SCALE
USERS GUIDE

Version 5.5.1

Team SCALE
UGC working group

April 12, 2024

SCALE

Scalable Computing for Advanced Library and Environment

Contents

Part 1

Overview

Chapter 1.1

Introduction

This user's manual is intended for first-time users of the regional climate/weather forecasting model SCALE-RM . The manual is based on the meteorology/climate library SCALE version 5.5.1. The current version of SCALE contains a regional model SCALE-RM and a global model SCALE-GM . This version of the user's manual explains how to use SCALE-RM in detail. A description of SCALE-GM is planned for the next release.

The structure of this document is as follows: Part 1 provides an overview of SCALE , Part 2 describes how to install SCALE-RM along with the system requirements. Using simple examples, Chapters 3.1 and 3.2 explain the basic use of SCALE-RM using examples of an ideal experiment and an real atmospheric experiment, respectively. Since these chapters are constructed as a series of tutorials, it is recommended that beginning users of SCALE-RM read these chapters meticulously. Parts 4 and 5 describe how to change the model configuration and explain data format and available functions and tools. Since each section is closed itself basically, these chapter can be utilized as a dictionary.

If you have any questions or comments, please contact us through the users' mailing list ⟨`scale-users@ml.riken.jp`⟩.

1.1.1   What is SCALE ?

Scalable Computing for Advanced Library and Environment, called SCALE , is a software library that helps conduct research on climate and weather forecasting on any computer with ease. This library covers all processes related to such research: pre-processing, simulation, post-processing, and analysis. It has the following advantages:

- SCALE is provided as an open-source software under the " BSD-2 license". It is free for use, modification, and redistribution, regardless of whether the user is a business or other enterprise.

- SCALE contains a regional model called the SCALE-RM (SCALE-Regional Model).

- SCALE prepares various schemes in a component, as outlined in the next section. They can be appropriately chosen according to the desired experiments of users.

- SCALE provides a framework for physical processes that can be called not only by SCALE-RM , but also by other numerical models.

For the details of the license, the interested reader can refer to the file `scale-5.5.1/LICENSE` under the main directory. This explanation of its use is also provided on the SCALE webpage (https://scale.riken.jp/).

In this section, the concept of SCALE and its relations to actual models are explained. It can be skipped, as it is not related directly to its practical use.

Relations between SCALE library and models
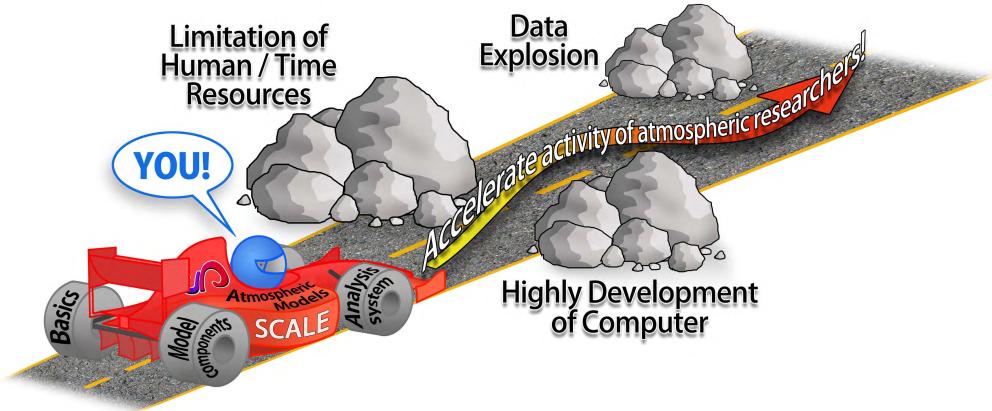
Figure 1.1.1: Aims of SCALE

SCALE was developed in RIKEN with several outside contributors, and its improvement and extension continue. Figure 1.1.1 shows the schematic concept of SCALE . As shown in this figure, SCALE aims to resolve various problems. The development of SCALE is considered in the context of its wide use by devices ranging from small PC clusters to next-generation supercomputers. For this purpose, scientists in meteorology/climate science and computer science are cooperating.

SCALE-RM is a numerical model that fully uses SCALE . This model is contained in the SCALE package, as shown in Fig. 1.1.2. SCALE manages the parallel processes, file I/O, and inner-communication. SCALE also provides the solver for atmospheric flow (dynamical core) and physical processes such as micro-physics and radiation processes. On the other hand, SCALE-RM is constructed by combining functions provided by SCALE . SCALE-RM itself reads the input data of atmospheric status as prognostic variable, and conducts time-integration. Users can select a scheme in every component according to simulations they want.
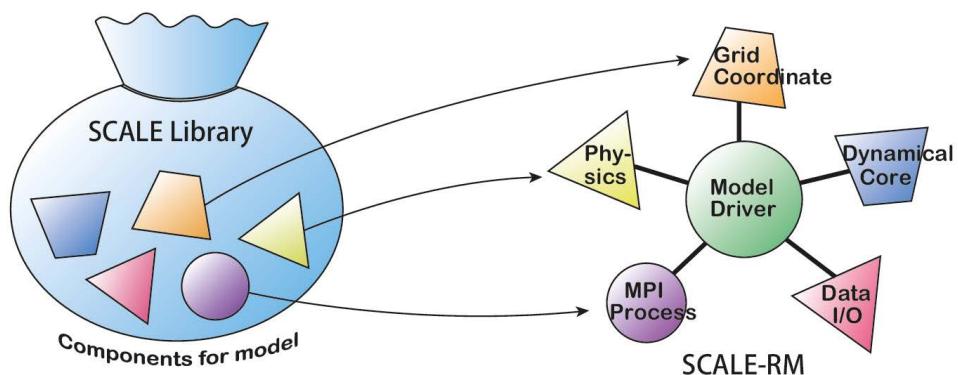
Figure 1.1.2: Relationship between the library SCALE and the model SCALE-RM

1.1.2   Structure of SCALE-RM

All schemes in all components of SCALE are available in SCALE-RM . The components are categorized into three parts: framework, dynamical core, and physical processes. Components with various schemes already implemented in the current version of SCALE-RM are listed below[*1)].

Framework

- The three-dimensional (3D) Cartesian grid system based on actual distance

- 2D domain decomposition by Message Passing Interface (MPI) communication

- Several map projections commonly used

- Domain nesting system (one-way, i.e., data transfer from parent domain to child domain.)

  - On-line nesting: concurrent execution of multiple domains).
  - Offline nesting: execution of computation in an inner domain after that in an outer domain.

- Collective execution system of multiple cases, i.e., bulk job system

- netCDF file I/O based on CF (Climate and Forecast) convention[*2)]

  - Selection of netCDF 3 and netCDF 4 formats

- Generation of initial data for an ideal experiment

- Generation of topographical and land-use data, converted from external data

- Generation of initial and boundary data from external data

  - GrADS [*3)] formats.

Dynamical core

- Governing equations: 3D fully compressible non-hydrostatic equations

- Spatial discretization: finite volume method

  - central advection schemes with 2nd, 4th, 6th, and 8th-order accuracy
  - upwind advection schemes with 3rd, 5th, and 7th-order accuracy

- Time integration: selection from the "fully explicit method" (HEVE) or the "horizontally explicit and vertically implicit methods" (HEVI)

  - Wicker and Skamarock (2002)'s 3 stage Runge–Kutta scheme with generally 2nd order accuracy
  - Heun-type 3 stage Runge–Kutta scheme with 3rd order accuracy
  - 4 stage Runge–Kutta scheme with 4th order accuracy
  - 7 stage Runge–Kutta scheme with 6th order accuracy (supported only for HEVE)
  - 11 stage Runge–Kutta scheme with 8th order accuracy (supported only for HEVE)

- Guarantee of non-negative value:

  - Flux corrected transport method (Zalesak, 1979)

[*1)] Refer to SCALE (2015),Sato et al. (2015), and Nishizawa et al. (2015) for the details of the model structure and the discretization method.
[*2)] http://cfconventions.org/
[*3)] http://cola.gmu.edu/grads/

– Koren (1993)'s filter: available only with the use of the 3rd-order upwind advection scheme

- Numerical filter: hyper-viscosity and diffusion with 2nd, 4th, 6th and 8th-order differential operators

- Topography: expressed using terrain-following coordinates

Physical processes

- Turbulence process: selectable from among the following

  – Smagorinsky (1963) & Lilly (1962)-type sub-grid scale turbulent model with the corrections by Brown et al. (1994) and Scotti et al. (1993)
  – Deardorff (1980) sub-grid scale turbulent model
  – MYNN level 2.5 boundary scheme (Mellor and Yamada (1982); Nakanishi and Niino (2004, 2009))

- Cloud microphysics: selectable from among the following

  – 3-class 1 moment bulk scheme (Kessler, 1969)
  – 6-class 1 moment bulk scheme (Tomita, 2008)
  – 6-class 2 moment bulk scheme (Seiki and Nakajima, 2014)
  – spectral bin scheme (Suzuki et al., 2010)

- Radiation process: a k-distribution-based broadband radiation transfer model (Sekiguchi and Nakajima (2008))

- Surface models

  – Land model: heat diffusion/bucket model
  – Ocean model: selectable from among the following
    * fixed to initial condition
    * input from external data
    * slab model
  – Urban model: a single-layer canopy model (Kusaka et al., 2001)
  – Heat transfer coefficient at land and in ocean: selectable from among the following
    * The bulk method using the universal function (Beljaars and Holtslag, 1991; Wilson, 2001; Nishizawa and Kitamura, 2018)
    * Louis-type bulk method (Uno et al., 1995)

Chapter 1.2

Notations used in this document

This document assumes an execution in the shell "bash" on some Unix system. If your environment is different, replace the commands by the relevant commands suitable for your environment. Unless there is a particular remark, this documentation obeys the following notation:

The command-line symbol for execution is expressed by `$` or `#`. The difference between the two notations is in the permission levels of program execution, as shown below:

```
#         <- command by root permission
$         <- command by user permission
```

A description enclosed in a rectangle expresses a message generated by the command line, as shown below.

```
− − − − command-line message
− − − − − − − − command-line message
− − − − − − − − − − − command-line message
```

On the other hand, a description enclosed in a rounded rectangle shows a part of configuration file including namelists, which can be edited as needed.

```
− − − − description in a file
− − − − − − − − description in a file
− − − − − − − − − − − description in a file
```

In this documentation, the FORTRAN namelist and its items are denoted by [namelist] and (item_of_namelist), respectively.

Part 2

Install

Chapter 2.1

Preparation

This chapter explains how to compile SCALE / SCALE-RM and the minimum computational requirements for their execution.

2.1.1 System environment

Recommended Hardwares

Although the necessary hardware depends on an experiment to be performed, the following are the specifications for conducting the tutorial as in Chapters 3.1 and 3.2.

- **CPU** : The system need to have more than or equal to two physical cores for the ideal experiment in the tutorial, and more than or equal to four for the real atmospheric experiment.

- **Memory** : The system needs 512 MB and 2 GB memory for the ideal experiment and the real atmospheric experiment, respectively. Note that this applies to the case involving the use of double-precision floating points.

- **HDD** : The system needs to have 3 GB of free disk space in the real atmospheric experiment.

Required Softwares

- **OS** : Linux OS, Mac OS

- **Compiler** : C, Fortran

Since the source code of SCALE is written in FORTRAN 2008 standard syntax, the compiler must support it. For example, GNU gfortran version 4.8 or later versions is necessary. Refer to Table 2.1.1 for compilers already confirmed as supported.

Required libraries

The required external libraries are described below:

- netCDF Library (http://www.unidata.ucar.edu/software/netcdf/)

Table 2.1.1: Compiler already checked

Name of compiler	
GNU (gcc/gfortran)	Version 4.8 or later is supported.
Intel (icc/ifort)	Version 15.0.0 or later is supported.
NVIDIA HPC SDK (nvcc/nvfortran)	

Table 2.1.2: Compiler list

Name of MPI library	
openMPI	Version 1.7.2 or later is supported.
Intel MPI	Version 5.0 or later is supported.

- MPI Library (e.g., openMPI http://www.open-mpi.org/)

- LAPACK (http://www.netlib.org/lapack/) (only required for SCALE-GM)

netCDF 4 is recommended, but netCDF 3 is also available. Note that with netCDF 3 there are some restrictions (See Section 5.1.1). You can also use distributed binary packages for Linux and Mac.

The MPI library should support the MPI 1.0/2.0 protocol. Refer to Table 2.1.2 for MPI libraries already confirmed as supported.

The tutorials in Chapters 3.1 and 3.2 assume that the above library environment has been prepared.

Drawing tools

In this subsection, drawing tools that can draw the initial conditions, boundary data, and the simulation results with SCALE-RM are introduced.

The GPhys and GrADS are used for a quick view and the drawing model output in the tutorial in Chapters 3.1 and 3.2, respectively. Other tools are also available, if they can be read in netCDF file format, which is the output of SCALE-RM .

- GPhys / Ruby-DCL by GFD DENNOU Club

  - URL: http://ruby.gfd-dennou.org/products/gphys/
  - Note: SCALE outputs the split files in netCDF format according to domain decomposition by the MPI process. "gpview" and/or "gpvect" in GPhys can directly draw the split file without post-processing.
  - How to install: On the GFD DENNOU Club webpage, the installation is explained for major OSs
    http://ruby.gfd-dennou.org/tutorial/install/

- Grid Analysis and Display System (GrADS) by COLA

  - URL: http://cola.gmu.edu/grads/
  - Note:This is among the most popular drawing tools, but the split files with netCDF generated by SCALE are not directly readable. The post-processing tool SNO is needed to combine the output of SCALE in one file that is readable from GrADS . Refer to SNO for installation instructions in Section 2.3.1, and to Chapters 3 and 4 and Section 4.6.1 for details pertaining to its use.
  - How to install: Refer to http://cola.gmu.edu/grads/downloads.

- Ncview: netCDF visual browser developed by David W. Pierce

  - URL: http://meteora.ucsd.edu/~pierce/ncview_home_page.html
  - Note: Ncview is a quick viewer for the netCDF file format. Although it cannot combine split files in SCALE , it is useful in drawing the result file by file.
  - How to install: Refer to http://meteora.ucsd.edu/~pierce/ncview_home_page.html

Useful tools (not always necessary)

- **Data conversion tool**: wgrib, wgrib2, NCL
  These tools can generate input data readable by SCALE-RM . In the tutorial for the real atmospheric experiment, wgrib is used.

- **Evaluation tool of computational performance**:The PAPI library[*1)] is available.

[*1)] http://icl.utk.edu/papi/

Chapter 2.2

Compile of SCALE

2.2.1 Download and environmental settings

The environment used in the explanation below is as follows:

- CPU: Intel Core i5 2410M 2 core /4 thread

- Memory: DDR3-1333 4GB

- OS: CentOS 6.6 x86-64, CentOS 7.1 x86-64, openSUSE 13.2 x86-64

- GNU C/C++, FORTRAN compiler

Obtaining the source code

The source code for the latest release can be downloaded from https://scale.riken.jp/download/index.html.
The directory `scale-5.5.1/` can be seen when the tarball file of the source code is extracted.

```
$ tar -zxvf scale-5.5.1.tar.gz
$ ls ./scale-5.5.1/
```

Setting Makedef file and environment variables

SCALE is compiled using a Makedef file specified in the environment variable "`SCALE_SYS`." Several variations of the Makedef file corresponding to computer environment are prepared in the directory `scale-5.5.1/sysdep/`. Choose a Makedef file according to your environment. Table 2.2.1 shows environments that Makedef files have supported. If there is no file for your environment, create a Makedif file by modifying any existing one. For example, if you use Linux x86-64 OS, a GNU compiler and openMPI, "Makedef.Linux64-gnu-ompi" is apropriate. An environment variable must also be specified as follows:

```
$ export SCALE_SYS="Linux64-gnu-ompi"
```

If the environment is always the same, it is convenient to explicitly describe the environmental variable in the environmental configuration file, such as `.bashrc`.
   SCALE requires netCDF . In the most cases, the pathes of netCDF may be automatically found by using "nc-config" command. If the paths cannot be found automatically, you should set the following paths. You must set the environmental variables for netCDF as follows:

```
$ export SCALE_NETCDF_INCLUDE="-I/opt/netcdf/include"
$ export SCALE_NETCDF_LIBS= \
      "-L/opt/hdf5/lib64 -L/opt/netcdf/lib64 -lnetcdff -lnetcdf -lhdf5_hl -lhdf5 -lm -lz"
```

Table 2.2.1: Examples of environments and their corresponding Makedef files.

OS/Computer	Compiler	MPI	Makedef file
	gcc/gfortran	openMPI	Makedef.Linux64-gnu-ompi
Linux OS x86-64	icc/ifort	intelMPI	Makedef.Linux64-intel-impi
Mac OS	gcc/gfortran	openMPI	Makedef.MacOSX-gnu-ompi
Fujitsu FX700/FX1000	fccpx/frtpx	mpiccpx/mpifrtpx	Makedef.FX700

2.2.2 Compile

Compile of SCALE-RM

Move to the SCALE-RM source directory and compile it by executing the following command:

```
$ cd scale-5.5.1/scale-rm/src
$ make -j 4
```

The number of `-j` option shows a number of parallel compile processes; it is four-parallel in this example. To reduce elapsed time of compile, it is better to specify this option to execute parallel compilation. Specify this number according to your environment. When a compilation is successful, the following three executable files are generated under the `scale-5.5.1/bin` directory.

```
scale-rm  scale-rm_init  scale-rm_pp
```

Compile of SCALE-GM

Move to the SCALE-GM source directory and compile it by executing the following command:

```
$  cd scale-5.5.1/scale-gm/src
$  make -j 4
```

When a compilation is successful, the following three executable files are generated under the scale-5.5.1/bin directory.

When a compilation is successful, the following executable files are created in the `scale-5.5.1/bin` directory. "fio" is original format based on binary with header information.

```
scale-gm       (executable binary of \scalegm)
gm_fio_cat     (cat command tool for fio format)
gm_fio_dump    (dump tool for fio format file)
gm_fio_ico2ll  (convert tool from icosahedral grid data with fio format to LatLon grid data)
gm_fio_sel     (sel command tool for fio format)
gm_mkhgrid     (generation tool of icosahedral horizontal grid using spring grid)
gm_mkllmap     (generation tool of LatLon horizontal grid)
gm_mkmnginfo   (tool for creating management file of MPI process)
gm_mkrawgrid   (generation tool of icosahedral horizontal grid)
gm_mkvlayer    (generation tool of vertical grid)
```

Points to note

If you want to compile them again, remove the created binary files by the following command:

```
$ make clean
```

Note that, the library already compiled is not deleted by this command. When you recompile the files by changing the compilation environment and options, use the following command to remove all files created by compilation:

```
$ make allclean
```

In SCALE , a compilation and an archive are conducted in directory `scale-5.5.1/scalelib/`. Object files are placed in a hidden directory called `".lib"` under this directory when you carry out a compilation.

When you want to compile it in debug mode, compile it using `"make -j 4 SCALE_DEBUG=T"` (All the environment variables applied at compile time are listed in Table 2.2.2). If you want to change the compilation options in detail, edit `Makedef.***` file.

