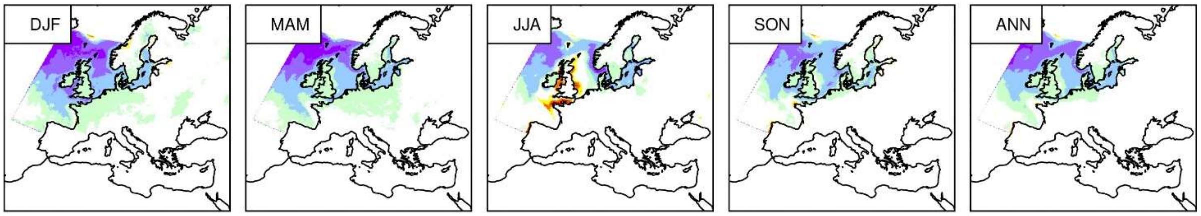


*Figure S1: Calculation time (green) versus coupling exchange duration including time spent to wait for other model components (red). See table S3 for a detailed view of the node balance of the displayed cases.*

a) ICPL266 - ICON266, T_S



b) ICPL266 - ICON266, T_2M

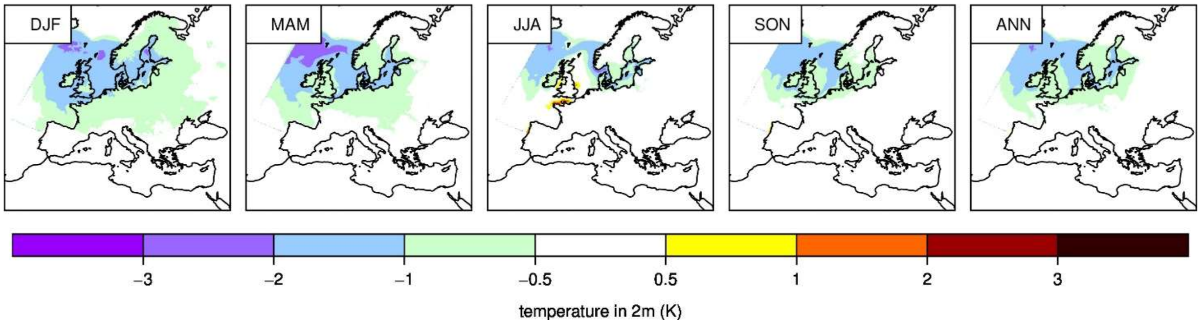
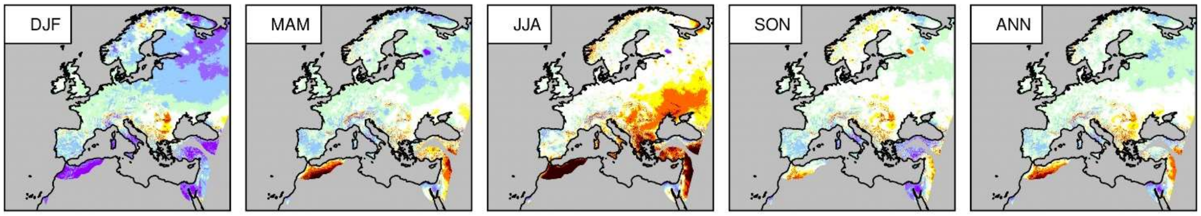