Table 2.2.2: List of environment variables applied at compile time

Environment variable	Description
SCALE_SYS	Selection of system (required)
SCALE_DEBUG	Use compile option for debug
SCALE_QUICKDEBUG	Use compile option for quick debug (detect floating point error while maintaining speed-up option)
SCALE_USE_SINGLEFP	Use single precision floating-point number (for all sources)
SCALE_ENABLE_OPENMP	Enable to use OpenMP
SCALE_ENABLE_OPENACC	Enable to use OpenACC
SCALE_USE_AGRESSIVEOPT	Conduct strong optimization (only for FX and intel, this can be a side effect)
SCALE_DISABLE_INTELVEC	Suppress option for vectorization (only with Intel compiler)
SCALE_NETCDF_INCLUDE	Include path of NetCDF library
SCALE_NETCDF_LIBS	Directory path of NetCDF library and specifying libraries
SCALE_ENABLE_PNETCDF	Use Parallel NetCDF
SCALE_COMPAT_NETCDF3	Limit to NetCDF3-compatible features
SCALE_ENABLE_MATHLIB	Use numerical calculation library
SCALE_MATHLIB_LIBS	Directory path of numerical library and specifying libraries
SCALE_ENABLE_PAPI	Use Performance Application Programming Interface (PAPI)
SCALE_PAPI_INCLUDE	Include path of PAPI library
SCALE_PAPI_LIBS	Directory path of PAPI library and specifying libraries
SCALE_DISABLE_LOCALBIN	Disable making local binary files in directory for test cases
SCALE_IGNORE_SRCDEP	Ignore dependency check at the compile time
SCALE_ENABLE_SDM	Use Super Droplet Method (SDM) model

Chapter 2.3

Compile of post-processing tool

2.3.1   Compilation of SNO

SNO is a post-processing tool for the SCALE-netCDF file generated by the SCALE version 5.3 or later. The output file of SCALE-RM is divided and stored in every computational node. SNO combines the output files (history file, `history.******.nc`) and converts them into a single netCDF file readable in GrADS . The compiled binary is used in the tutorials of Chapters 3.1 and 3.2. The details of how to use is explained in Section 4.6.1.

SNO uses the SCALE library, i.e., `libscale.a`, generated at the time of compilation of SCALE-RM , for the compilation of SNO . This library is located under the `scale-5.5.1/lib` directory. To compile SNO , execute the following command after the compilation of SCALE-RM :

```
$  cd scale-5.5.1/scale-rm/util/sno
$  make
```

If the compilation is succeeded, the executable binary file is generated under the `scale-5.5.1/bin` directory. The execution of SNO is as follows:

```
$  mpirun -n 2 ./sno sno.conf
```

In this example, SNO is executed with two MPI processes by using "mpirun" command. The last argument is the configuration file for SNO .

Part 3

Tutorial for SCALE-RM

Chapter 3.1

Operation check and basic usage

3.1.1 Introduction

In this chapter, the basic operations of SCALE-RM for numerical experiments are explained. For this purpose, an ideal experiment case is prepared. It is strongly recommended that the user perform this tutorial because it includes a check for whether the compilation of SCALE in Part 2 has been completed. This chapter assumes that the following file has been already generated:

```
scale-5.5.1/bin/scale-rm
scale-5.5.1/bin/scale-rm_init
scale-5.5.1/bin/scale-rm_pp
scale-5.5.1/bin/sno
```

Furthermore, GrADS is used as a drawing tool. "gpview" can also be used for the confirmation of the result. Refer to Section 2.1.1 for their installation procedures.

The tutorial is described in order of preparation: creating the initial data, conducting the simulation, post-processing the output, and drawing the results.

3.1.2 How to execute model

Experimental setup

For this tutorial for an ideal experiment, a case of cumulus cloud convection in a 2-dimensional model is used here. In this experiment, a cumulus appears and develops by providing a typical atmospheric vertical profile and an initial disturbance in the lower troposphere. Table 3.1.1 shows the experimental settings.

Preparations

This ideal experiment is conducted in the directory of `scale-rm/test/tutorial/ideal`. Move to this directory and create a static link to the executable binary in scale-5.5.1/bin as follows:

```
$ cd scale-rm/test/tutorial/ideal
$ ln -s ../../../../bin/scale-rm      ./
$ ln -s ../../../../bin/scale-rm_init ./
```

where "`scale-rm`" is the executable binary of the simulation and "`scale-rm_init`" is a tool for creating the initial and the boundary conditions.

Creating initial conditions

To create the initial conditions, the configuration file for `scale-rm_init` is required. The configuration file `init_R20kmDX500m.conf` has been prepared according to Table 3.1.1. Reading the

Table 3.1.1: Experimental settings of ideal experiment

	Configuration	Note
Number of MPI processes	1: eastward direction, 2: northward direction	2 MPI parallelization
Horizontal grid spacing	500 m: eastward direction, 500 m: northward direction	The 2-dimensional experiment in the northward and vertical directions.
Number of horizontal grids	1: eastward direction, 40: northward direction	
Number of vertical layers	97 layers with model top of 20 km	Stretched grid with finer resolution in the lower layers.
Lateral boundary condition	Periodic condition	Both eastward and northward
Time step	5 sec	10 sec in microphysics scheme
Integration time	3,600 sec	The total number of time step is 720.
Time interval of data output	300 sec	
Physical scheme	Only the microphysics scheme	6-class single-moment bulk model (Tomita, 2008)
Initial vertical profile	GCSS Case1 squall line (Redelsperger et al., 2000)	The wind profile based on Ooyama (2001) is given as the vertical shear.
Initial disturbance	Warm bubble	Horizontal radius of 4 km and vertical radius of 3km with maximum intensity of 3K.

configuration file, `scale-rm_init` calculates the stratified atmospheric structure and the initial disturbance.

The general form of the executable command in SCALE-RM is given as follows:

```
$ mpirun -n [the number of processes] \\
  [executable binary name] [the configuration file]
```

The number of processes using MPI parallel processing is given at [the number of processes]. The name of the executable binary is given to [executable binary name], such as `scale-rm`, `scale-rm_init`, and so on. The configuration file, where the experimental settings are described, is given to [the configuration file]. If `sample/init_R20kmDX500m.conf` is used as the configuration file and `scale-rm_init` using two-MPI parallel is executed, give the command as follows:

```
$ cp  sample/init_R20kmDX500m.conf  init_R20kmDX500m.conf
$ mpirun  -n  2  ./scale-rm_init  ./init_R20kmDX500m.conf
```

If it is successfully completed, the following message is output in the command line:

```
 *** Start Launch System for SCALE-RM
 *** Execute preprocess? : T
 *** Execute model? : F
 *** End Launch System for SCALE-RM
```

Through the above, the following three files are generated under the given directory:

```
init_LOG.pe000000
init_00000101-000000.000.pe000000.nc
init_00000101-000000.000.pe000001.nc
```

The entire calculation domain is horizontally divided by the number of MPI processes. The number followed by `pe` in the file name shows the process number of MPI. In log file `init_LOG.pe000000`, detailed information that is not displayed in the command-line is recorded. Although the two MPI processes are used in the case, only the log file of the 0th process (master rank) is output as default. If the execution is normally concluded, the statements below are output at the end of this LOG file:

```
+++++ Closing LOG file
```

The two files `init_00000101-000000.000.pe000000.nc` and `init_00000101-000000.000.pe000001.nc` are the initial-condition files, and each is approximately 600 KB. The file whose name ends with ".nc" is formatted by netCDF . It can be directly read by GPhys/Ruby-DCL and ncview.

Execution of simulation

The number of parallel processes is required to be the same as that when creating the initial conditions. The configuration file for the run is `sample/run_R20kmDX500m.conf`.

```
$ cp  sample/run_R20kmDX500m.conf  ./run_R20kmDX500m.conf
$ mpirun  -n  2  ./scale-rm  run_R20kmDX500m.conf
```

If a computer satisfying the necessary requirements is used, the calculation is concluded within two minutes. The following three files are then generated under the given directory:

```
LOG.pe000000
history.pe000000.nc
history.pe000001.nc
```

When the execution concludes normally, the following message is output at the end of this LOG file:

```
+++++ Closing LOG file
```

The two files `history.pe000000.nc` and `history.pe000001.nc` are history files containing the results of the calculation. They are formatted by netCDF , and size of each file is approximately 1.5 MB.

3.1.3 Post-processing and drawing

In this section, we explain post-processing and the method of drawing the calculation result. In the tutorial, the distributed files in netCDF format are merged into one file and converted into a single netCDF file that can be read by GrADS . The binary form makes it easy for users to analyze the result. Link to the post-processing tool SNO compiled in Section 2.3.1:

```
$ ln -s ../../../../bin/sno  ./
```

The method of execution of SNO is the same as that of SCALE-RM , i.e.,

```
$ mpirun -n [the number of the processes] ./sno [the configuration file]
```

The configuration file `sample/sno_R20kmDX500m.conf` is intended for special uses of SNO . Give this configuration file to SNO and execute it as follows:

```
$ cp  sample/sno_R20kmDX500m.conf  ./sno_R20kmDX500m.conf
$ mpirun  -n  2  ./sno  ./sno_R20kmDX500m.conf
```

If there is no error message and the following message is displayed to the standard output, the conversion is completed without problem:

```
 *** End SCALE-NetCDF Operator
```

The execution of SNO should be handled, so that the number of MPI processes is a divisor of the number of the grids in the calculation domain which does not contain the halo region. The following file is generated under the same directory by this execution:

```
merged_history.pe000000.nc
```

The netCDF file is the converted file obtained by merging the divided files, and readable in GrADS by "sdfopen" command without the "ctl" file.

To confirm whether the calculation is satisfactory, draw a figure using GrADS script `checkfig_ideal.gs`. Note that the grammar depends on the version of GrADS . If a warning appears, the GrADS script should be rewritten appropriately:

```
$ grads -blc checkfig_ideal.gs
```

If it is successfully completed, the following files are generated:

```
ideal_qhyd.png
ideal_W.png
```

The same figures as Fig. 3.1.1 can be found in the simulation, and post-processing is successfully concluded.

3.1.4 Guideline for further study

In this chapter, the method for the execution of SCALE-RM was explained by using a simple ideal experiment. We recommend studying methods of changing the model resolution, the calculation domain, and the number of MPI processes for further study. With regard to the ideal experiment, several files of other configurations, e.g., to increase resolution, the number of domains, and the physical scheme, are prepared in the directory "sample" under the same directory as was used in this experiment. These configuration files are useful to change such configurations. Moreover, various ideal experimental settings have been prepared in the directory "`scale-rm/test/case`." For some ideal experiments, it may be necessary to carry out the "make" command again in the same directory as in the configuration file because some test cases need special source codes according to their experimental settings. The procedures for the generation of the initial conditions and those for simulation execution are the same as in the tutorial in this chapter.

It is important to study the method for the configuration of physical processes, such as cloud microphysics, radiation, and turbulence schemes. Methods to alter them in detail are described in Part 4.
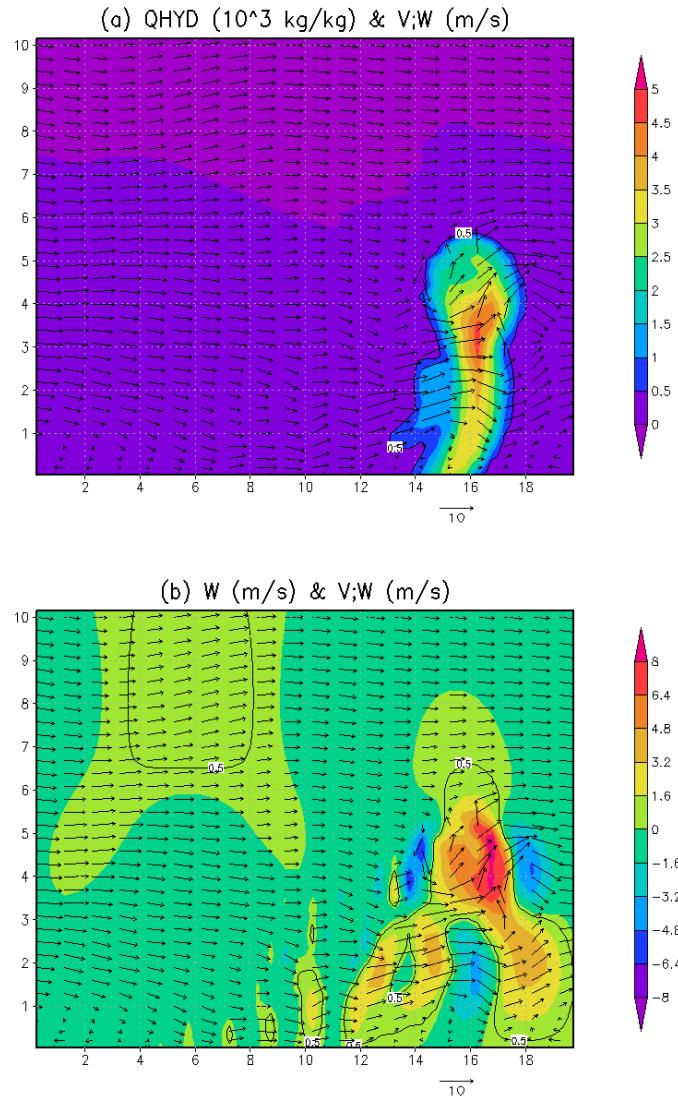
Figure 3.1.1: The horizontal-vertical cross-section after t=1200 s (20 minutes later); Figure (a) shows the mass concentration of the hydrometeor, and Figure (b) shows the vertical velocity. In both of figures, the vector indicates flow.

Chapter 3.2

Real atmosphere experiment

3.2.1 Overview

In this chapter, the basic execution procedure of the real atmospheric experiment is described using a simple case according to the workflow in Fig. 3.2.1.

1. Preparations for input data. The input data must be prepared by users themselves.

2. `pp`: Making topographical data

3. `init`: Making initial and boundary data

4. `run`: Executing the simulation

5. `sno`: Converting netCDF output data (optional)

Hereinafter, the absolute path `scale-5.5.1/scale-rm/test/tutorial/` is denoted by `${Tutorial_DIR}`.

The settings for the calculation domain used in this tutorial are given in Table 3.2.1. Figure 3.2.2 shows the target domain. Since this tutorial focuses on learning how to conduct real atmospheric experiments using SCALE-RM quickly, the experiment is designed to be completed in a short time. Note that this setting may not be appropriate as a physically valid experiment, and in practical application one should examine the experimental setup as needed.
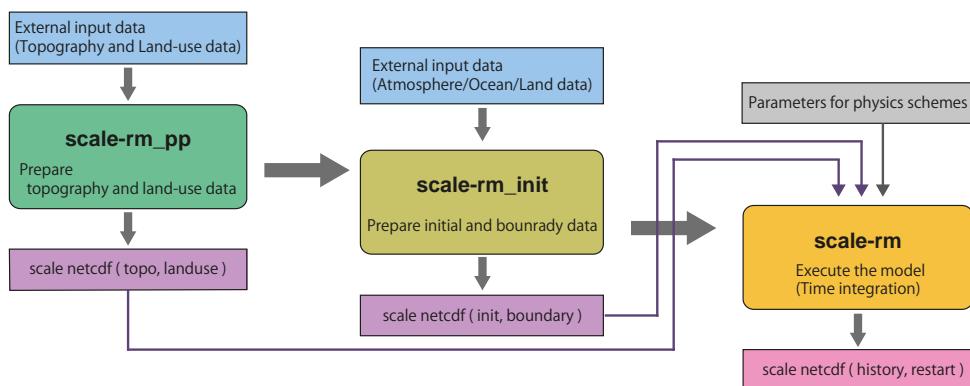
Figure 3.2.1: SCALE-RM procedure of model execution

Table 3.2.1: Overview of experimental settings

Item	Configuration
MPI process decomposition (east-west × north-south)	2 × 2 (total: 4 processes)
Number of horizontal grids (east-west × north-south)	90 × 90
Number of vertical layers	36
Horizontal grid intervals	$\Delta x = \Delta y = 20$km
Integration period	July 14, 2007, 18UTC - July 15 00UTC (6-hour integration)
Time step	90 s/step (total:240 steps)
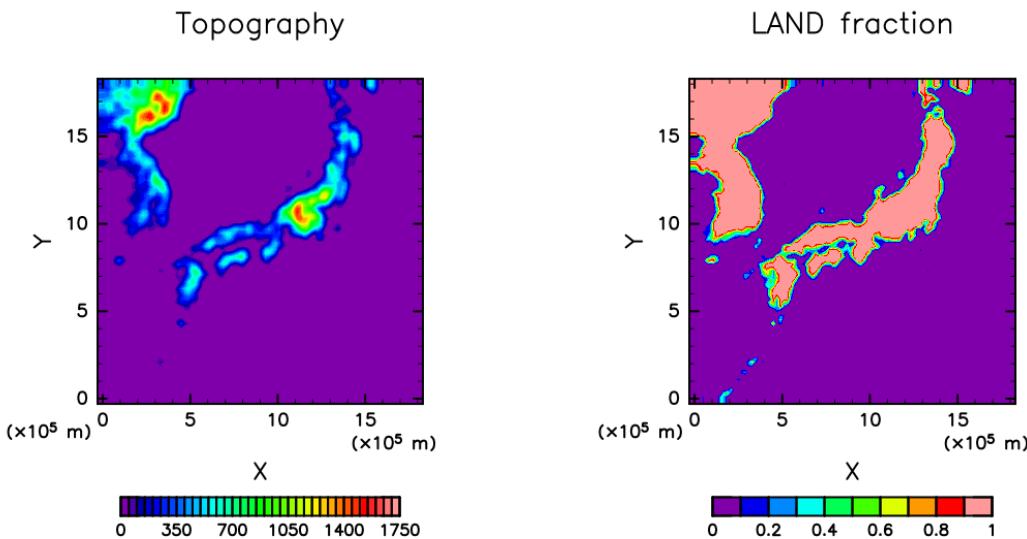
Figure 3.2.2: Topographical and land-ocean distribution in the domain

3.2.2 Preparations for input data (boundary data)

When a realistic atmospheric experiment is conducted, boundary data need to be provided to SCALE-RM . Table 3.2.2 shows the items for external input data to create boundary data. The variables denoted by blue character in this table are always needed, whereas the others (black character) are optional.

Topography and land-use classification data

The external topography and land-use classification data are used to obtain the altitude, the ocean to land ratio, the lake ratio, urban covering and vegetation rates, and the classification of land use at every grid point. In order to allow users to calculate at any areas over the globe, the altitude data GTOPO30 from the USGS (U.S. Geological Survey) and land-use classification data from the GLCCv2 are provided in SCALE-RM . These files have already been formatted for SCALE-RM .

1. Downloading database
   Obtain the data for altitude and land-use classification formatted for SCALE-RM from https://scale.riken.jp/archives/scale_database.tar.gz, and extract them to any directory:

Table 3.2.2: Items of external input data for real atmospheric experiments

Data to create the topography and land-use data (in general, in-situ data)
Altitude data
Land-use classification data

data to create the initial and boundary data for SCALE-RM (in general, GCM data)
Information for latitude and longitude of the parent model
— 3D atmospheric data —
zonal and meridional winds, temperature, specific humidity (relative humidity), pressure, geopotential height
— 2D atmospheric data—
sea-level pressure, surface pressure, zonal and meridional wind at 10m, temperature and specific humidity (relative humidity) at 2m
— 2D land data —
land and sea map in the parent model
surface skin temperature
information of depth of soil data in the parent model, soil temperature, soil moisture (volume content or degree of saturation)
— 2D ocean data at the surface —
sea surface temperature (omitted if skin temperature is also used for SST)

```
$ tar -zxvf scale_database.tar.gz
$ ls
   scale_database/topo/     <- altitude data
   scale_database/landuse/ <- land use classification data
```

2. Setting of the path
   To prepare the files used in the realistic atmospheric experiment, the "making tool for the complete settings of the experiment" is available. To use the tool, set the directory name including above database to an environment variable `SCALE_DB` (hereinafter, it denoted as `SCALE_DB`):

   ```
   $ export SCALE_DB="${path_to_directory_of_scale_database}/scale_database"
   ```

   where `${path_to_directory_of_scale_database}` is the directory name which the `tar` file including topography and land-use database is extracted in. For example, if the absolute path where you expanded `scale_database.tar.gz` was `/home/user/scale`, you need to set as follows.