*Figure S2: Seasonal (DJF, MAM, JJA, SON) and annual (ANN) T_S and T_2M (K) difference between ICPL266 compared to ICON266 for the period of 2010-2018.*

a) ICON266



b) ICPL266

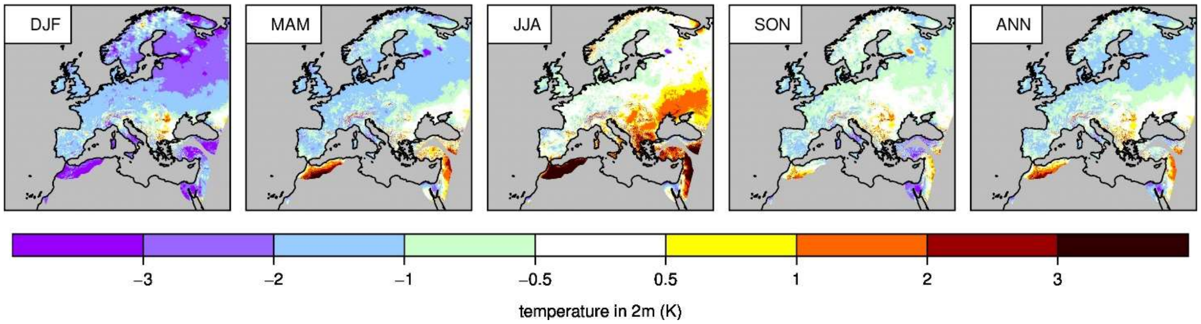


*Figure S3: Seasonal (DJF, MAM, JJA, SON) and annual (ANN) bias of T_2M (K) of a) ICON266 and b) ICPL266 compared to the E-OBS data for the period of 2010-2018.*