   ```
   $ export SCALE_DB="/home/user/scale/scale_database"
   ```

Data for atmosphere, land, and sea surface temperature

The initial and boundary data are readable when they are converted into four-byte binary (GrADS form. Hereinafter, it is denoted by "binary data."). The users prepare the "binary" data by themselves as mentioned above. However, the programs to prepare the "binary" data are provided in the directory `${Tutorial_DIR}/real/tools/` for the execution of this tutorial. A procedure is explained below. Note that it is assumed that the installation of `wgrib`[*1)] is complete to use NCEP FNL (Final) Operational Global Analysis data with grib1 format.

1. Downloading data Download 12-hour data from July 14, 2007 1800UTC from the NCAR website http://rda.ucar.edu/datasets/ds083.2/ and place them in the directory `${Tutorial_DIR}/real/tools/FNL_input/grib1/2007`. The following is the data list, formatted by grib1:

[*1)] http://www.cpc.ncep.noaa.gov/products/wesley/wgrib.html

```
fnl_20070714_18_00.grib1
fnl_20070715_00_00.grib1
```

2. Conversion of data from grib to binary format
   Execute `convert_FNL-grib2grads.sh` in the directory `${Tutorial_DIR}/real/tools/`:

```
$ cd ${Tutorial_DIR}/real/tools/
$ sh convert_FNL-grib2grads.sh 2007071418 2007071500 FNL_input FNL_output
```

The following files are found if it is successful:

```
$ ls FNL_output/*/*
    FNL_output/200707/FNL_ATM_2007071418.grd
    FNL_output/200707/FNL_ATM_2007071500.grd
    FNL_output/200707/FNL_LND_2007071418.grd
    FNL_output/200707/FNL_LND_2007071500.grd
    FNL_output/200707/FNL_SFC_2007071418.grd
    FNL_output/200707/FNL_SFC_2007071500.grd
```

If the generation of the intended files is failed because of changing data format and variable names in NCEP-FNL data, `convert_FNL-grib2grads.sh` should be fixed according to the used NCEP-FNL data.

3.2.3   Preparation for experiment set

In the real atmospheric experiment, many execution procedures and a large number of files are needed in comparison with the ideal experiment. In addition, it is necessary to maintain consistency between settings in the configuration file (`***.conf`) between pre-processing `pp`, initial value making `init`, and simulation execution `run`. The inconsistency between these settings and the lack of files in the preparation step may cause an abnormal model run. To avoid such situations, the tool "the making tool for the complete settings of the experiment" is prepared for the generation of a set of necessary files. You first move to the following directory and prepare a series of files for the tutorial for the real atmospheric experiment using the next procedure:

```
$ cd ${Tutorial_DIR}/real/
$ ls
    Makefile : Makefile for generation of a set of necessary files.
    README   : README related to use of the script
    USER.sh  : Description of experimental setting.
    config/  : Each of configurations for generation of a set of files
               (basically, unnecessary for users to be rewritten)
    sample/  : sample script of USER.sh
    data/    : tools for the tutorial
    tools/   : tools for initial condition used in the tutorial
               (basically, users do it themselves except for the tutorial case)
$ make
$ ls experiment/      : directories added by the above make command
    init/
    pp/
    run/
    sno/
```

According to the settings described in `USER.sh`, an experiment set is generated under the directory `experiment` when the make command is executed. Refer to Section 4.2.1 for a detailed explanation of "the making tool for the complete settings of the experiment."

3.2.4 Creating topographical data: pp

Move to the directory pp and create topographical data for the experiment as follows:

```
$ cd ${Tutorial_DIR}/real/experiment/pp/
$ ls
   pp.d01.conf
   scale-rm_pp
```

In the directory pp, there exists configuration file pp.d01.conf. It is necessary to edit pp.d01.conf according to the experiment settings, such as the position of the domain and the number of grids. Since pp.d01.conf has already been edited for this tutorial, it can be used without any change. The setting of this experiment is shown in Table 3.2.1.

In Namelists in pp.d01.conf, the parameters related to the domain are configured in [PARAM_PRC_CARTESC], [PARAM_ATMOS_GRID_CARTESC_INDEX], and [PARAM_ATMOS_GRID_CATESC]. The domain is decomposed along each of the X and Y directions into two domains. Thus, four MPI processes are used. The total number of grids is (IMAXG)=90 and (JMAXG)=90 along the X and Y directions, respectively. Thus, the number of grids per MPI process is 45 (= 90/2). The grid spacings in each direction (DX, DY) in [PARAM_ATMOS_GRID_CARTESC] is 20,000 m (20 km). This means that the domain of calculation is an area of 1,800 km × 1,800 km because one side has length 90 × 20 km.

```
&PARAM_PRC_CARTESC
 PRC_NUM_X = 2,
 PRC_NUM_Y = 2,
 PRC_PERIODIC_X = .false.,
 PRC_PERIODIC_Y = .false.,
/

&PARAM_ATMOS_GRID_CARTESC_INDEX
 KMAX = 36,
 IMAXG = 90,
 JMAXG = 90,
/

&PARAM_ATMOS_GRID_CARTESC
 DX = 20000.0,
 DY = 20000.0,
 FZ(:) = 80.841, 248.821, 429.882, 625.045, 835.409, 1062.158,
       1306.565, 1570.008, 1853.969, 2160.047, 2489.963, 2845.575,
       3228.883, 3642.044, 4087.384, 4567.409, 5084.820, 5642.530,
       6243.676, 6891.642, 7590.074, 8342.904, 9154.367, 10029.028,
      10971.815, 11988.030, 13083.390, 14264.060, 15536.685, 16908.430,
      18387.010, 19980.750, 21698.615, 23550.275, 25546.155, 28113.205,
 BUFFER_DZ = 5000.0,
 BUFFER_DX = 400000.0,
 BUFFER_DY = 400000.0,
/
```

scale-rm_pp has a particular namelist of [PARAM_CONVERT]. If (CONVERT_TOPO)=.true., the altitudes are processed. If (CONVERT_LANDUSE)=.true., land-use classification data are processed.

```
&PARAM_CONVERT
 CONVERT_TOPO = .true.,
 CONVERT_LANDUSE = .true.,
/
```

(GTOPO30_IN_DIR) in [PARAM_CNVTOPO_GTOPO30] and (GLCCv2_IN_DIR) in [PARAM_CNVLANDUSE_GLCCv2] specify the locations of altitude data and land-use classification data, respectively.

```
&PARAM_CNVTOPO_GTOPO30
  GTOPO30_IN_DIR = "./topo/GTOPO30/Products",
  GTOPO30_IN_CATALOGUE = "GTOPO30_catalogue.txt",
/

&PARAM_CNVLANDUSE_GLCCv2
  GLCCv2_IN_DIR = "./landuse/GLCCv2/Products",
  GLCCv2_IN_CATALOGUE = "GLCCv2_catalogue.txt",
  limit_urban_fraction = 0.3D0,
/
```

After preparation of the configuration file, execute `scale-rm_pp` to create topographical data by the following command:

```
$ mpirun  -n  4  ./scale-rm_pp  pp.d01.conf
```

In the case of this tutorial, the number of MPI processes is four as in Table 3.2.1. When the job is finished normally, the following message is output at the end of the log file: `pp_LOG_d01.pe000000`.

```
+++++ Closing LOG file
```

Furthermore, the files `topo_d01.pe######.nc` (file size of approximately 310 KB) and `landuse_d01.pe######.nc` (file size of approximately 380 KB) are generated, dividing four files according to the MPI processes used. `######` represents the MPI process number. The information concerning altitude, the ocean and land ratio, the lake ratio, urban covering and vegetation rates, and the classification of land use are stored at every grid point in these files.

OPTION

When "gpview" is installed, you can confirm whether topographical data has been correctly generated by the following command:

```
$ gpview topo_d01.pe00000*@topo --aspect=1 --nocont --range 0:1750 --int 50
$ gpview landuse_d01.pe00000*@FRAC_LAND --aspect=1 --nocont
```

The same figure as Fig. 3.2.2 is generated if the results are correct.

3.2.5 Creating the initial and boundary data: init

Move to the directory `init` and create the initial and boundary data for the SCALE-RM simulation as follows:

```
$ cd ${Tutorial_DIR}/real/experiment/init
$ ls
   init.d01.conf
   init.launch.conf
   param.bucket.conf
   scale-rm_init
```

In the directory `init`, there exists the configuration file `init.d01.conf`. The file `init.launch.conf` also exists but is not used here. It is necessary to edit the file `init.d01.conf` according to such

experimental settings as `pp.d01.conf`. `init.d01.conf` has been already edited for this tutorial experiment as shown in Table 3.2.1. To create the initial and boundary data, the topographical data generated in the previous section is used. This is set in `init.d01.conf` to refer the relative path as follows:

```
&PARAM_TOPOGRAPHY
  TOPOGRAPHY_IN_BASENAME = "../pp/topo_d01",
/

&PARAM_LANDUSE
  LANDUSE_IN_BASENAME = "../pp/landuse_d01",
/
```

In particular, the contents of [PARAM_MKINIT_REAL_ATMOS], [PARAM_MKINIT_REAL_OCEAN] and [PARAM_MKINIT_REAL_LAND] are handled. It should be confirmed that the settings in `init.d01.conf` are correct.

```
&PARAM_MKINIT_REAL_ATMOS
  NUMBER_OF_FILES = 2,                      : number of files read
  FILETYPE_ORG = "GrADS",                   : choose from Table 4.1.2
  BASENAME_ORG = "namelist.grads_boundary.FNL.2005053112-2015011400",
  BASENAME_BOUNDARY = "boundary_d01",    : output name of boundary data
  BOUNDARY_UPDATE_DT = 21600.0,             : time interval of input data
  PARENT_MP_TYPE = 3,
  USE_FILE_DENSITY = .false.,               : use the atmospheric density in the parent model or
                                              not?
/

&PARAM_MKINIT_REAL_OCEAN
  .........
  INTRP_OCEAN_SFC_TEMP = "mask",            : how to treat the missing value of SST
  INTRP_OCEAN_TEMP = "mask",                : how to treat the missing value of SST
/

&PARAM_MKINIT_REAL_LAND
  ..........
  USE_FILE_LANDWATER = .true.,              : use soil moisture data in the parent model or not?
  INTRP_LAND_TEMP = "mask",                 : how to treat the missing value of soil temperature
  INTRP_LAND_WATER = "fill",                : how to treat the missing value of soil moisture
  INTRP_LAND_SFC_TEMP = "fill",             : how to treat the missing value of surface temperate
/
```

The file format of the meteorological field data is specified in (FILETYPE_ORG). In this case, it is given as "GrADS " to read data in GrADS format. Refer to Section 4.1.2 for the details of the input file.

Link the input data (FNL) that are converted into binary form in Section 3.2.2 to the current working directory. A shell script for this appropriate linkage is prepared as `"gradsinput-link_FNL.sh"` in the directory `${Tutorial_DIR}/real/data`:

```
$ cp ../../data/gradsinput-link_FNL.sh ./
$ sh gradsinput-link_FNL.sh
```

If the following linkages are found, it is successfully concluded.

```
ATM_00000.grd -> ../tools/FNL_output/200707/FNL_ATM_2007071418.grd
ATM_00001.grd -> ../tools/FNL_output/200707/FNL_ATM_2007071500.grd
LND_00000.grd -> ../tools/FNL_output/200707/FNL_LND_2007071418.grd
LND_00001.grd -> ../tools/FNL_output/200707/FNL_LND_2007071500.grd
SFC_00000.grd -> ../tools/FNL_output/200707/FNL_SFC_2007071418.grd
SFC_00001.grd -> ../tools/FNL_output/200707/FNL_SFC_2007071500.grd
```

Then, link a namelist file to the directory `init` to read the binary (GrADS) data format in SCALE.

```
$ ln -s ../../data/namelist.grads_boundary.FNL.2005053112-2015011400 ./
```

After the above preparations, execute the `scale-rm_init` using four MPI processes.

```
$ mpirun -n 4 ./scale-rm_init init.d01.conf
```

If the job finishes normally, the following files are generated:

```
$ ls
   boundary_d01.pe000000.nc
   boundary_d01.pe000001.nc
   boundary_d01.pe000002.nc
   boundary_d01.pe000003.nc
   init_d01_20070714-180000.000.pe000000.nc
   init_d01_20070714-180000.000.pe000001.nc
   init_d01_20070714-180000.000.pe000002.nc
   init_d01_20070714-180000.000.pe000003.nc
   init_LOG_d01.pe000000
```

The file `init_LOG_d01.pe000000` is a log file. The following message is output at the end of the file `init_LOG_d01.pe000000`:

```
 +++++ Closing LOG file
```

The file sizes of the boundary and initial data, `boundary_d01.pe######.nc` and `init_d01_20070714-180000.000.pe######.nc`, are approximately 18.9 MB and 12.6 MB, respectively, where `######` represents the MPI process number.

NOTICE: Reduction of memory usage when reading input data _____

In SCALE-RM , only the master process reads input data and broadcasts the data to each node. This algorithm can reduce file I/O and process the input data fast. On the other hand, the algorithm may cause insufficient memory error when reading large input data, especially on a high-performance computing system. To avoid the error, you can choose the method that each node reads input data by itself.

```
&PARAM_MKINIT_REAL_ATMOS
 SERIAL_PROC_READ = .false.,    : Does only the master process read input data?
/
```

The default setting is `.true.`, which means that the master process only reads input data. When setting `.false.`, each node accesses to only the data it needs, resulting in the reduction of memory usage. However, because the algorithm places a load on file I/O, the computational performance may deteriorate due to locking file access.

OPTION

When "gpview" is installed, one can confirm whether the initial and boundary data have been created correctly by the following command:

```
$ gpvect --wsn=1 --scalar --slice z=1500 --nocont --aspect=1 --range=0.002:0.016    \
         --int 0.001 --xintv=10 --yintv=10 --unit_vect                              \
         init_d01_20070714-180000.000.pe00*@QV  init_d01_20070714-180000.000.pe00*@MOMX \
         init_d01_20070714-180000.000.pe00*@MOMY --title "QV, MOMX, MOMY"
```

If the same figure as Fig. 3.2.3 is found, it is successfully concluded.
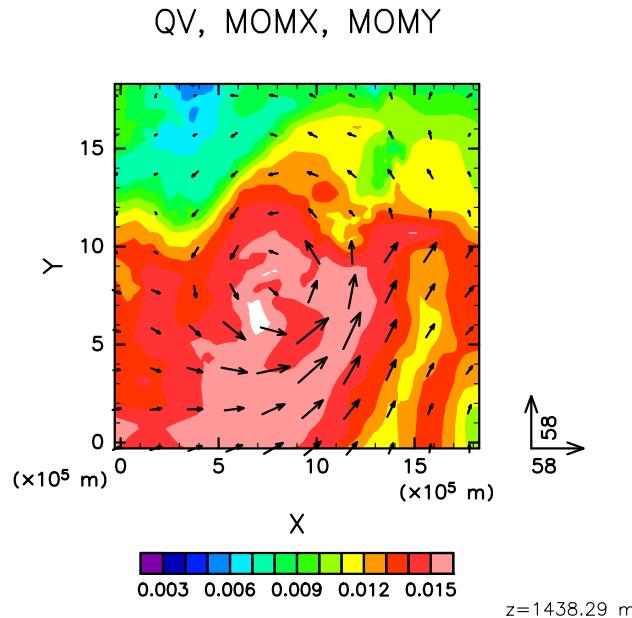
Figure 3.2.3: Initial field at z=1500m for the tutorial experiment. The color indicates specific humidity and the vector horizontal momentum flux.

3.2.6   Execution of simulation: run

Preparations for run.conf

Move to the directory `run`.

```
$ cd ${Tutorial_DIR}/real/experiment/run
```

In this directory, the configuration file `run.d01.conf` has already been prepared according to the tutorial settings shown in Table 3.2.1. Although the file `run.launch.conf` also exists in this directory, it is not used here.

The topographical data and the initial/boundary data previously generated are used for this execution of the simulation model. These files are configured in the following part of `run.d01.conf`:

```
&PARAM_TOPOGRAPHY
 TOPOGRAPHY_IN_BASENAME = "../pp/topo_d01",
/

&PARAM_LANDUSE
 LANDUSE_IN_BASENAME = "../pp/landuse_d01",
/

&PARAM_RESTART
 RESTART_OUTPUT = .true.,
 RESTART_OUT_BASENAME = "restart_d01",
 RESTART_IN_BASENAME = "../init/init_d01_20070714-180000.000",
/

&PARAM_ATMOS_BOUNDARY
 ATMOS_BOUNDARY_TYPE = "REAL",
 ATMOS_BOUNDARY_IN_BASENAME = "../init/boundary_d01",
 ATMOS_BOUNDARY_USE_DENS = .true.,
 ATMOS_BOUNDARY_USE_QHYD = .false.,
 ATMOS_BOUNDARY_ALPHAFACT_DENS = 1.0,
 ATMOS_BOUNDARY_LINEAR_H = .false.,
 ATMOS_BOUNDARY_EXP_H = 2.0,
/
```

With regard to temporal integration, it is configured in [PARAM_TIME] in run.d01.conf. Specify the initial time in (TIME_STARTDATE) in UTC. In the tutorial, it is given as 18:00 UTC on July 14, 2007. The integration period is specified at (TIME_DURATION). The time steps for the physical processes can be configured process by process.

```
&PARAM_TIME
 TIME_STARTDATE = 2007, 7, 14, 18, 0, 0,    : starting time of integration
 TIME_STARTMS = 0.D0,
 TIME_DURATION = 6.0D0,                      : integration period
 TIME_DURATION_UNIT = "HOUR",               : unit of TIME_DURATION
 TIME_DT = 90.0D0,                          : time step for tracer advection
 TIME_DT_UNIT = "SEC",                      : unit of TIME_DT
 TIME_DT_ATMOS_DYN = 45.0D0,                : time step of dynamical process except for tracer
                                              advection
 TIME_DT_ATMOS_DYN_UNIT = "SEC",            : unit of TIME_DT_ATMOS_DYN

 ...............

/
```

The output of the results of the calculation are configured in (PARAM_FILE_HISTORY).

```
&PARAM_FILE_HISTORY
 FILE_HISTORY_DEFAULT_BASENAME = "history_d01",    : output file name
 FILE_HISTORY_DEFAULT_TINTERVAL = 3600.D0,         : time interval for output
 FILE_HISTORY_DEFAULT_TUNIT = "SEC",               : unit of output time interval
 FILE_HISTORY_DEFAULT_TSTATS_OP = "none",
 FILE_HISTORY_DEFAULT_DATATYPE = "REAL4",
 FILE_HISTORY_DEFAULT_ZCOORD = "model",            : no vertical interpolation
 FILE_HISTORY_OUTPUT_STEP0 = .true.,               : output at initial time(t=0) or not
/
```

According to the above setting, variables listed in the following (HISTORY_ITEM) are output. It

is possible to change output intervals at every variable in (HISTORY_ITEM) by adding options if needed. The mean value instead of the snapshot value can be output. Refer to Section 4.2.6 for the details.

```
&HISTORY_ITEM NAME="MSLP" /        : Sea-level pressure
&HISTORY_ITEM NAME="PREC" /        : Precipitation intensity (2D)
&HISTORY_ITEM NAME="OLR" /         : Outgoing longwave radiation (2D)
&HISTORY_ITEM NAME="U10m" /        : Zonal wind vector at 10 m (2D)
&HISTORY_ITEM NAME="V10m" /        : Meridional wind vector at 10 m (2D)
&HISTORY_ITEM NAME="U10" /         : Horizontal wind speed along X direction at 10 m(2D)
&HISTORY_ITEM NAME="V10" /         : Horizontal wind speed along Y direction at 10 m(2D)
&HISTORY_ITEM NAME="T2" /          : Temperature at 2m (2D)
&HISTORY_ITEM NAME="Q2" /          : Specific humidity at 2m (2D)
&HISTORY_ITEM NAME="SFC_PRES" /    : Surface pressure (2D)
&HISTORY_ITEM NAME="SFC_TEMP" /    : Bulk surface temperature(2D)
&HISTORY_ITEM NAME="DENS" /        : Density (3D)
&HISTORY_ITEM NAME="QV" /          : Specific humidity (3D)
&HISTORY_ITEM NAME="QHYD" /        : Mass concentration of total hydrometeor (3D)
&HISTORY_ITEM NAME="PRES" /        : Pressure (3D)
&HISTORY_ITEM NAME="Umet" /        : Zonal wind vector (3D)
&HISTORY_ITEM NAME="Vmet" /        : Meridional wind vector (3D)
&HISTORY_ITEM NAME="U" /           : Horizontal wind speed along X direction (3D)
&HISTORY_ITEM NAME="V" /           : Horizontal wind speed along Y direction (3D)
&HISTORY_ITEM NAME="T" /           : Temperature (3D)
&HISTORY_ITEM NAME="W" /           : Vertical wind speed (3D)
&HISTORY_ITEM NAME="Uabs" /        : Absolute value of wind velocity(3D)
&HISTORY_ITEM NAME="PT" /          : Potential temperature (3D)
&HISTORY_ITEM NAME="RH" /          : Relative humidity (3D)
```

If other schemes for the dynamics process and the physical processes are used, configure [PARAM_ATMOS_DYN] for the dynamical process and [PARAM_ATMOS, PARAM_OCEAN, PARAM_LAND, PARAM_URBAN] for the physical processes. Refer to Sections 4.3.1 and 4.4 for the details.

Execution of simulation

The lists below are the necessary files for the execution. They have already been prepared:

```
$ ls
   MIPAS  PARAG.29  PARAPC.29  VARDATA.RM29  cira.nc
                     : parameter file for radiation scheme
   run.d01.conf       : configuration file
   param.bucket.conf : parameter file for land schemes
   scale-rm           : executable binary of SCALE-RM
   run.launch.conf    : launch file for nesting calculations (not used in the tutorial)
```

If all preparations are complete, execute SCALE-RM by using four-MPI parallel:

```
$ mpirun -n 4 ./scale-rm run.d01.conf >& log &
```

The execution takes some time to complete. It takes 10-20 minutes in the recommended environment. It is convenient to execute it as a background job so that the standard output is written to a file. The processed results are output to the file "LOG_d01.pe000000" while the computation is progressing. If the job finishes normally, the following messages are output in the file:

```
+++++ Closing LOG file
```

The following files are also generated:

```
$ ls
 history_d01.pe000000.nc
 history_d01.pe000001.nc
 history_d01.pe000002.nc
 history_d01.pe000003.nc
```

The size of each file is approximately 34 MB. The output files (`history_d01.pe######.nc`) are split according to number of MPI processes, where `######` represents the MPI process number. In these files, the variables specified in `(HISTORY_ITEM)` are output. The files are formatted by NetCDF, corresponding to climate and forecast (CF) metadata convention.

3.2.7 Quick look at simulation result: sno

In this section, we explain how to use SNO. The SNO program merges the netCDF files (`history.**.nc`) [*2], which are divided process by process, into a single netCDF file that can be read in GrADS . The simulation results are also validated using the converted netCDF data.

Conversion to GrADS binary

Use `sno` to convert a single netCDF file from the history files of netCDF having divided every process. Only a minimum of the procedures is explained here. Refer to Section 4.6.1 for their detailed use.

Move to the directory `sno`:

```
$ cd ${Tutorial_DIR}/real/experiment/sno
$ ls
   sno -> ../../../../../../bin/sno
   sno.hgridope.d01.conf
   sno.vgridope.d01.conf
```

There are some configuration files and a binary file in this directory. The binary file is linked to the executable file compiled in Section 2.3.1. As an example, a procedure to convert 3-dim variables on model levels to pressure levels and interpolate to equivalent longitude-latitude data is explained here.

Since `sno` cannot simultaneously convert the data for the vertical and horizontal direction, they must be converted separately as follows. First, `sno` converts the data from model levels to pressure levels. When `sno` is executed, the number of processes needs to be a divisor of the number of the grids in the calculation domain. The number of processes used here is four.

```
$ mpirun -n 4 ./sno sno.vgridope.d01.conf
```

Next, `sno` converts the vertically interpolated data horizontally to longitude-latitude data and then combines the node-separated netCDF files into a single file. Due to the limitation of the number of processes that can be used in parallel execution of SNO , one process is used here.

```
$ mpirun -n 1 ./sno sno.hgridope.d01.conf
```

The conversion succeeds only if the following messages are found in the standard output without an error message:

```
   *** End SCALE-NetCDF Operator
```

If the process has succeeded, the below files are created.

[*2] If "gpview" is installed, it can also be used for drawing. This tool is more suitable for quick check because it is available without the conversion of history data.

```
merged-p_history_d01.pe000000.nc
merged-p_history_d01.pe000001.nc
merged-p_history_d01.pe000002.nc
merged-p_history_d01.pe000003.nc
merged-h_history_d01.pe000000.nc
```

Validation of simulation result

Confirm the calculation results using a GrADS script `checkfig_real.gs`:

```
$ cp ../../data/checkfig_real.gs ./
$ grads -blc checkfig_real.gs
```

The following files are generated when the conversion is successfully finished. Note that the script changes accordingly when a warning appears. This is because the syntax is different according to the version of GrADS .

```
real_mslp.png
real_prec.png
real_wind.png
```

If the calculation is successful, the same figures as Figs. 3.2.4, 3.2.5, and 3.2.6 are obtained.
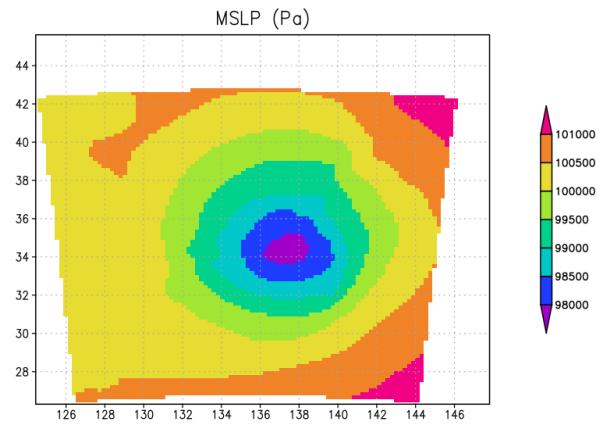
Figure 3.2.4: Sea-level pressure after 6 hours
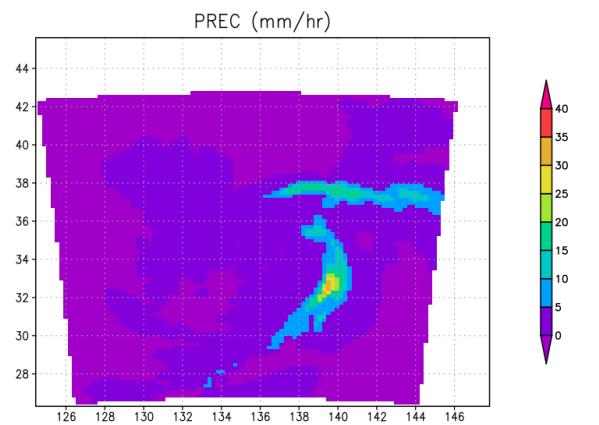
Figure 3.2.5: Precipitation flux after 6 hours
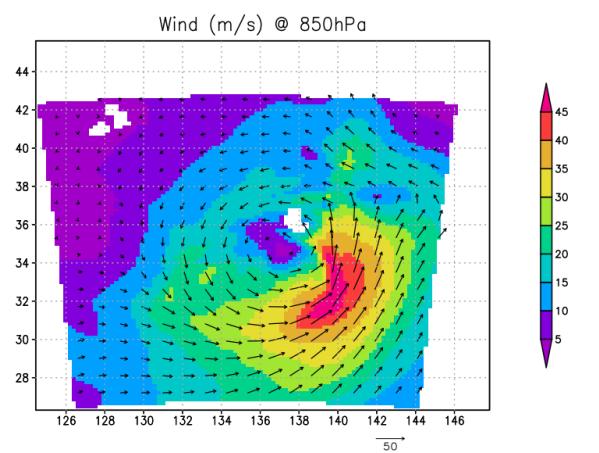
Figure 3.2.6: Wind velocity after 6 hours

Part 4

Various settings

Chapter 4.1

Pre-processing

4.1.1 Setting the topography

SCALE-RM employs the terrain-following coordinates to represent topography. In these coordinates, the bottom face of the lowest grid is given such that it can follow the surface altitude, i.e., topography. Topography data used in the program `scale-rm_init` and `scale-rm` should be prepared in SCALE format beforehand, except for some ideal experiments. SCALE-RM has a pre-processing function to convert external topography data to the format.

The program `scale-rm_pp` converts external topography data when `(CONVERT_TOPO)` in `[PARAM_CONVERT]` is `.true.`. Then, the converted data is saved in the file specified by `(TOPOGRAPHY_OUT_BASENAME)` in `[PARAM_TOPOGRAPHY]`. The generated file is used in `scale-rm_init` and `scale-rm` by setting to `(TOPOGRAPHY_IN_BASENAME)` in `[PARAM_TOPOGRAPHY]`.

```
&PARAM_CONVERT
 CONVERT_TOPO = .true.,                      ; Whether execute the conversion of topography
                                               data

 ......
/
&PARAM_TOPOGRAPHY
 TOPOGRAPHY_IN_BASENAME = ''                 ; (Input) Base name of input topography file
 TOPOGRAPHY_IN_VARNAME = 'topo'             ; (Input) Variable name of topography in the in-
                                               put file
 TOPOGRAPHY_IN_AGGREGATE = .false.          ; (Input) Whether read from one file with
                                               PnetCDF
 TOPOGRAPHY_IN_CHECK_COORDINATES = .false.  ; (Input) Whether check coordinate variables in
                                               the input file
 TOPOGRAPHY_OUT_BASENAME = ''                ; (Output) Base name of output file
 TOPOGRAPHY_OUT_AGGREGATE = .false.         ; (Output) Whether aggregate to one file with
                                               PnetCDF
 TOPOGRAPHY_OUT_DTYPE = 'DEFAULT'           ; (Output) Data type to output
                                                   ("DEFAULT","REAL4","REAL8")
/
```

The configurations for the conversion, including information of external input data, are specified in `[PARAM_CNVTOPO]`.

```
&PARAM_CNVTOPO
  CNVTOPO_name = 'NONE',                    ; 'NONE','GTOPO30','DEM50M', or 'USERFILE'
  CNVTOPO_UseGTOPO30 = .false.,             ; Use GTOPO30 dataset?
  CNVTOPO_UseDEM50M = .false.,              ; Use DEM50M dataset?
  CNVTOPO_UseUSERFILE = .false.,            ; Use user-defined dataset?
  CNVTOPO_smooth_type = 'LAPLACIAN',        ; Type of smoothing filter
                                              ("OFF", "LAPLACIAN", "GAUSSIAN")
  CNVTOPO_smooth_maxslope_ratio = 5.D0,     ; Maximum allowable ratio of inclination to Δz/Δx
  CNVTOPO_smooth_maxslope = -1.D0,          ; Maximum allowable angle of inclination [deg]
  CNVTOPO_smooth_local = .true.,            ; Try to continue smoothing, for only grids whose an-
                                              gles of inclination exceed the maximum value?
  CNVTOPO_smooth_trim_ocean = .true.        ; Switch to keep ocean coast line
  CNVTOPO_smooth_itelim = 10000,            ; Number limit of the smoothing iteration
  CNVTOPO_smooth_hypdiff_niter = 20,        ; Number of the repetitions of the hyperdiffusion
  CNVTOPO_smooth_hypdiff_order = 4,         ; The order of the hyperdiffusion
  CNVTOPO_copy_parent = .false.,            ; The topography in the buffer region of child domain
                                              is copied from parent domain?

/
```

As external topography data, SCALE-RM supports GTOPO30 provided by the U.S. Geological Survey and DEM50M provided by the Geospatial Information Authority of Japan. The user-defined binary data is also supported (refer to Section 4.1.1.1 for the details).

The types of the external data used are specified by `(CNVTOPO_(UseGTOPO30|UseDEM50M|UseUSERFILE))` and/or `(CNVTOPO_name)`. It the switches of `(CNVTOPO_(UseGTOPO30|UseDEM50M|UseUSERFILE))` are set to `.true.`, the GTOPO30, DEM50M, and user-defined data are used, respectively. The default settings of `(CNVTOPO_(UseGTOPO30|UseDEM50M|UseUSERFILE))` are `.false.`. User can use `(CNVTOPO_name)` to specify which data to be used, instead of the individual switches; preset values of `(CNVTOPO_(UseGTOPO30|UseDEM50M|UseUSERFILE))` are set by `(CNVTOPO_name)` as shown in Table 4.1.1. The mark of asterisk ($*$) in Table 4.1.1 indicates that settings of the `(CNVTOPO_(UseGTOPO30|UseDEM50M|UseUSERFILE))` are also valid.

The combination of these datasets is also available. Set all parameters of `(CNVTOPO_UseGTOPO30)`, `(CNVTOPO_UseDEM50M)`, and `(CNVTOPO_UseUSERFILE)`, which are to be used, to `.true.`.

Table 4.1.1: The relationship between `(CNVTOPO_name)` and `(CNVTOPO_UseGTOPO30)`, `(CNVTOPO_UseDEM50M)`, and `(CNVTOPO_UseUSERFILE)`.

CNVTOPO_name	CNVTOPO_UseGTOPO30	CNVTOPO_UseDEM50M	CNVTOPO_UseUSERFILE
NONE	$*$	$*$	$*$
GTOPO30	.true.	.false.	.false.
DEM50M	.false.	.true.	.false.
USERFILE	$*$	$*$	.true.

The program makes the data according to the settings described above as follows:

1) Set all simulation grid points to undefined value.

2) If `(CNVTOPO_UseGTOPO30)`=`.true.`, interpolate GTOPO30 dataset to the simulation grid points.

3) If `(CNVTOPO_UseDEM50M)`=`.true.`, interpolate DEM50M dataset to the simulation grid, and overwrite the data of step 2 where the interpolated DEM50M data is not the missing value.

4) If `(CNVTOPO_UseUSERFILE)`=`.true.`, interpolate the user-defined data to the simulation grid, and overwrite the data of step 3 where the interpolated user data is not the missing value.

5) Set zero where the grid points that remain undefined.

6) Apply smoothing.

The GTOPO30 and DEM50M data are interpolated with the bi-linear interpolation method, while the user-defined data is interpolated with the method specified in the configuration (see Section 4.1.1.1).

To use the GTOPO30 data, the directory where the data is stored and the path of the catalogue file must be specified by (GTOPO30_IN_DIR) and (GTOPO30_IN_CATALOGUE) in [PARAM_CNVTOPO_GTOPO30], respectively. If the data is stored as explained in Section 3.2.2, they should be $SCALE_DB/topo/GTOPO30/Products and $SCALE_DB/topo/GTOPO30/Products/GTOPO30_catalogue.txt, respectively ($SCALE_DB must be expanded to the actual path.). In the same manner, for the DEM50M data, (DEM50M_IN_DIR) and (DEM50_IN_CATALOGUE) in [PARAM_CNVTOPO_DEM50M] is need to be set such as $SCALE_DB/topo/DEM50M/Products and $SCALE_DB/topo/DEM50M/Products/DEM50M_catalogue.txt, respectively. Configurations for the user-defined data are explained in Section 4.1.1.1.

There are two types of filter for smoothing the elevation with the steep slope in the interpolated topography: Laplacian and Gaussian filters. The type is specified by (CNVTOPO_smooth_type). The Laplacian filter is used in default. In the smoothing operation, the filter is applied multiple times until the angle of the topography slope is below the maximum allowable angle $\theta_{\max}$. The maximum allowable angle [radian] is calculated as follows:

$$\theta_{\max} = \arctan(\text{RATIO} \times \Delta z / \Delta x),$$

where Δz and Δx are the vertical and horizontal grid spacings at the bottom layer, respectively. As shown in the above equation, $\theta_{\max}$ depends on spatial resolution. Greater RATIO results in a finer topography. Note that the risk of numerical instability increases as RATIO becomes greater. The RATIO is specified by (CNVTOPO_smooth_maxslope_ratio), and its default value is 0.5. Instead of (CNVTOPO_smooth_maxslope_ratio), you can directly set the maximum angle $\theta_{\max}$ by setting (CNVTOPO_smooth_maxslope) in degree. The number limit of the smoothing iteration is 10000 times in default. You can increase the limit by setting (CNVTOPO_smooth_itelim). When (CNVTOPO_smooth_local) is set to .true., the operation for smoothing is continued only at the grid points where the smoothing is not completed.

Additional hyperdiffusion is prepared to smooth the rugged terrain on a small spatial scale. We recommend applying this filtering to reduce the numerical noise induced by the rugged terrain in the simulation. The order of the hyperdiffusion is specified by (CNV_TOPO_smooth_hypdiff_norder). The hyperdiffusion is applied repeatedly for the number of times specified by (CONV_TOPO_smooth_hyperdiff_niter). If (CNVTOPO_smooth_hypdiff_niter) is set to negative, the hyperdiffusion is not applied.

The smoothing filter and the hyperdiffusion blur coast line and raise the elevation of the sea surface near the coast above zero. If (CNVTOPO_smooth_trim_ocean) is set to .true., the land is trimmed along the coast line in each iteration of the filtering operation and then the coast line is fixed. Note that the total volume of the soil is not conserved in this case.

(CNVTOPO_copy_parent) is the parameter used for the nesting computation. Since the spatial resolution of the child domain is higher than the parent domain in general, the topography in the child domain is finer than that in the parent domain. At this time, problems may occur due to an inconsistency between the atmospheric data in the buffer region of the child domain and the boundary data which is generated from the parent data. To avoid this problem, the topography in the buffer region of the child domain can be the same as that in the parent domain by setting (CNVTOPO_copy_parent)=.true.. If there is no parent domain, (CNVTOPO_copy_parent) must be .false.. Section 4.2.8.1 provides a more detailed explanation of the case that involves the use of (CNVTOPO_copy_parent).

4.1.1.1 Preparation of user-defined topography

When (CNVTOPO_UseUSERFILE) is set to .true., the scale-rm_pp creates the SCALE topography data from the user-defined data according to [PARAM_CNVTOPO_USERFILE]. The input data supports the type of "GrADS" and "TILE"; they are specified by (USERFILE_TYPE). The details for these file types are described in Section 4.1.3. The available items for [PARAM_CNVTOPO_USERFILE] as follows:

```
&PARAM_CNVTOPO_USERFILE
  USERFILE_TYPE = '',                      ; "GrADS" or "TILE"
  USERFILE_DTYPE = 'REAL4',                ; (for TILE) Type of the data
                                             ("INT2","INT4","REAL4","REAL8")
  USERFILE_DLAT = -1.0,                    ; (for TILE) Interval of the grid (latitude,degree)
  USERFILE_DLON = -1.0,                    ; (for TILE) Interval of the grid (longitude,degree)
  USERFILE_CATALOGUE = '',                 ; (for TILE) Name of the catalogue file
  USERFILE_DIR = '.',                      ; (for TILE) Directory path of input file
  USERFILE_yrevers = .false.,              ; (for TILE) If data is stored from north to south, set .true.
  USERFILE_MINVAL = 0.0,                   ; (for TILE) The data lower than MINVAL is recognized as
                                             missing value
  USERFILE_GrADS_FILENAME = '',            ; (for GrADS) Name of the namelist for the GrADS data
  USERFILE_GrADS_VARNAME = 'topo',         ; (for GrADS) Name of the target variable in the namelist
  USERFILE_GrADS_LATNAME = 'lat',          ; (for GrADS) Name of latitude in the namelist
  USERFILE_GrADS_LONNAME = 'lon',          ; (for GrADS) Name of longitude in the namelist
  USERFILE_INTERP_TYPE = 'LINEAR',         ; (for GrADS) Type for the horizontal interpolation
  USERFILE_INTERP_LEVEL = 5,               ; (for GrADS) Level of the interpolation
/
```

The read data is interpolated at the target grid. The type of the interpolation is specified by (USERFILE_INTERP_TYPE), and two types are supported: LINEAR and DIST-WEIGHT. Pleas see Section 4.1.2 for the details. The number of neighbor points is specified by (USERFILE_INTERP_LEVEL) in the case of DIST-WEIGHT.