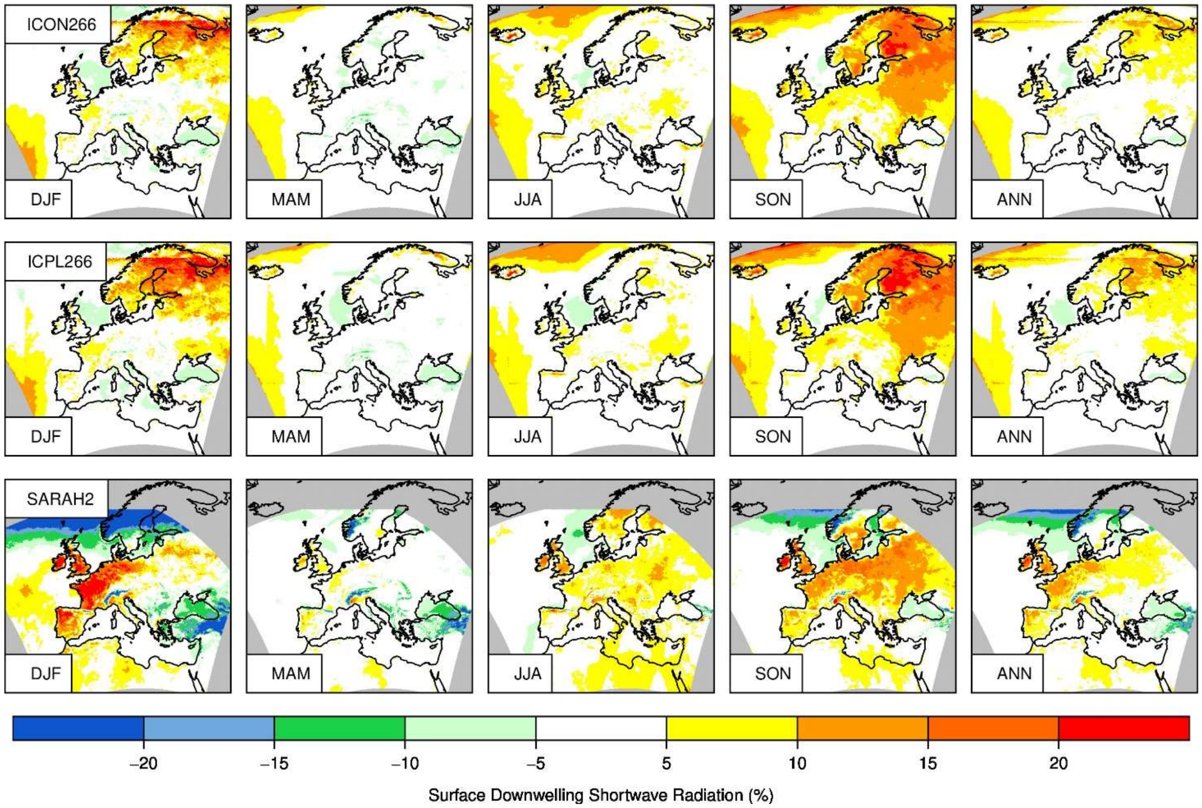


*Figure S4: Seasonal (DJF, MAM, JJA, SON) and annual (ANN) of shortwave downward radiation difference (%) for ICON266 (top), ICPL266 (middle) and the SARAH2 data (bottom) compared to the ERA5 data for the period of 2010-2018.*

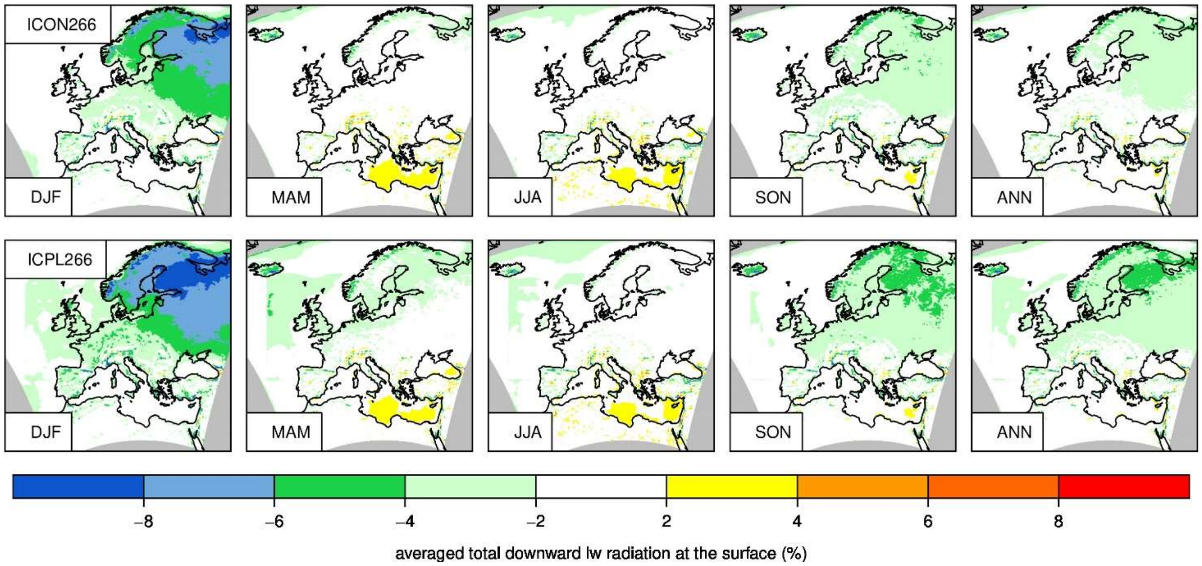
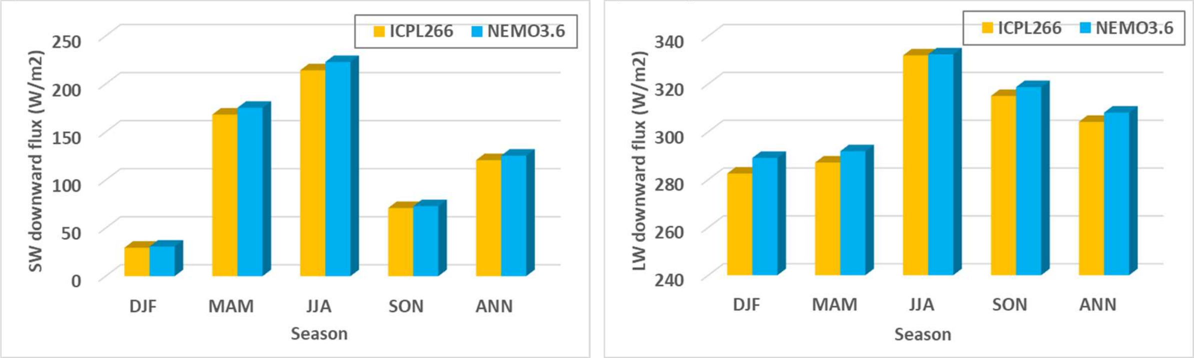
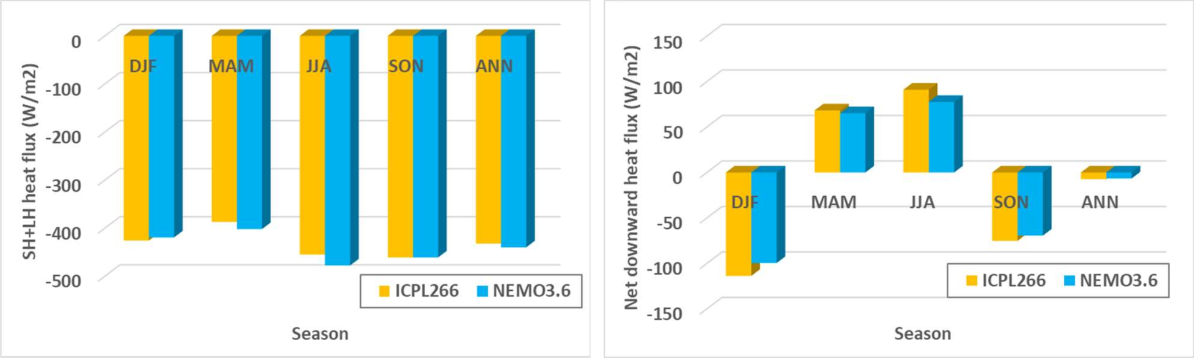