For the "GrADS" type, the namelist file specified by (USERFILE_GrADS_FILENAME) is required to set the information of the GrADS binary data. Please see Section 4.1.2 for the details of the namelist file. As the default, the variable name for the topography, latitude, and longitude are "topo", "lat", and "lon", respectively. If the name is different from the default value, it can be specified with (USERFILE_GrADS_VARNAME), (USERFILE_GrADS_LATNAME), and (USERFILE_GrADS_LONNAME), respectively.

For the "TILE" type, the catalogue file specified by (USERFILE_CATALOGUE) is required; it contains information of the name of individual tiled data files and the area where the tiled data covers. Refer the catalogue files of the $SCALE_DB/topo/DEM50M/Products/DEM50M_catalogue.txt and $SCALE_DB/topo/GTOPO30/Products/GTOPO30_catalogue.txt as samples. The sample setting of [PARAM_CNVTOPO_USERFILE] for the "TILE" data is as follows. In this sample, the catalogue file named catalogue.txt is located in the directory ./input_topo. The data is stored with a 2-byte integer.

```
&PARAM_CNVTOPO_USERFILE
  USERFILE_CATALOGUE = "catalogue.txt",   ; Name of the catalogue file
  USERFILE_DIR = "./input_topo",          ; Directory path of input file
  USERFILE_DLAT = 0.0083333333333333D0,   ; Interval of the grid (latitude, degree)
  USERFILE_DLON = 0.0083333333333333D0,   ; Interval of the grid (longitude, degree)
  USERFILE_DTYPE = "INT2",                ; Type of the data
                                            ("INT2", "INT4", "REAL4", "REAL8")
  USERFILE_yrevers = .true.,              ; Data is stored from north to south in latitudinal
                                            direction?
/
```

4.1.2 How to prepare initial and boundary data

Table 4.1.2: External input data supported in SCALE

Data type	FILETYPE_ORG	Note
NetCDF format	NetCDF	NetCDF file format.
Binary format	GrADS	Another namelist for data input is required.

The program `scale-rm_init` converts external data into boundary and initial data by using the configuration file `init.conf`. It can treat various types of external data as shown in Table 4.1.2. The format of input data is specified by (FILETYPE_ORG) in [PARAM_MKINIT_REAL_(ATMOS|OCEAN|LAND)].

The "NetCDF format" is mainly used for offline nesting experiments (Sec. 4.2.8.2). Other netCDF formatted data, such as WRF data, is also available.

The "Binary format" is defined as binary format with single-precision floating point values that FORTRAN can directly access. The example of practical usage of binary format data is described in Sec. 3.2.2.

Other format data, such as GRIB/GRIB2 data, can be read in SCALE by converting the data to the binary format data. Note that the format of output files in the latest version is different from that in the version 5.3 or older. Therefore, the init/boundary files which are made in the version 5.3 or older can't be used in the current version (SCALE 5.5.1).

Settings common to NetCDF format and Binary format

The base name of the initial file to be output is set by (RESTART_OUT_BASENAME) in [PARAM_RESTART] as follows:

```
&PARAM_RESTART
 RESTART_OUTPUT = .true., ; whether initial (or restart) files is output.
 RESTART_OUT_BASENAME = "init", ; base name of initial (or restart) files
/
```

To generate initial file, (RESTART_OUTPUT) is set to .true.. The base name of initial file is specified by (RESTART_OUT_BASENAME). For example, in the tutorial described in Sec. 3.2.1, (RESTART_OUT_BASENAME) = "init_d01". These settings are used when you want to output restart files in executing SCALE-RM (see Sec. 4.2.7 for details). The structure of generated restart files is same as that of initial files.

The settings for input data and boundary files are specified by [PARAM_MKINIT_REAL_(ATMOS|OCEAN|LAND)] in the file `init.conf`.

```
&PARAM_MKINIT_REAL_ATMOS
  NUMBER_OF_FILES = 1,                       ; Number of input files
  NUMBER_OF_TSTEPS = 1,                      ; Number of time step for each input files for Binary
                                             format
  FILETYPE_ORG = '',                         ; Select from Table 4.1.2
  BASENAME_ORG = '',                         ; Information about input files
                                                 (the specified value depends on FILETYPE_ORG)
  BASENAME_ADD_NUM = .false.,                ;  Switch  to  number  the  name  of  file  when
                                             NUMBER_OF_FILES=1
  BASENAME_BOUNDARY = '',                    ; Base name of boundary files
  BOUNDARY_UPDATE_DT = 0.0,                  ; Time interval for input data [s]
  USE_FILE_DENSITY = .false.,                ; Switch to use data of density stored in input files
  PT_dry = .true.,                           ; If the potential temperature is defined associated
                                             with dry air
  USE_NONHYDRO_DENS_BOUNDARY = .false.,      ; Switch to use nonhydrostatic density as boundary
                                             data
  USE_SFC_DIAGNOSES = .false.,               ; Switch to use diagnostic variables at surface in par-
                                             ent model
  USE_DATA_UNDER_SFC = .true.,               ; Switch to use values below the surface in parent
                                             model
  SAME_MP_TYPE = .false.,                    ;  (For SCALE format) Whether the microphysics
                                             scheme is same as that used in parent model
  UPPER_QV_TYPE = 'ZERO',                    ; How to fill the QV value when it is missing at upper
                                             layers ("ZERO" , "COPY")
  INTRP_TYPE = 'LINEAR',                     ;  Type  of  horizontal  interpolation  ("LINEAR",
                                             "DIST-WEIGHT")
  SERIAL_PROC_READ = .true.,                 ; Whether only the master process read input data?
  SKIP_VERTICAL_RANGE_CHECK = .false.,       ; Turn it to true when the top layer-height of the file
                                             data is lower than the model level
/
&PARAM_MKINIT_REAL_OCEAN
  NUMBER_OF_FILES = 1,                       ; Number of input files
  NUMBER_OF_TSTEPS = 1,                      ; Number of time steps for each input files for Binary
                                             format
  FILETYPE_ORG = '',                         ; Select from Table 4.1.2
  BASENAME_ORG = '',                         ; Information about input file
                                                 (the specified value depends on FILETYPE_ORG)
  INTRP_OCEAN_SFC_TEMP = 'off',              ; (For GrADS format) ("off", "mask", "fill")
  INTRP_OCEAN_TEMP = 'off',                  ; (For GrADS format) ("off", "mask", "fill")
  SERIAL_PROC_READ = .true.,                 ; Whether only the master process read input data?
/
&PARAM_MKINIT_REAL_LAND
  NUMBER_OF_FILES = 1,                       ; Number of input files
  NUMBER_OF_TSTEPS = 1,                      ; Number of time steps for each input files for Binary
                                             format
  FILETYPE_ORG = '',                         ; Select from Table 4.1.2
  BASENAME_ORG = '',                         ; Information about input file
                                                 (the specified value depends on FILETYPE_ORG)
  USE_FILE_LANDWATER = .true.,               ; Switch to use data of soil moisture stored in input
                                             file
  INTRP_LAND_TEMP = 'off',                   ; (For GrADS format) ("off", "mask", "fill")
  INTRP_LAND_WATER = 'off',                  ; (For GrADS format) ("off", "mask", "fill")
  INTRP_LAND_SFC_TEMP = 'off',               ; (For GrADS format) ("off", "mask", "fill")
  ELEVATION_CORRECTION = .true.,             ; Switch to correct the difference of topography eleva-
                                             tion from parent model
  SERIAL_PROC_READ = .true.,                 ; Whether only the master process read input data?
/
```

(NUMBER_OF_FILES) is the number of input files. The program scale-rm_init reads these files

enumerated from 00000 to the given number (NUMBER_OF_FILES)-1. For the case of (NUMBER_OF_FILES)=1 where the enumeration is not automatically performed, you should set (BASENAME_ADD_NUM)=.true. to enumerate the files. (NUMBER_OF_TSTEPS) is the number of time steps stored in each input file. It is ignored when (FILETYPE_ORG) is "NetCDF".

(BOUNDARY_UPDATE_DT) is the time interval of input data; the time interval of boundary data is the same as that of the input data. (BASENAME_BOUNDARY) is the base name of the output boundary files. If (BASENAME_BOUNDARY) is not specified, no boundary files are output. For example, in the above case, the boundary file(s) are created only for the atmospheric variables; the initial file is prepared for all atmospheric, ocean, and land variables. In order to perform model integration, the boundary data for atmospheric variables must have two time steps at least. On the other hands, whether boundary data for variables of ocean and land is necessary depends on schemes employed in executing models.

(INTRP_TYPE) is the type of horizontal interpolation; "LINEAR" and "DIST-WEIGHT" are available. If "LINEAR" is selected, the bi-linear interpolation is used. This type can not be used when the total number of horizontal gird cells in either the x- or y-direction is 1, i.e., (IMAXG) = 1 or (JMAXG) = 1, such as a case of 2-dimensional experiment or an unstructured grid data which is stored in one-dimensional array in the horizontal direction. If "DIST-WEIGHT" is selected, distance-weighted mean of the nearest N-neighbors is used. In the case of distance-weighted mean, the number of the neighbors is set by (COMM_CARTESC_NEST_INTERP_LEVEL) in [PARAM_COMM_CARTESC_NEST].

(upper_qv_type) is the type how to fill the missing value in qv data at the upper layers. For (upper_qv_type)='ZERO', the value of QV is set to zero. For (upper_qv_type)='COPY', the value at the highest layer where input data exists is copied. The default value of upper_qv_type is "ZERO".

In SCALE-RM , as default, only the master process reads input data, and then broadcasts the data to each node. At this time, the algorithm may cause insufficient memory error when reading large input data, especially on a high-performance computing system. By setting (SERIAL_PROC_READ) to .false., each node accesses by itself to only the data it needs, resulting in the reduction of memory usage. Note that, the computational performance may deteriorate due to locking file access because the algorithm places a load on file I/O.

The above configurations except for (BASENAME_BOUNDARY) can be shared between ATMOS and OCEAN (or LAND); namely, if the items of namelist are not specified in [PARAM_MKINIT_REAL_(OCEAN|LAND)], the values same as that in [PARAM_MKINIT_REAL_ATMOS] are used.

Settings for PARAM_MKINIT_REAL_ATMOS

The method to calculate density is specified by (USE_FILE_DENSITY) and (USE_NONHYDRO_DENS_BOUNDARY). For the default setting where both of values are .false., the density in initial and boundary data is calculated with hydrostatic balance, i.e., $\frac{dp}{dz} = -\rho g$ from the input data of temperature and humidity. If (USE_FILE_DENSITY) = .true., the density read from the input file(s) is used as well as the other variables. Regardless of the (USE_FILE_DENS), if (USE_NONHYDRO_DENS_BOUNDARY) = .true., the density of boundary data is calculated using the equation of state, i.e., $\rho = p/RT$ and the input temperature, pressure, and humidity data (the initial data is not affected). This density is generally consistent with that in the parent model. The reason why the option for density boundary data is provided is the following. In most cases, the density satisfying the hydrostatic balance is preferred for the initial data to reduce initial shock in the simulation. Therefore, (USE_FILE_DENSITY)=.false. is recommended. On the other hand, since the density constructed with the hydrostatic balance could differ from that in the parent model (or realistic value), this may cause a significant mass bias in the simulation. In such cases, in terms of representation of the simulation such as pressure field, it could be better to use consistent density with the parent's one using (USE_NONHYDRO_DENS_BOUNDARY)=.true.. The vertical acceleration or waves due to the use of hydrostatically inbalanced density as the boundary data is expected to be quickly dumped by the lateral nudging.

(USE_SFC_DIAGNOSES) is the switch of method to calculate the values at layers on SCALE below the lowermost layer of input data. If (USE_SFC_DIAGNOSES) = .true., the surface quantities, such as T2, RH2, U10, V10, and PSFC are used. Otherwise, constant potential temperature and hydrostatic balance are assumed. (USE_DATA_UNDER_SFC) is switch whether the input data below

its surface is used or ignored.

Settings for `PARAM_MKINIT_REAL_LAND`

The soil water is configured by `(USE_FILE_LANDWATER)` of `[PARAM_MKINIT_REAL_LAND]` in `init.conf`. There are two options in preparation of soil moisture: i) the data provided from the input file (`(USE_FILE_LANDWATER) = .true.`) and ii) a constant value in the entire region (`(USE_FILE_LANDWATER) = .false.`). In the case of the option i), either fraction of volume of soil moisture (`SMOISVC`) or saturation ratio (`SMOISDS`) is required as the 3D soil moisture data. Here, the fraction of volume of soil moisture is the ratio of water volume (V_w) to soil volume (V), i.e., V_w/V, and the saturation ratio is the ratio of water volume (V_w) to void spaces in V (V_v), i.e., V_w/V_v. In the case of the option ii), the saturation ratio is specified by `(INIT_LANDWATER_RATIO)` whose default value is 0.5. The porosity of the soil (V_v/V) depends on the land use.

```
&PARAM_MKINIT_REAL_LAND
  USE_FILE_LANDWATER = .false.    ; Switch to give soil moisture by input files.  The default is .true.
  INIT_LANDWATER_RATIO = 0.5      ; (For USE_FILE_LANDWATER=.false.) saturation ratio
  ..........
/
```

`(ELEVATION_CORRECTION)` in `[PARAM_MKINIT_REAL_LAND]` determines whether soil temperature and land surface temperature for the initial and boundary data are corrected according to the topographical difference between the parent model and the SCALE-RM . In general, topographies of the parent model and the SCALE-RM are different. Therefore, if soil temperature and land surface temperature for the initial and boundary data are made by simply interpolating the data of the parent model, there could be inconsistency of temperature according to the height difference of the topography. If `(ELEVATION_CORRECTION)` is `.true.`, soil temperature and land surface temperature for the initial and boundary data are corrected according to the height difference. For example, if the topography made by the SCALE-RM is Δh higher than that of the parent model, soil temperature and land surface temperature are uniformly reduced by $\Delta h \Gamma$, where Γ is lapse rate of the international standard atmosphere (ISA): $\Gamma = 6.5 \times 10^{-3}$ [K/m]. The default value is `(ELEVATION_CORRECTION) = .true.`.

Input from NetCDF format data

An example of configurations in `[PARAM_MKINIT_REAL_(ATMOS|OCEAN|LAND)]` for the NetCDF format data is as follows:

```
&PARAM_MKINIT_REAL_ATMOS
  NUMBER_OF_FILES = 2,
  FILETYPE_ORG = "NetCDF",
  BASENAME_ORG = "history_d01",
  BASENAME_ADD_NUM = .true.,
  BASENAME_BOUNDARY = 'boundary_d01',
  SAME_MP_TYPE = .false.,
…
/
&PARAM_MKINIT_REAL_OCEAN
  FILETYPE_ORG = "NetCDF",
  BASENAME_ORG = "history_d01",
…
/
&PARAM_MKINIT_REAL_LAND
  FILETYPE_ORG = "NetCDF",
  BASENAME_ORG = "history_d01",
…
/
```

(FILETYPE_ORG) is set to "NetCDF". The base name of input files is specified by (BASENAME_ORG). Given BASENAME_ORG="history_d01", the input file should be named history_d01.peYYYYYY.nc or history_d01.nc when the number of input file is 1. When the number of input files is larger than 1 or (BASENAME_ADD_NUM) = .true. in [PARAM_RESTART], the name of the input files should be numbered as "history_d01_XXXXX.peYYYYYY.nc" or "history_d01_XXXXX.nc".

If cloud microphysical scheme used is same as that in the parent model, specify (SAME_MP_TYPE).

For the output files of SCALE or WRF, necessary information is automatically determined internally. For other files, namelist providing such information is required.

In the [PARAM_MKINIT_REAL_(ATMOS|LAND|OCEAN)_NetCDF], file type is specified by (FILE_TYPE). It should be "SCALE-RM", "WRFARW", "NAMELIST", or "AUTO". If it is "AUTO", the type is detected automatically. The namelist file is specified by (NM_FILE) of [PARAM_MKINIT_REAL_(ATMOS|LAND|OCEAN)_NetCDF] as follows:

```
&PARAM_MKINIT_REAL_ATMOS_NetCDF
 FILE_TYPE = "NAMELIST", ; "SCALE-RM", "WRFARW", "NAMELIST" or "AUTO"
 NM_FILE = "namelist_netcdf.nml", ; namelist file name
 MIXING_RATIO = .false.,
/
&PARAM_MKINIT_REAL_LAND_NetCDF
 FILE_TYPE = "NAMELIST",
 NM_FILE = "namelist_netcdf.nml",
/
&PARAM_MKINIT_REAL_OCEAN_NetCDF
 FILE_TYPE = "NAMELIST",
 NM_FILE = "namelist_netcdf.nml",
/
```

Definition of the magnitude of the water contents in the file is specified by (MIXING_RATION). It is ".true." when it is mixing ratio, that is ratio to the dry air mass, otherwise ratio to the total air mass.

In the namelist file, the information is specified as follows:

```
&NetCDF_MAPPROJECTION ; settings for map projecton
 mapping_name = "lambert_conformal_conic",
 longitude_of_central_meridian = 135.0,
 longitude_of_projection_origin = 135.0,
 latitude_of_projection_origin = 34.5,
 standard_parallel = 30.0, 40.0,
 rotation = 0.0D0
/
&NetCDF_DIMS
 zname = "z", ; name of the coordinate in the z dimension
 zhname = "zh", ; name of the coordinate in the z dimension for the staggered grid
 xname = "x", ; name of the coordinate in the x dimension
 xhname = "xh", ; name of the coordinate in the x dimension for the staggered grid
 yname = "y", ; name of the coordinate in the y dimension
 yhname = "yh", ; name of the coordinate in the y dimension for the staged grid
 tname = "time", ; name of the coordinate in the time dimension
/
&NetCDF_ITEM item="Umet", name="u", xstg=.true. /
&NetCDF_ITEM item="Vmet", name="v", ystg=.true. /
&NetCDF_ITEM item="T", name="t", offset=273.0 /
...
```

The correspondence of the names of coordinates are specified by the `[NetCDF_DIMS]`. The names for the staggered grid is required only for atmosphere.

The correspondence of variable names are specified with the `[NetCDF_ITEM]`. The explanations of `[NetCDF_ITEM]` is described in Table **??**. The list of the `(item)` is in the table 4.1.5. Note that other item is available for tracer variables which depending on experimental configuration.

Table 4.1.3: Variables of `[NetCDF_ITEM]`

item of `NetCDF_ITEM`	Explanation	Note
item	Variable name	Select from Table 4.1.5
name	Variable name in the netCDF file	
zstg	If the variable is at the staggered point in the z-direction	(optional)
xstg	If the variable is at the staggered point in the x-direction	(optional)
ystg	If the variable is at the staggered point in the y-direction	(optional)
offset	Offset value	(optional)
factor	Factor value	(optional)

Input from binary format data

If the binary data is used as input files, you need to prepare data according to the format used in GrADS . For the details, see http://cola.gmu.edu/grads/gadoc/aboutgriddeddata.html#structure .