*Figure S5: Seasonal (DJF, MAM, JJA, SON) and annual (ANN) of longwave downward radiation difference (%) for ICON266 (top), and ICPL266 (bottom) compared to the ERA5 data for the period of 2010-2018.*

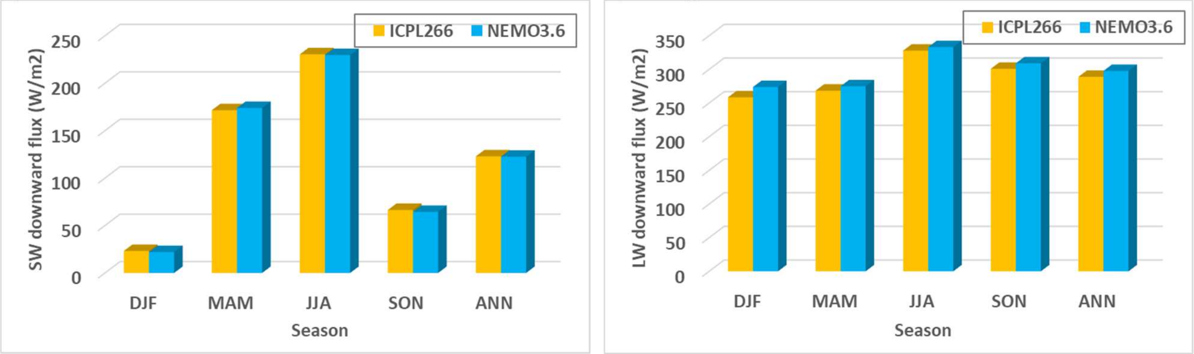

a) Shortwave downward radiation

b) Longwave downward radiation

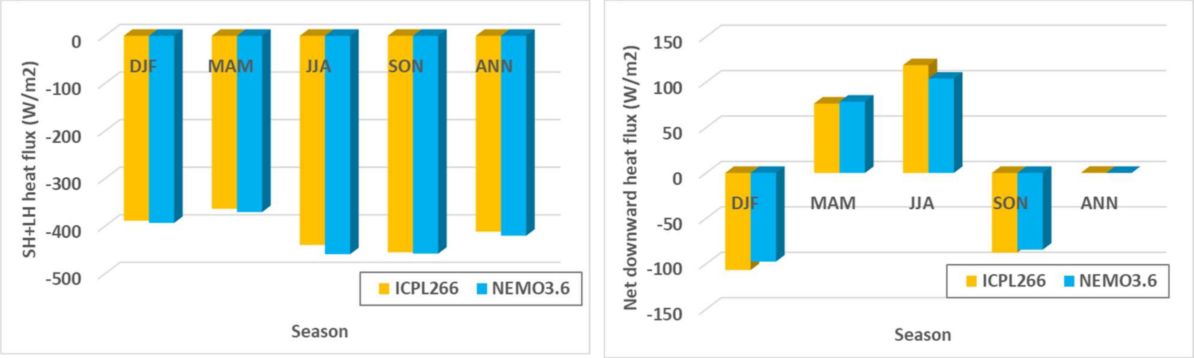

c) Sensible + Latent heat flux

d) Net downward heat flux

*Figure S6: Seasonal flux (W/m2, positive downward) of a) shortwave downward radiation, b) longwave downward radiation, d) sum of sensible and latent heat flux, and d) net downward heat flux of ICPL266 and NEMO3.6 averaged over the North Sea for the period of 2010-2018.*


a) Shortwave downward radiation

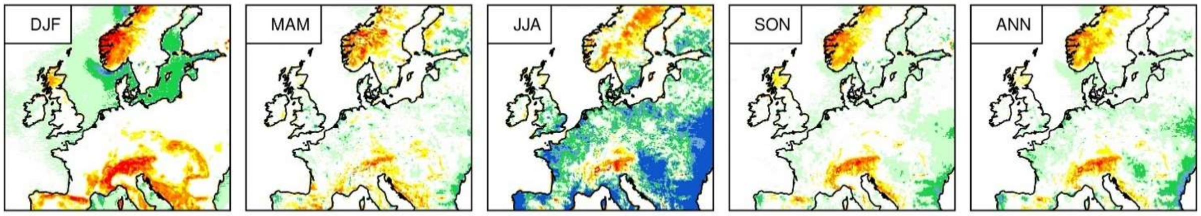
b) Longwave downward radiation


c) Sensible + Latent heat flux

d) Net downward heat flux

*Figure S7: Similar to Figure S6 but for the Baltic Sea.*

a) ICON266



b) ICPL266

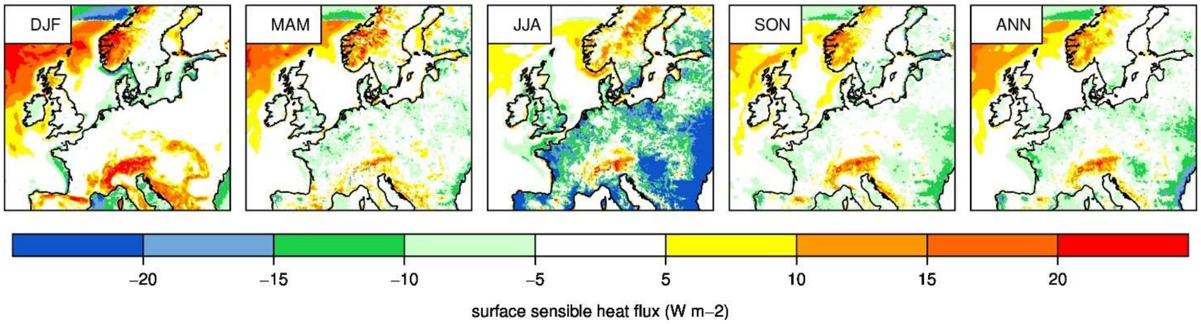
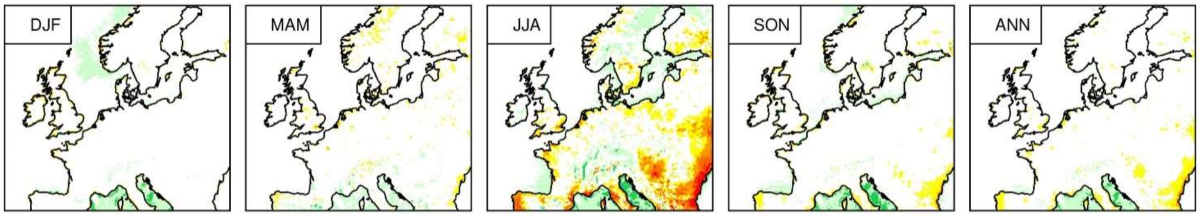


*Figure S8: Seasonal (DJF, MAM, JJA, SON) and annual (ANN) mean of sensible heat flux (W/m$^2$, positive downward) difference for a) ICON266 and b) ICPL266 compared to the ERA5 data for the period of 2010-2018.*

a) ICON266



b) ICPL266

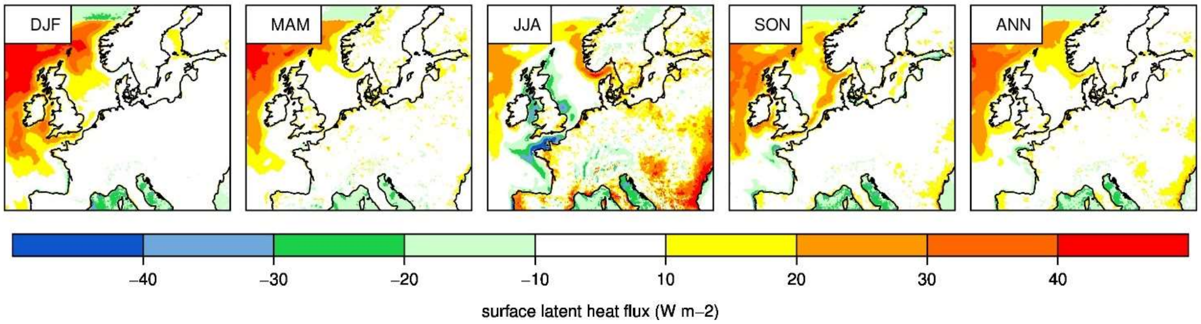
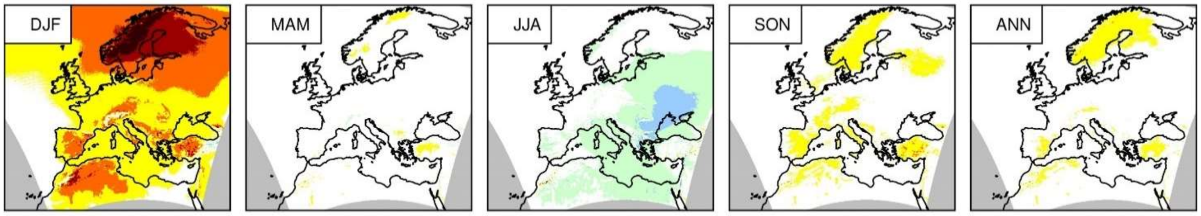