An example of configurations in `[PARAM_MKINIT_REAL_(ATMOS|OCEAN|LAND)]` for the binary format data is as follows:

```
&PARAM_MKINIT_REAL_ATMOS
 NUMBER_OF_FILES = 2,
 FILETYPE_ORG = "GrADS",
 BASENAME_ORG = "namelist.grads_boundary.FNL.2005053112-2015011400",
 BASENAME_ADD_NUM = .true.,
 BASENAME_BOUNDARY = 'boundary_d01',
 BOUNDARY_UPDATE_DT = 21600.0,
...
/
&PARAM_MKINIT_REAL_OCEAN
 FILETYPE_ORG = "GrADS",
 BASENAME_ORG = "namelist.grads_boundary.FNL.2005053112-2015011400",
 INTRP_OCEAN_SFC_TEMP = "mask",
 INTRP_OCEAN_TEMP = "mask",
...
/
&PARAM_MKINIT_REAL_LAND
 FILETYPE_ORG = "GrADS",
 BASENAME_ORG = "namelist.grads_boundary.FNL.2005053112-2015011400",
 INTRP_LAND_TEMP = "fill",
 INTRP_LAND_WATER = "fill",
 INTRP_LAND_SFC_TEMP = "fill",
...
/
```

`(FILETYPE_ORG)` is set to `"GrADS"`. In SCALE-RM , the file name and the data structure of the GrADS formatted binary data are specified in the namelist file specified by `(BASENAME_ORG)` instead of the "ctl" file. The namelist file needs to be prepared beforehand by users.

The following is an example of the namelist file `namelist.grads_boundary*` which provides information of the data file name and data structure.

```
#
# Dimension
#
&GrADS_DIMS
 nx = 360,    ; Default value of the number of grids in x direction
 ny = 181,    ; Default value of the number of grids in y direction
 nz = 26,     ; Default value of the number of layers in z direction
/


#
# Variables
#
&GrADS_ITEM name='lon', dtype='linear', swpoint=0.0d0, dd=1.0d0 /
&GrADS_ITEM name='lat', dtype='linear', swpoint=90.0d0, dd=-1.0d0 /
&GrADS_ITEM name='pressure', dtype='levels', lnum=26,
   lvars=100000,97500,.........,2000,1000, /
&GrADS_ITEM name='height', dtype='map', fname='FNLatm', startrec=1, totalrec=125 /
&GrADS_ITEM name='U', dtype='map', fname='FNLatm', startrec=27, totalrec=125 /
&GrADS_ITEM name='V', dtype='map', fname='FNLatm', startrec=53, totalrec=125 /
&GrADS_ITEM name='T', dtype='map', fname='FNLatm', startrec=79, totalrec=125 /
&GrADS_ITEM name='RH', dtype='map', fname='FNLatm', startrec=105,totalrec=125, nz=21 /
&GrADS_ITEM name='MSLP', dtype='map', fname='FNLsfc', startrec=1, totalrec=9 /
&GrADS_ITEM name='SFC_PRES', dtype='map', fname='FNLsfc', startrec=2, totalrec=9 /
&GrADS_ITEM name='SFC_TEMP', dtype='map', fname='FNLsfc', startrec=3, totalrec=9 /
&GrADS_ITEM name='topo', dtype='map', fname='FNLsfc', startrec=4, totalrec=9 /
&GrADS_ITEM name='lsmask', dtype='map', fname='FNLsfc', startrec=5, totalrec=9 /
&GrADS_ITEM name='U10', dtype='map', fname='FNLsfc', startrec=6, totalrec=9 /
&GrADS_ITEM name='V10', dtype='map', fname='FNLsfc', startrec=7, totalrec=9 /
&GrADS_ITEM name='T2', dtype='map', fname='FNLsfc', startrec=8, totalrec=9 /
&GrADS_ITEM name='RH2', dtype='map', fname='FNLsfc', startrec=9, totalrec=9 /
&GrADS_ITEM name='llev', dtype='levels', lnum=4, lvars=0.05,0.25,0.70,1.50, /
      missval=9.999e+20 /
&GrADS_ITEM name='LAND_TEMP', dtype='map', fname='FNLland', nz=4, startrec=1, totalrec=8,
      missval=9.999e+20 /
&GrADS_ITEM name='LAND_WATER', dtype='map', fname='FNLland', nz=4, startrec=5, totalrec=8,
      missval=9.999e+20 /
```

The default value of the number of grids is specified as `nx, ny, nz` in [GrADS_DIMS]. Configurations for the variables are individually specified by [GrADS_ITEM]. The explanations of [GrADS_ITEM] is described in Table 4.1.6.

The base name of input files is set as `fname` in the [GrADS_ITEM]. Given the base name is set as `fname="filename"`, the input binary data file should be named `filename.grd` in the case that the number of the input binary file is one ((NUMBER_OF_FILES) = 1). In the case that there is more than one input files or (BASENAME_ADD_NUM) = .true. in [PARAM_RESTART], prepare the files numbered as `"filename_XXXXX.grd"`.

If the number of grids for a variable is different from the default value, the value is specified by `nx, ny, nz` of [GrADS_ITEM]. For example, specific humidity (`QV`) and relative humidity (`RH`) in upper levels are not always available. In such case, the number of layers where the data exist is specified as `nz`.

The variables required for the calculation with SCALE-RM are listed in Table 4.1.7.

Table 4.1.4: Item names for (item) in [NetCDF_ITEM]

Item name (item)	Explanation	Note
lon	longitude	
lat	latitude	
height	height	
pressure	pressure	
DENS	air density	
W	vertical wind speed	
MOMZ	vertical moment	
Umet	eastward wind speed	
U	wind speed in the x-direction	
MOMX	moment in x-direction	
Vmet	northward wind speed	
V	wind speed in the y-direction	
MOMY	moment in y-direction	
PT	potential temperature	c.f. (PT_dry) in [PARAM_MKINIT_REAL_ATMOS]
RHOT	density weighted potential temperature	c.f. (PT_dry) in [PARAM_MKINIT_REAL_ATMOS]
T	temperature	
QV	vapor mass	c.f. (mixing_ratio) in [PARAM_MKINIT_REAL_ATMOS_NETCDF]
RH	relative humidity	
QC	ratio of cloud water mass	c.f. (mixing_ratio) in [PARAM_MKINIT_REAL_ATMOS_NETCDF]
QR	ratio of rain water mass	c.f. (mixing_ratio) in [PARAM_MKINIT_REAL_ATMOS_NETCDF]
QI	ratio of cloud ice mass	c.f. (mixing_ratio) in [PARAM_MKINIT_REAL_ATMOS_NETCDF]
QS	ratio of snow mass	c.f. (mixing_ratio) in [PARAM_MKINIT_REAL_ATMOS_NETCDF]
QG	ratio of graupel mass	c.f. (mixing_ratio) in [PARAM_MKINIT_REAL_ATMOS_NETCDF]
MSLP	mean sea level pressure	
SFC_PRES	surface pressure	
Umet10	eastward 10m wind speed	
U10	10m wind speed in the x-direction	
Vmet10	northward 10m wind speed	
V10	10m wind speed in the y-direction	
T2	2m temperatuer	
Q2	2m vapor mass	
RH2	2m relative humidity	

Table 4.1.5: Item names for (item) in [NetCDF_ITEM], following the previous page.

Item name (item)	Explanation	Note
lz	soil depth	
topo	surface elevation	
lsmask	land mask	
LAND_TEMP	soil temperature	
LANS_WATER	soil water	
LAND_SFC_TEMP	land surface temperature	
LAND_SFC_ALB_IR_dir	land albedo for IR (direct)	
LAND_SFC_ALB_IR_dif	land albedo for IR (diffused)	
LAND_SFC_ALB_NIR_dir	land albedo for NIR (direct)	
LAND_SFC_ALB_NIR_dif	land albedo for NIR (diffued)	
LAND_SFC_ALB_VIS_dir	land albedo for VIS (direct)	
LAND_SFC_ALB_VIS_dif	land albedo for VIS (diffued)	
LAND_SFC_EMIS_IR_dif	land emissivity for IR (diffued)	
URBAN_SFC_TEMP	urban surface temperature	
OCEAN_TEMP	sea temperature	
OCEAN_SFC_TEMP	sea surface temperature	
OCEAN_SFC_ZOM	sea surface roughness	
OCEAN_SFC_ALB_IR_dir	ocean albedo for IR (direct)	
OCEAN_SFC_ALB_IR_dif	ocean albedo for IR (diffused)	
OCEAN_SFC_ALB_NIR_dir	ocean albedo for NIR (direct)	
OCEAN_SFC_ALB_NIR_dif	ocean albedo for NIR (diffued)	
OCEAN_SFC_ALB_VIS_dir	ocean albedo for VIS (direct)	
OCEAN_SFC_ALB_VIS_dif	ocean albedo for VIS (diffued)	
OCEAN_SFC_EMIS_IR_dif	ocean emissivity for IR (diffued)	

Table 4.1.6: Variables of [GrADS_ITEM]

item of GrADS_ITEM	Explanation	Note
name	Variable name	Select from Table 4.1.7
dtype	Data type	"linear","levels" or "map"
namelist at (dtype)="linear" (specific use of "lon", "lat")		
fname	Header name of files	
swpoint	Value of start point	
dd	Increment	
namelist at (dtype)"=levels" (specific use of "plev", "llev")		
lnum	Number of levels (layers)	
lvars	Values of each layer	
namelist at (dtype)="map"		
startrec	Recorded number of variables (item)	time at t=1
totalrec	Recorded length of all variables per time	
missval	Missing value	(option)
nx	Number of grids in x direction	(option)
ny	Number of grids in y direction	(option)
nz	Number of layers in z direction	(option)
yrev	If the data is stored from the north to south, set .true.	(option)

Table 4.1.7: Variable list of (name) in [GrADS_ITEM]. The asterisk means "it is optional but recommended as possible". The double-asterisk means "it is available but not recommended".

	Variable name (name)	Explanation	Unit	Data type (dtype)
	lon	longitude data	[deg.]	linear, map
	lat	latitude data	[deg.]	linear, map
	pressure, plev	pressure data	[Pa]	levels, map
	height, HGT	geopotential height data	[m]	map
*	DENS	air density	[kg/m3]	map
	U	eastward wind speed	[m/s]	map
	V	northward wind speed	[m/s]	map
**	W	vertical wind speed	[m/s]	map
	T	temperature	[K]	map
	RH	relative humidity (optional if QV is given)	[%]	map
	QV	specific humidity (optional if RH is given)	[kg/kg]	map
**	QC	ratio of cloud water mass	[kg/kg]	map
**	QR	ratio of rain water mass	[kg/kg]	map
**	QI	ratio of cloud ice mass	[kg/kg]	map
**	QS	ratio of snow mass	[kg/kg]	map
**	QG	ratio of graupel mass	[kg/kg]	map
**	MSLP	sea level pressure	[Pa]	map
**	SFC_PRES, PSFC	surface pressure	[Pa]	map
**	U10	eastward 10m wind speed	[m/s]	map
**	V10	northward 10m wind speed	[m/s]	map
**	T2	2m temperature	[K]	map
**	RH2	2m relative humidity (optional if Q2 is given)	[%]	map
**	Q2	2m specific humidity (optional if RH2 is given)	[kg/kg]	map
*	TOPO	topography of GCM	[m]	map
*	lsmask	ocean–land distribution of GCM	0:ocean,1:land	map
	LAND_SFC_TEMP, SFC_TEMP, SKINT	surface temperature	[K]	map
	llev	soil depth	[m]	levels
	LAND_TEMP, STEMP	soil temperature	[K]	map
	LAND_WATER, SMOISVC	soil moisture (volume fraction) (optional if SMOISDS is given)	[-]	map
	SMOISDS	soil moisture (saturation ratio) (optional if SMOISVC is given)	[-]	map
	OCEAN_SFC_TEMP, SST	sea surface temperature (optional if SKINT is given)	[K]	map

4.1.3　How to prepare input data from user-defined data

Some schemes requires other input data than initial and boundary data. In order to use these schemes, the data are necessary be prepared before the simulation. The input data must be SCALE-netCDF file and have the grid information according to the experimental settings. To prepare the input data, SCALE-RM provide a system to convert user-defined data to a SCALE-netCDF file. When (CNVERT_USER) = .true. in [PARAM_CONVERT], the data conversion can be executed by `scale-rm_init`. The supported formats of the user-defined data are shown in Table 4.1.8. Since only one variable can be converted in a single execution, the execution must be repeated as for the number of the user-defined variables to be converted. At this moment, the conversion is available only for spatially two dimensional data.

Table 4.1.8: Types of user-defined data supported in SCALE

Data type	Description
GrADS	GrADS binary data.
TILE	horizontally decomposed tiled binary data.

The configurations for the user-defined data are specified in [PARAM_CNVUSER] as follows:

```
&PARAM_CNVUSER
  CNVUSER_FILE_TYPE = '',              ; Type of the user-defined data.  See Table
                                         4.1.8.
  CNVUSER_INTERP_TYPE = 'LINEAR',      ; Type for the horizontal interpolation.
  CNVUSER_INTERP_LEVEL = 5,            ; Level of the interpolation.
  CNVUSER_OUT_TITLE = 'SCALE-RM 2D Boundary',  ; Title in the output file.
  CNVUSER_OUT_VARNAME = '',            ; Variable name in the output file.
  CNVUSER_OUT_VARDESC = '',            ; Description of the output variable.
  CNVUSER_OUT_VARUNIT = '',            ; Unit of the output variable.
  CNVUSER_OUT_DTYPE 'DEFAULT',         ; Data type of the output variable (DE-
                                         FAULT,INT2,INT4,REAL4,REAL8).
  CNVUSER_NSTEPS = 10,                 ; The number of time steps in the data.
  CNVUSER_OUT_DT = -1.0D0,             ; Time interval in second of the output vari-
                                         able.
  CNVUSER_GrADS_FILENAME = '',         ; Name of the namelist for the GrADS data.
  CNVUSER_GrADS_VARNAME = '',          ; Name of the target variable in the namelist.
  CNVUSER_GrADS_LATNAME = 'lat',       ; Name of latitude in the namelist.
  CNVUSER_GrADS_LONNAME = 'lon',       ; Name of longitude in the namelist.
  CNVUSER_TILE_DIR = '',               ; Directory name where the catalogue and tiled
                                         data files exist.
  CNVUSER_TILE_CATALOGUE = '',         ; Name of the catalogue file.
  CNVUSER_TILE_DLAT = 1.0D0,           ; Latitude interval in the tiled data.
  CNVUSER_TILE_DLON = 1.0D0,           ; Longitude interval in the tiled data.
  CNVUSER_TILE_DTYPE = 'REAL4',        ; Data type of the tiled data
                                         (INT2,INT4,REAL4,REAL8).
/
```

The type and level for the horizontal interpolation is specified by (CNVUSER_INTERP_TYPE) and (CNVUSER_INTERP_LEVEL), respectively. See Section 4.1.2 for the details for the interpolation.

(CNVUSE_OUT_TITLE), (CNVUSE_OUT_VARNAME), (CNVUSE_OUT_VARDESC), and (CNVUSE_OUT_VARUNIT) are settings for the title, variable name, variable description, and variable unit in the output file, respectively. When the data type is GrADS and the (CNVUSE_OUT_VARNAME) is not set, (CNVUSER_GrADS_VARNAME) is used as the output variable name.

If (CNVUSER_OUT_DTYPE) is DEFAULT, it is the same as the data type used in the simulation; it is REAL8 as default, and REAL4 when SCALE-RM is compiled with the environment variable SCALE_USE_SINGLEFP = "T" (See Section 2.2.2).

The number of time steps contained in the user-defined data is specified by (CNVUSER_NSTEPS). The initial time of the time coordinate in the output file is specified by (TIME_STARTDATE) in [PARAM_TIME] (See Section 4.2.5), and the time interval in second is specified by (CNVUSER_OUT_DT).

For the GrADS type, another namelist file which contains structure of input file is necessary. (USERFILE_GrADS_FILENAME) is used to set the namelist file. See Section 4.1.2 for the details of the namelist file. As the default, the variable name for latitude, and longitude are lat, and lon, respectively. If the name is different from the default value, it can be specified with (CNVUSER_GrADS_LATNAME), and (CNVUSER_GrADS_LONNAME), respectively. Two samples to convert GrADS-type use-defined data are provided at scale/scale-rm/test/framework/cnvuser The following is an example of the namelist file.

```
#
# Dimension
#
&GrADS_DIMS
 nx = 361,
 ny = 181,
 nz = 1,
/
#
# Variables
#
&GrADS_ITEM name='lon', dtype='linear', swpoint=0.0D0, dd=1.0D0 /
&GrADS_ITEM name='lat', dtype='linear', swpoint=-90.0D0, dd=1.0D0 /
&GrADS_ITEM name='var', dtype='map', fname='fname_in', startrec=1, totalrec=1, \
         bintype='int1', yrev=.true. /
```

The TILE-type data means the horizontally decomposed tiled data. The individual tiled data should a direct access binary file. The binary data type is specified by (CNVUSER_TILE_DTYPE). The data must be structured on a latitude-longitude grid with equal intervals, specified by (CNVUSER_TILE_DLAT) and (CNVUSER_TILE_DLON) for latitude and longitude, respectively. A catalogue file is required; the catalogue contains information of the name of individual tiled data files and the area which the tiled data covers. The name of the catalogue file is specified by (CNVUSER_TILE_CATALOGUE). The catalogue and tiled data files are located at the directory specified by (CNVUSER_TILE_DIR). Each line of the catalogue file describes the serial number, the southernmost and the northernmost latitudes, the westernmost and the easternmost longitudes, and the name of the tiled data. The following is an example of the catalogue file for the global data decomposed into total 4 (2 x 2) files.

```
001 -90.0 0.0 -180.0 0.0 TILE_sw.grd    ; latitude: -90–0, longitude: -180–0, name: TILE_sw.grd
002 -90.0 0.0 0.0 180.0 TILE_se.grd     ; latitude: -90–0, longitude: 0–180, name: TILE_se.grd
003 0.0 90.0 -180.0 0.0 TILE_nw.grd     ; latitude: 0–90, longitude: -180–0, name: TILE_nw.grd
004 0.0 00.0 0.0 180.0 TILE_ne.grd      ; latitude: 0–90, longitude: 0–180, name: TILE_ne.grd
```

Chapter 4.2

Framework for SCALE-RM

4.2.1 Supporting tool for preparation of configuration files

The configuration files ***.conf for experiment are required to be individually prepared for pp, init, run. Some items in namelist are required to be the same among these configuration files. If there is an inconsistency among them, the model does not run properly. To avoid such mistakes, a convenient supporting tool for the preparation of configuration files is provided as follows:

```
$ cd ${Tutorial_DIR}/real/
$ ls
   Makefile
     : Makefile for generation of necessary files for experiment
   README
     : README file
   USER.sh
     : Shell script for specifying the settings
   config/
     : Configuration files for each setting (not necessary for user to rewrite it)
   sample
     : Sample scripts of USER.sh
   data
     : Files for tutorial real
   tools
     : Files for tutorial real, converting FNL data from grib format to binary format
```

Although this tool is intended for the tutorial experiment of real atmosphere as initial setting, users can change the settings in USER.sh.