*Figure S9: Seasonal (DJF, MAM, JJA, SON) and annual (ANN) mean of latent heat flux (W/m$^2$, positive downward) difference for a) ICON266 and b) ICPL266 compared to the ERA5 data for the period of 2010-2018.*

a) ICON266



b) ICPL266

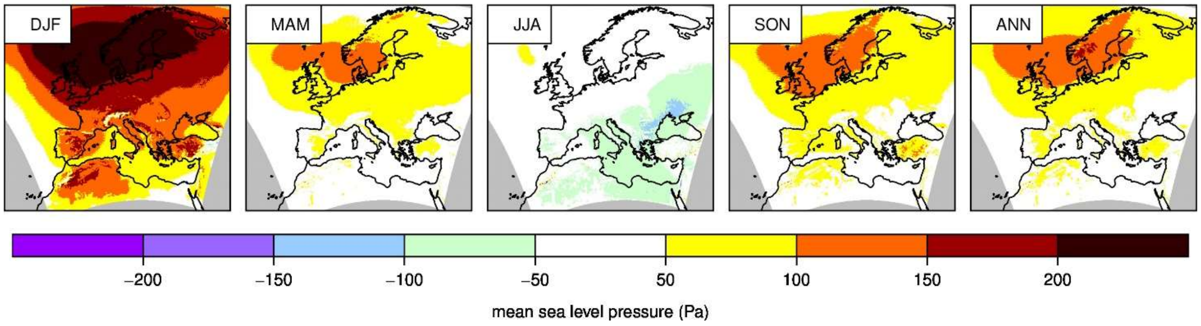
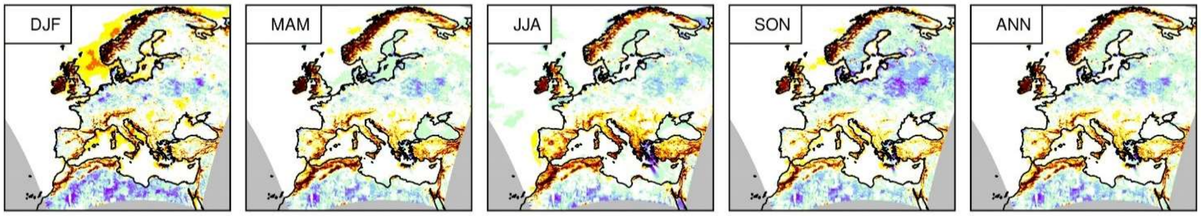


*Figure S10: Seasonal (DJF, MAM, JJA, SON) and annual (ANN) mean of mean sea level pressure (Pa) difference for a) ICON266 and b) ICPL266 compared to the ERA5 reanalysis data for the period of 2010-2018.*

a) ICON266



b) ICPL266

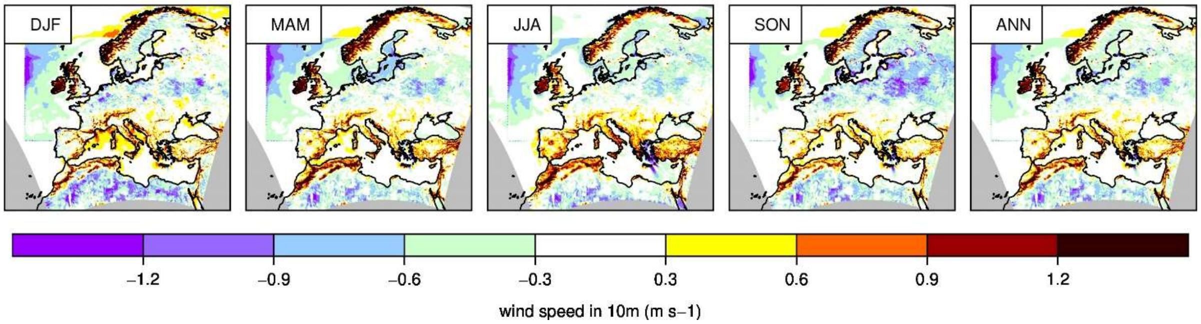


*Figure S11: Seasonal (DJF, MAM, JJA, SON) and annual (ANN) mean of 10-M wind speed (m/s) difference for a) ICON266 and b) ICPL266 compared to the ERA5 data for the period of 2010-2018.*