Under the directory sample/, several sample scripts are provided for the typical configuration. Use them as necessary by copying them to USER.sh.

```
$ ls sample/
  USER.default.sh
     : USER.sh same as for the tutorial experiment of real atmosphere
     ( single domain )
  USER.offline-nesting-child.sh
     : for the child domain for off-line nesting experiment
  USER.offline-nesting-parent.sh
     : for the parent domain for off-line nesting experiment
  USER.online-nesting.sh
     : for the on-line nesting experiment
```

How to use the tool

The use of this tool is described as follows (or refer to README):

1. Edit `USER.sh` according to experimental settings intended by the users.

2. Execute `make` command.

All the necessary files for the experiment are generated under the directory `experiment`.
Since the configuration in `USER.sh` is that for `tutorial` of real atmosphere, it is recommended to leave the tutorial setting as a different file as follows:

```
$ mv experiment/ tutorial/
       : (In the case of already existing experiment directory)
$ cp USER.sh USER_tutorial.sh
... edit USER.sh ...
$ make
$ cp -rL experiment/ arbitrary_directory/
       : ``arbitrary_place'' means  directory name of arbitrary place
```

Editing `USER.sh`

In the beginning part of script file, there is `NUM_DOMAIN` for specifying the number of domains. Then, there are some items specified to configuration files generated by the script, and so change the items appropriately. When an item is followed by `"# require parameters for each domain"`, values of the item must be described in proportion to the number of domains and separated by space. Note that if the number of item's values and the number of nesting domains according to `NUM_DOMAIN` are different, the set of files for the experiment cannot be generated. When items do not appear in `USER.sh`, edit directly the configuration file generated under the directory `experiment`.

4.2.2   Configuration File Converter from Previous Version

From version 5.4 to 5.5, there are some changes in namelist parameters. So, we provide the configuration file converter from 5.4 to 5.5. It requires "ruby" (https://www.ruby-lang.org/en/).
Usage of the converter is the following:

```
$ ruby scale-5.5.1/utils/config-converter/config-converter_5.4-5.5.rb \\
       old.conf > new.conf
```

4.2.3 Setting the target domain

In this section, the number of grids, and the target domain and its relationship with the MPI process are explained. The calculation domain is determined by the horizontal grid spacing and the number of grid points. The parallelization is implemented by a 2D domain decomposition in the horizontal directions.

Figure 4.2.1 shows an example of this relationship. The total numbers of grids are specified by (IMAXG, JMAXG) in [PARAM_ATMOS_GRID_CARTESC_INDEX]. The entire domain is divided into (PRC_NUM_X) along the X direction and (PRC_NUM_Y) in the Y direction; the (PRC_NUM_X) and (PRC_NUM_Y) are specified in [PARAM_PRC_CARTESC]. The process number starts zero and is numbered in order from the bottom left to the top right (Arrow in Fig. 4.2.1). Each subdomain is managed by an MPI process, each of which takes charge of a grid block of (IMAX) × (JMAX) × (KMAX), where (KMAX) is the number of grids along the vertical direction specified in [PARAM_ATMOS_GRID_CARTESC_INDEX]. Care is taken to ensure that (IMAX) and (JMAX) must be identical for all MPI processes. Therefore, (PRC_NUM_X) and (PRC_NUM_Y) must be a divisor of (IMAXG) and (JMAXG), respectively.

The total number of grids along each horizontal direction and that in the entire domain are summarized as

$$\text{IMAXG} = \text{IMAX} \times \text{PRC\_NUM\_X} \tag{4.2.1}$$

$$\text{JMAXG} = \text{JMAX} \times \text{PRC\_NUM\_Y}. \tag{4.2.2}$$

$$\text{Total number of grids in the domain}$$

$$= \text{IMAXG} \times \text{JMAXG} \times \text{KMAX},$$

$$= (\text{IMAX} \times \text{PRC\_NUM\_X}) \times (\text{JMAX} \times \text{PRC\_NUM\_Y}) \times (\text{KMAX}). \tag{4.2.3}$$

When (IMAXG) and (JMAXG) are specified, (IMAX) and (JMAX) are internally calculated. Alternatively, user can specify (IMAX) and (JMAX) instead of (IMAXG) and (JMAXG). In this case, (IMAXG) and (JMAXG) are internally calculated by using Eqs. (4.2.1) and (4.2.2), respectively.

The size of the entire domain is determined as follows:

$$\text{Domain length in the X direction} = \text{IMAXG} \times \text{DX} \tag{4.2.4}$$

$$\text{Domain length in the Y direction} = \text{IMAXG} \times \text{DY}, \tag{4.2.5}$$

where (DX, DY) is grid spacings specified in [PARAM_ATMOS_GRID_CARTESC] as described in subsection 4.2.3.2.

In order to perform a horizontal-vertical two-dimensional experiment, set both the (IMAXG) and (PRC_NUM_X) to 1. In this case, the motion in the Y-Z section is calculated. The value of DX does not affect the simulation result.

In the next subsections, the number of grids, the grid interval, and the configuration of the MPI processes are described in more detail. Note that it is necessary that these settings must be identical among the configuration files for scale-rm_pp, scale-rm_init, and scale-rm.

4.2.3.1 Setting the number of horizontal and vertical grids

The number of grids is specified in [PARAM_ATMOS_GRID_CARTESC_INDEX] in the configuration files.

```
&PARAM_ATMOS_GRID_CARTESC_INDEX
 KMAX = 97,                          ; Number of vertical layers
 IMAXG = 40,                         ; Total number of grids along the X direction
 JMAXG = 25,                         ; Total number of grids along the Y direction
/
```
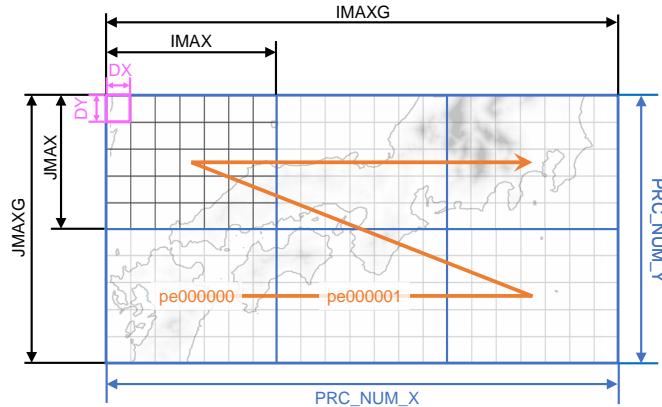
Figure 4.2.1: Relation between horizontal grid interval (`DX`, `DY`), the number of grids (`(IMAX)`, `(JMAX)`) per MPI process, the total number of grids (`(IMAXG)`, `(JMAXG)`) in the entire domain, and the number of MPI processes (`(PRC_NUM_X)`, `(PRC_NUM_Y)`). The blue part corresponds to a region managed by an MPI process. The six digit number following "pe" indicates the process number.

4.2.3.2 Setting grid intervals along the horizontal and vertical directions

Excluding the buffer region explained in Section 4.2.3.5, the horizontal grid intervals are configured only equidistantly, whereas the vertical grid intervals are configured freely. When the grid intervals are configured uniformly along all directions, specify the grid intervals for the x, y, and z directions with `(DX, DY, DZ)` in `[PARAM_ATMOS_GRID_CARTESC]`, respectively. The unit is [m].

```
&PARAM_ATMOS_GRID_CARTESC
 DX = 500.D0,                 ; Grid interval along the X direction
 DY = 500.D0,                 ; Grid interval along the Y direction
 DZ = 500.D0,                 ; Grid interval along the vertical (Z) direction
/
```

An arbitrary non-uniform grid can be specified in the vertical direction. Since the model employs the C-grid system, the points of definition for the velocity vector and other scalars are staggered, deviating by a half grid. In this document, the scalar location is called the center point and the half-grid-deviated location the face point. Refer to Figure 4.2.2 for the details. The face point of the vertical grids can be specified with `(FZ(:))` in `[PARAM_ATMOS_GRID_CARTESC]` as an array [*1]. Note that the number of elements specified in `(FZ(:))` should correspond to the number of vertical layers (`(KMAX)` in `[PARAM_ATMOS_GRID_CARTESC_INDEX]`). The following file for the ideal experiment is shown as an example:

[*1] In this case, the same precision as used in the simulation is recommended to be specified. By default, the model is compiled as a double-precision floating point model.
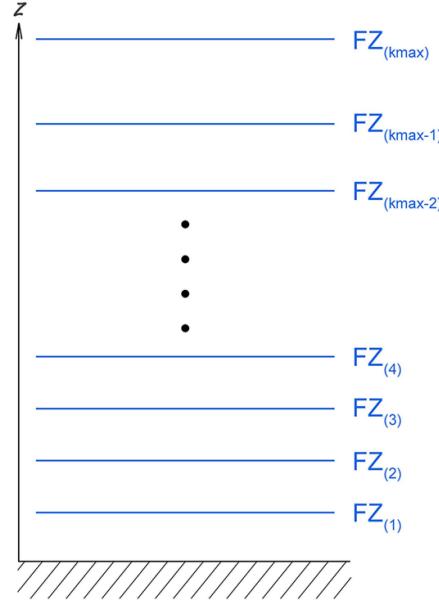
Figure 4.2.2: The definition of the face point in SCALE-RM . If (FZ) is given in [PARAM_ATMOS_GRID_CARTESC], the top height at the first layer is given for the value at $k = 1$. Note that $k = 1$ is not the ground surface height.

```
&PARAM_ATMOS_GRID_CARTESC
  DX = 500.D0,              Grid interval along the X direction (equidistant) [m]
  DY = 500.D0,              Grid interval along the Y direction (equidistant) [m]
  FZ(:) =                   Location at face point along the Z direction [m]
  80.000000000000000D0 ,
  168.00000190734863D0 ,
  264.80000610351567D0 ,
  ........
  14910.428862936289D0 ,
  15517.262523292475D0 ,
  16215.121232702089D0 ,
  17017.658748523147D0 ,
  17940.576891717363D0 ,
  19001.932756390710D0 ,
  20222.492000765058D0 ,
  BUFFER_DZ = 5000.D0,      Refer to Section 4.2.3.5
  BUFFFACT = 1.0D0,         Refer to Section 4.2.3.5
/
```

The above setting is processed at a topographical height of 0 m. The location of the vertical grids at the non-zero topography is appropriately treated by the terrain-following coordinate.

The locations of the vertical grids are configured arbitrarily. However, an unusual configuration sometimes leads to numerical instability. To avoid it, the tool for the generation of vertical grids is supported as a FORTRAN program `make_vgrid.f90` in the directory `scale-5.5.1/scale-rm/util/makevgrid/` with several samples of the namelist. If needed, use them as references. The tool generates the values of (FZ(:)) directly. Copy and paste them in the configuration file.

4.2.3.3 Setting the number of MPI processes

The number of MPI processes is specified in [PARAM_PRC_CARTESC] in the configuration file.

```
&PARAM_PRC_CARTESC
  PRC_NUM_X = 2,        ; number of divisions by MPI parallelization in the X direction (zonal direc-
                          tion)
  PRC_NUM_Y = 1,        ; number of divisions by MPI parallelization in the Y direction (meridional
                          direction)
/
```

It should be noted that the number of MPI process must be a divisor of the total grid number (IMAXG) or (JMAXG) in the x- and y-directions, respectively. Otherwise, the program is terminated immediately with the following message:

```
  number of IMAXG should be divisible by PRC_NUM_X
```

or

```
  number of JMAXG should be divisible by PRC_NUM_Y
```

.

The total number of MPI processes is given by PRC_NUM_X × PRC_NUM_Y. The above example expresses a two-MPI parallel by dividing the domain into two sub-domains along the X direction, but not dividing along the Y direction. The total number of processes must be given as the number of MPI processes in the MPI command at submitting job. If this condition is not satisfied, the program is terminated immediately without calculation and the following message is output to the standard output.

```
  xxx total number of node does not match that requested. Check!
```

Since the input and output files of SCALE-RM are divided process by process according to the MPI, the total number of files is changed according to the number of MPI processes. For example, the initial and boundary files made by two-MPI processes cannot be used for model execution by four-MPI processes. If the number of MPI processes is changed, it is necessary to edit [PARAM_PRC_CARTESC] in pp.conf, init.conf, and run.conf, and then conduct once again the processes of pp and init. As another way, the postprocess SNO is also available for redistribution (Please refer Sec.4.6.1).

4.2.3.4 Setting sponge layer

SCALE-RM adopts height coordinate system in vertical. The uppermost boundary condition is a rigid lid, and the sound and gravity waves often reflect at the model top. To reduce worse effects of these reflecting waves, the damping layer named "sponge layer" is placed in the upper part of the model domain. In the sponge layer, a vertical velocity is dumped by Rayleigh friction. The relaxation time scale (= e-folding time) of damping is minimum at the model top and it increases with decreasing the height. Below the bottom boundary of sponge layer, the relaxation time scale is set to infinity. There are two methods to set the thickness of the sponge layer in [PARAM_ATMOS_DYN].

1. specify number of layer of the sponge layer
   The number of layer appointed in (ATMOS_DYN_wdamp_layer) is set as the sponge layer. The number is counted from the model top.

2. specify bottom boundary height [m] of the sponge layer
   The layer that is higher than altitude appointed in (ATMOS_DYN_wdamp_height) is set as the

sponge layer.

Both parameters above are not set by default, and the sponge layer is not applied. If both are set, `(ATMOS_DYN_wdamp_layer)` is given priority.

The relaxation time at the uppermost boundary is specified by `(ATMOS_DYN_wdamp_tau)`. The unit is [second]. This parameter is not allowed to set the value smaller than `(TIME_DT_ATMOS_DYN)`. When `(ATMOS_DYN_wdamp_tau)` is not specified explicitly, the value ten times as large as `(TIME_DT_ATMOS_DYN)` is automatically set. Please refer to section 4.2.5 for `(TIME_DT_ATMOS_DYN)`. The example of concrete setting is shown in section 4.3.1.2.

4.2.3.5 Setting Buffer Region and Boundary Nudging Method

In general, disagreement in values between input data as boundary condition and actual calculation output occurs at the lateral boundaries. They generate several problems, such as nonphysical mode, in calculation. To avoid these problems, the "buffer region" is placed in the domain.

As shown in Fig.4.2.3, SCALE-RM places the buffer region just inside the calculation domain. In the buffer region, prognostic variables are updated to be close to the specified values of boundary data and/or the parent model data with a certain relaxation time. Hereinafter, this relaxation is called nudging.

Buffer Region

The width of the buffer region is specified in `[PARAM_ATMOS_GRID_CARTESC]` in the configuration file. Note again that the configuration in all procedures must be identical. There are two methods to configure the width of the buffer region.

1. specify number of grid of the buffer region with `(BUFFER_NX, BUFFER_NY, BUFFER_NZ)`

2. specify width [m] of the buffer region with `(BUFFER_DX, BUFFER_DY, BUFFER_DZ)`

Both parameters above are not set by default, and no buffer regions are set. If both are set, `(BUFFER_NX, BUFFER_NY, BUFFER_NZ)` is given priority. The buffer regions along the horizontal directions are placed at the four domain boundaries, whereas those along the vertical direction are placed just at the top of the domain. Nothing is affected in the bottom region. Note that the actual target region unaffected by the nudging (the region excluding the buffer regions) narrows compared to the calculation domain because the buffer region is placed on the inside of calculation domain.

Two examples are as below.

```
&PARAM_ATMOS_GRID_CARTESC
BUFFER_NX = 30,           ; The number of grid for the buffer region along the X direction
BUFFER_NY = 30,           ; The number of grid for the buffer region along the Y direction
BUFFFACT = 1.D0,          ; Stretched factor for grid intervals in the buffer region
/
```

```
&PARAM_ATMOS_GRID_CARTESC
BUFFER_DZ = 5000.D0,      ; The width of the buffer region along the Z direction from the top
                           of the model (a reference) [m]
BUFFER_DX = 300000.D0,    ; The width of the buffer region along the X (zonal) direction ( a
                           reference ) [m]
BUFFER_DY = 300000.D0,    ; The width of the buffer region along the Y (meridional) direction (
                           a reference ) [m]
BUFFFACT_Z = 1.20D0,      ; Stretched factor for grid intervals along the Z direction
BUFFFACT_X = 1.05D0,      ; Stretched factor for grid intervals along the X (zonal) direction
BUFFFACT_Y = 1.05D0,      ; Stretched factor for grid intervals along the Y (meridional) direction
/
```
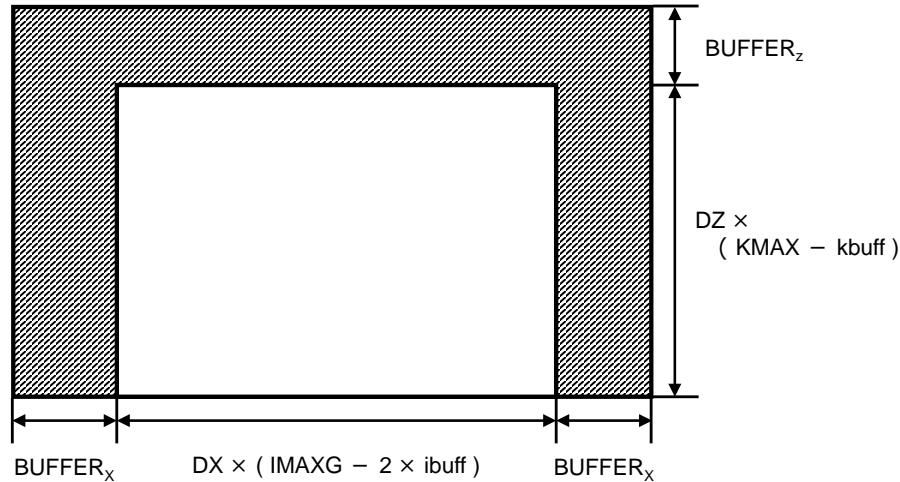
Figure 4.2.3: Location of the buffer region in the entire calculation domain: the shaded area indicates the buffer region. This figure shows the XZ cross-section. It is the same as the YZ cross-section.

The setting procedure of buffer region for the X direction is described as follows. The number of grids `ibuff` in the buffer region is equal to `BUFFER_NX`. If `BUFFER_DX` is configured instead of `BUFFER_NX`, `ibuff` is automatically calculated as the minimum integer satisfying the following condition:

$$\sum_{n=1}^{\texttt{ibuff}} \texttt{BDX}(n) \geq \texttt{BUFFER\_DX}.$$

Thus, it should be noted that the width of the buffer region $\texttt{BUFFER}_\texttt{X}$ $(= \sum_{n=1}^{\texttt{ibuff}} \texttt{BDX}(n))$ does not always correspond to (BUFFER_DX). At the end, the actual target region excluded by the buffer region is expressed as

$$\texttt{DX} \times (\texttt{IMAXG} - 2 \times \texttt{ibuff}). \tag{4.2.6}$$

Although the situations along the Y and Z directions are similar to this, note that the actual target region along the Z direction is expressed as

$$\texttt{DZ} \times (\texttt{KMAX} - \texttt{kbuff}), \tag{4.2.7}$$

using the number of grids `kbuff` in the upper buffer region.

In general, there is no clear criterion for setting the width and locating grids in the buffer region. This depends on a problem to be solved. In SCALE-RM , the followings are recommended: the number of grids in the vertical buffer region at the top of the model is greater than 5, whereas that in the lateral boundaries is approximately 20∼40. Depending on the experiment, it may be necessary to increase the number of grids in the buffer region, to increase the buffer region itself by using the appropriate stretch factor, to tune relaxation time, and so on. The relaxation time is explained below.

The grid intervals in the buffer region are the same as (DX, DY, DZ) in [PARAM_ATMOS_GRID_CARTESC] by default. But, it is possible for them to be stretched by setting (BUFFFACT) > 1. This specification of (BUFFFACT) is applied in all directions if the grid intervals are uniformly specified. When the stretched factor is configured separately in every direction, specify (BUFFFACT_X, BUFFFACT_Y,

BUFFFACT_Z). Note that in case of the configuration of vertical levels by giving (FZ(:)) (refer to 4.2.3.2), the above stretched settings have no effect along the vertical direction.

The grid interval BDX in the buffer region is determined as follows:

$$BDX(n) = DX \times BUFFFACT^n,$$

where n denotes the index of grids in the buffer region, in the order directed from the inner to the outer region in the domain. The grid interval is the same as the inner domain at (BUFFFACT=1.0), whereas it increases from the inner to the outer region by a factor of 1.2 at (BUFFFACT=1.2). Although any value of (BUFFFACT) can be configured, the value from 1.0 to 1.2 is recommended to avoid numerical instability.

Finally, the width of the buffer region BUFFER_X is as follows:

$$BUFFER_X = DX \times \frac{BUFFFACT\left(BUFFFACT^{ibuff} - 1\right)}{BUFFFACT - 1} \tag{4.2.8}$$

Even if the same width of buffer region (BUFFER_DX) is specified, the number of grids in the buffer region decreases with increasing (BUFFFACT). When given by (BUFFER_NX), only the width of buffer region is changed.

Nudging Methods in Buffer Region

[PARAM_ATMOS_BOUNDARY] has parameters to configure the nudging in the buffer region. The boundary data type is configurable by (ATMOS_BOUNDARY_TYPE) in [PARAM_ATMOS_BOUNDARY] (Table 4.2.1.)

Table 4.2.1: Choices of the boundary data type

Value	Description of type
NONE	Do not nudge
CONST	Nudge to a prescribed constant value
INIT	Nudge to the initial value
OFFLINE	Nudge to value read from a file (temporally unchanged)
REAL	Nudge to time-dependent value of the parent model or domain

The following is the parameters in [PARAM_ATMOS_BOUNDARY].

```
&PARAM_ATMOS_BOUNDARY
  ATMOS_BOUNDARY_TYPE = 'NONE',            ; The boundary data type. See Table 4.2.1.
  ATMOS_BOUNDARY_IN_BASENAME = '',         ; File name of the boundary data for OFFLINE or REAL
                                             type

  ATMOS_BOUNDARY_IN_CHECK_COORDINATES \
    = .true.,                              ; Flag to check coordinate variables in the boundary
                                             data file.

(continued on next page)
```

```
ATMOS_BOUNDARY_OUT_BASENAME = '',          ; File name to output the initial boundary data.
ATMOS_BOUNDARY_OUT_TITLE \
  = 'SCALE-RM BOUNDARY CONDITION',         ; Title for the output file.
ATMOS_BOUNDARY_OUT_DTYPE = 'DEFAULT',      ; Data type (REAL4 or REAL8) for the output.
ATMOS_BOUNDARY_USE_DENS = .false.,         ; Switch of the nudging for the density.
ATMOS_BOUNDARY_USE_VELZ = .false.,         ; Switch for the w.
ATMOS_BOUNDARY_USE_VELX = .false.,         ; Switch for the u.
ATMOS_BOUNDARY_USE_VELY = .false.,         ; Switch for the v.
ATMOS_BOUNDARY_USE_PT = .false.,           ; Switch for the θ.
ATMOS_BOUNDARY_USE_QV = .false.,           ; Switch for the vapor.
ATMOS_BOUNDARY_USE_QHYD = .false.,         ; Switch for the hydrometeors.
ATMOS_BOUNDARY_VALUE_VELZ = 0.0D0,         ; Value of the w. Only for CONST type.
ATMOS_BOUNDARY_VALUE_VELX = 0.0D0,         ; Value of the u. Only for CONST type.
ATMOS_BOUNDARY_VALUE_VELY = 0.0D0,         ; Value of the v. Only for CONST type.
ATMOS_BOUNDARY_VALUE_PT = 300.0D0,         ; Value of the θ. Only for CONST type.
ATMOS_BOUNDARY_VALUE_QTRC = 0.0D0,         ; Value of the vapor. Only for CONST type.
ATMOS_BOUNDARY_ALPHAFACT_DENS = 1.0D0,     ; Factor of the 1/τ for the density.
ATMOS_BOUNDARY_ALPHAFACT_VELZ = 1.0D0,     ; Factor for the w.
ATMOS_BOUNDARY_ALPHAFACT_VELX = 1.0D0,     ; Factor for the u.
ATMOS_BOUNDARY_ALPHAFACT_VELZ = 1.0D0,     ; Factor for the v.
ATMOS_BOUNDARY_ALPHAFACT_PT = 1.0D0,       ; Factor for the θ.
ATMOS_BOUNDARY_ALPHAFACT_QTRC = 1.0D0,     ; Factor for the vapor.
ATMOS_BOUNDARY_SMOOTHER_FACT = 0.2D0,      ; Factor of the horizontal smoother against the point-
                                             wise difference.
ATMOS_BOUNDARY_FRACZ = 1.0D0,              ; Fraction for the nudging region to the buffer region
                                             in the z-direction.
ATMOS_BOUNDARY_FRACX = 1.0D0,              ; Fraction in the x-direction.
ATMOS_BOUNDARY_FRACY = 1.0D0,              ; Fraction in the y-direction.
ATMOS_BOUNDARY_TAUZ = DT * 10.0D0,         ; Time scale of the nudging at the top boundary (in
                                             second).
ATMOS_BOUNDARY_TAUX = DT * 10.0D0,         ; Time scale at the western and eastern boundaries.
ATMOS_BOUNDARY_TAUY = DT * 10.0D0,         ; Time scale at the southern and northern bound-
                                             aries.
ATMOS_BOUNDARY_LINEAR_V = .false.,         ; Profile type of the time scale in the z-direction. If
                                             .true., it is a linear profile, otherwise a sinusoidal
                                             profile.
ATMOS_BOUNDARY_LINEAR_H = .false.,         ; Profile type in the x- and y-direction. If .true., it
                                             is a linear profile, otherwise a exponential profile.
ATMOS_BOUNDARY_EXP_H = 2.0D0,              ; Factor of the exponent of the exponential profile.
ATMOS_BOUNDARY_DENS_ADJUST = .false.,      ; Switch of the mass flux adjustment.
ATMOS_BOUNDARY_DENS_ADJUST_TAU \
  = -1.0D0,                                ; Time scale of the density nudging when the mass
                                             flux adjustment is enabled (in second).
```

The tendency due to the nudging is written as

$$\left.\frac{\partial \phi_{k,i,j}}{\partial t}\right|_{\mathrm{nudging}} = -\alpha \Delta \phi_{k,i,j} \tag{4.2.9}$$

$$+\alpha_s \left(\frac{\Delta \phi_{k,i-1,j} + \Delta \phi_{k,i+1,j} + \Delta \phi_{k,i,j-1} + \Delta \phi_{k,i,j+1}}{8} - \frac{\Delta \phi_{k,i,j}}{2} \right),$$

where $\Delta \phi$ is difference from the boundary data and $\alpha_s = \alpha \times$ ATMOS_BOUNDARY_SMOOTHER_FACT. The α is the maximum of those in the tree directions α_x, α_y and α_z. The αs depend on a length scale e as

$$e = \max \left(1 - \frac{d}{\mathrm{BUFFER} \times \mathrm{ATMOS\_BOUNDARY\_FRAC}}, 0 \right), \tag{4.2.10}$$

where d is distance from the boundary. If (`ATMOS_BOUNDARY_LINEAR_V`) = `.true.`,

$$\alpha_z = e_z/\tau_z, \tag{4.2.11}$$

otherwise

$$\alpha_z = \sin^2(\pi e_z/2)/\tau_z, \tag{4.2.12}$$

where τ_z is (`ATMOS_BOUNDARY_TAUZ`). For the horizontal direction, if (`ATMOS_BOUNDARY_LINEAR_H`) = `.true.`,

$$\alpha_x = e_x/\tau_x, \tag{4.2.13}$$

otherwise

$$\alpha_x = e_x \exp\{-(1 - e_x) \times \texttt{ATMOS\_BOUNDARY\_EXP\_H}\}/\tau_x. \tag{4.2.14}$$

α_y is derived by the same way as α_x.

The τ is the relaxation time at the boundary ($d = 0$) with which the difference between the simulated value and the boundary value becomes $1/e$. On the other hand, two grid scale component of $\Delta\phi$ becomes $1/e$ with the time of $\tau/\texttt{ATMOS\_BOUNDARY\_SMOOTHER\_FACT}$ by the second term of the right-hand side of Eq. 4.2.10. The default value of the τ is ten times of (`TIME_DT`). Please refer to Section 4.2.5 for (`TIME_DT`).

If (`ATMOS_BOUNDARY_TYPE`) = "`REAL`" in [`PARAM_ATMOS_BOUNDARY`], the nudging at the top and lateral boundaries is applied for the horizontal velocities, potential temperature, density, and vapor, regardless of the settings of (`ATMOS_BOUNDARY_USE_{VELX,VELY,PT,DESN,QV}`). The nudging for the vertical velocity and hydrometeors is applied when (`ATMOS_BOUNDARY_USE_VELZ`)=`.true.` and (`ATMOS_BOUNDARY_USE_QHYD`)=`.true.`, respectively. In the case of an online nesting simulation (See Section 4.2.8.3), the same settings with the "`REAL`" boundary type is applied to the child domain, except that (`ONLINE_USE_VELZ`) and (`ONLINE_BOUNDARY_USE_QHYD`) in [`PARAM_COMM_CARTESC_NEST`] are used instead of (`ATMOS_BOUNDARY_USE_VELZ`) and (`ATMOS_BOUNDARY_USE_QHYD`), respectively.

The density nudging is effective to reduce the bias of total mass in the simulation. However, it also decreases the pressure gradient in the nudging region; the pressure gradient plays a role to transfer the information of boundary data into the inner domain especially in outflow region. As a way to reduce the mass bias keeping the pressure gradient, the adjustment of the mass flux is available. By using the adjustment of the mass flux, the density nudging could be weakened. If (`ATMOS_BOUNDARY_DENS_ADJUSTMENT`) = `.true.`, the mass flux at the lateral boundaries is adjusted so that the difference in the total mass between the parent model and the simulation becomes small. For example, if the total mass in the simulation is less than that in the parent model, the ρu at the western boundary and the ρv at the southern boundary are increased, and the ρu at the eastern boundary and the ρv at the northern boundary are decreased to enlarge the total mass convergence. The mass flux adjustment is available only for the `REAL` boundary type and child domain of the online nesting. When the mass flux adjustment is enabled, the time scale of the density nudging is specified by (`ATMOS_BOUNDARY_DENS_ADJUSTMENT_TAU`). If the value is negative, the time scale is internally determined as 6 times smaller than the time interval of the boundary data. To weaken the density nudging, set the time scale larger than (`ATMOS_BOUNDARY_TAUX`) and (`ATMOS_BOUNDARY_TAUY`). The adjustment method in SCALE-RM is constructed under the assumption that the total mass change due to convergence of the mass flux at the lateral boundaries is much larger than that due to the physical processes, such as precipitation and surface latent heat flux. The validity of the assumption depends on target domain and situation.

There exists a similar dumping method at near the top boundary, i.e., Rayleigh dumping. See Section 4.2.3.4.

4.2.3.6   Setting vertical grids for ocean, land, and urban models

Settings for horizontal grids of ocean, land, and urban models are same as those for the atmospheric model. The numbers of the grids are specified by (`IMAXG, JMAXG`) in [`PARAM_ATMOS_GRID_CARTESC_INDEX`]. The horizontal grid intervals are specified by (`DX, DY`) in [`PARAM_ATMOS_GRID_CARTESC`]. On the other hand, settings for vertical grids should be specified for each model separately.

Please see Sec. 4.4.7, Sec. 4.4.8 and Sec. 4.4.9 for vertical grid settings for ocean, land and urban models, respectively.

4.2.4 Setting the map projection

In SCALE-RM , grids are allocated based on actual distance. The latitude and longitude for all grids are calculated by using the latitude and longitude of a certain reference point using a map projection. All information pertaining to the latitude and longitude of the grids are output to all output files in netCDF format. The locations of the domain and the map projection can be configured in (PARAM_MAPPROJECTION). This configuration must be shared among the configuration files pp.conf, init.conf, and run.conf. An example is as follows:

```
&PARAM_MAPPROJECTION
 MAPPROJECTION_basepoint_lon = 138.727778D0,
 MAPPROJECTION_basepoint_lat = 35.360556D0,
 MAPPROJECTION_type = 'MER',                    ; Choose from Table 4.2.2.
 /
```

Table 4.2.2: Map projections selectable in SCALE-RM

MAPPROJECTION_type	Map projection
NONE	No map projection for ideal experiment (default)
LC	Lambert conformal conic projection
PS	Polar stereo projection
MER	Mercator projection
EC	Equi-rectangular projection

(MAPPROJECTION_basepoint_lat, MAPPROJECTION_basepoint_lon) are the latitude and longitude at a reference point. This reference point is the center of the calculation domain by the default setting. In SCALE-RM , the north and south latitudes are expressed by positive and negative values, respectively. The east and west longitudes are also expressed by positive and negative values, respectively. It is possible to express longitude using an angle greater than 180°. In above setting, the center of the domain is configured at 35.360556°N and 138.727778°E, where N and E denote the north latitude and east longitude, respectively. The entire calculation domain is placed at the center with a specified size.

(MAPPROJECTION_type) provides the kind of map projection and MER the Mercator projection. Table 4.2.2 shows the selectable map projections in the current version of SCALE-RM . In the case of the Mercator projection, the standard latitude, to which a cylinder is tangent, is specified as (MAPPROJECTION_M_lat) in units of degrees. In general, the standard latitude is set to the Equator. In SCALE-RM , (MAPPROJECTION_basepoint_lat) is used as standard latitude, unless (MAPPROJECTION_M_lat) is explicitly specified. This is because this method is more precise with less distortion than the usual Mercator projection.

The text below explains the Lambert conformal conic projection, the usage frequency of which is highest. The example is the same as that in the file used in the tutorial for the real atmospheric experiment.

```
&PARAM_MAPPROJECTION
 MAPPROJECTION_basepoint_lon = 135.220404,
 MAPPROJECTION_basepoint_lat = 34.653396,
 MAPPROJECTION_type = 'LC',
 MAPPROJECTION_LC_lat1 = 30.0,
 MAPPROJECTION_LC_lat2 = 40.0,
 /
```

In SCALE-RM , two standard parallels are used for this projection. The north and south of them are specified in (MAPPROJECTION_LC_lat1, MAPPROJECTION_LC_lat2) in units of degrees. In the region between the latitudes, the ratio of the latitudinal to longitudinal length is adjusted
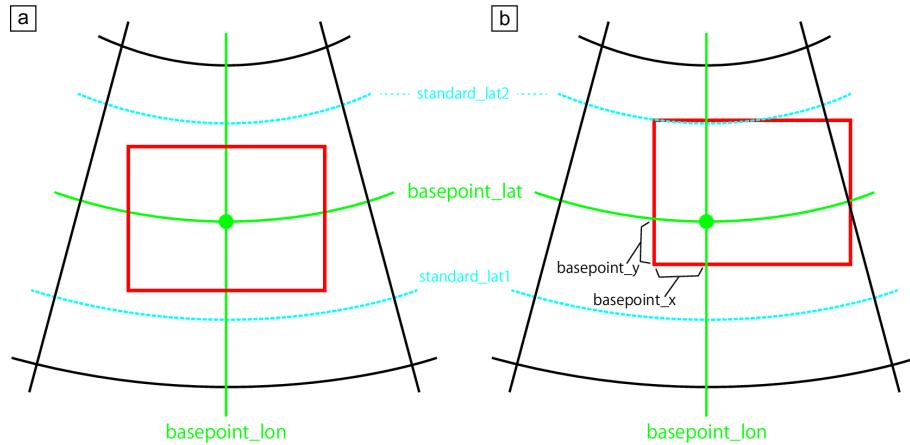
Figure 4.2.4: The relationship between projection center and calculation domain.; (a) Default setting and (b) the case where the center of projection is displaced at the center of the domain. The red line indicates the boundary of the calculation domain.

to be close to the Earth's ellipsoid face.

Furthermore, it is possible for the reference point ((MAPPROJECTION_basepoint_lon), (MAPPROJECTION_basepoint_lat)) to be displaced from the default setting, i.e., domain center as follows:

```
&PARAM_MAPPROJECTION
 MAPPROJECTION_basepoint_lon = 135.220404,
 MAPPROJECTION_basepoint_lat = 34.653396,
 MAPPROJECTION_basepoint_x = 100.0,
 MAPPROJECTION_basepoint_y = 100.0,
 MAPPROJECTION_type = 'LC',
 MAPPROJECTION_LC_lat1 = 30.0,
 MAPPROJECTION_LC_lat2 = 40.0,
/
```

The location of the reference point is specified as the distance from the south-west corner (bottom-left corner).; (MAPPROJECTION_basepoint_x) and (MAPPROJECTION_basepoint_y) are the distances in units of meters between the south-west corner and the reference point in X direction and in Y direction, respectively. If they are not specified, the center of projection corresponds to that of the domain. Figure 4.2.4 shows both cases.

4.2.5 Setting integration period and time step

The integration period and time step are configured appropriately according to experimental design. The time step depends on the spatial resolution of the model. A shorter time step is sometimes required to avoid numerical instability. The period of integration and the time step are configured in [PARAM_TIME] in the configuration file for simulation run .

```
&PARAM_TIME
  TIME_STARTDATE = 2014, 8, 10, 0, 0, 0,    Start date of integration: it is required for the cal-
                                            culation of the radiation process.
  TIME_STARTMS = 0.D0,                       Start date [mili sec]
  TIME_DURATION = 12.0D0,                    Integration    time    [init    is    defined    by
                                            TIME_DURATION_UNIT]
  TIME_DURATION_UNIT = "HOUR",               Unit for TIME_DURATION
  TIME_DT = 60.0D0,                          Time step for time integration
  TIME_DT_UNIT = "SEC",                      Unit for TIME_DT
  TIME_DT_ATMOS_DYN = 30.0D0,                Time step for calculation of dynamical process
  TIME_DT_ATMOS_DYN_UNIT = "SEC",            Unit for TIME_DT_ATMOS_DYN
  TIME_DT_ATMOS_PHY_CP = 600.0D0,            Time step for calculation of cumulus parameteriza-
                                            tion process
  TIME_DT_ATMOS_PHY_CP_UNIT = "SEC",         Unit for TIME_DT_ATMOS_PHY_CP
  TIME_DT_ATMOS_PHY_MP = 60.0D0,             Time step for calculation of microphysics process
  TIME_DT_ATMOS_PHY_MP_UNIT = "SEC",         Unit for TIME_DT_ATMOS_PHY_MP
  TIME_DT_ATMOS_PHY_RD = 600.0D0,            Time step for calculation of radiation process
  TIME_DT_ATMOS_PHY_RD_UNIT = "SEC",         Unit for TIME_DT_ATMOS_PHY_RD
  TIME_DT_ATMOS_PHY_SF = 60.0D0,             Time step for calculation of bottom boundary con-
                                            dition (surface process) for atmosphere
  TIME_DT_ATMOS_PHY_SF_UNIT = "SEC",         Unit for TIME_DT_ATMOS_PHY_SF
  TIME_DT_ATMOS_PHY_TB = 60.0D0,             Time step for calculation of turbulence process
  TIME_DT_ATMOS_PHY_TB_UNIT = "SEC",         Unit for TIME_DT_ATMOS_PHY_TB
  TIME_DT_ATMOS_PHY_BL = 60.0D0,             Time step for calculation of boundary layer process
  TIME_DT_ATMOS_PHY_BL_UNIT = "SEC",         Unit for TIME_DT_ATMOS_PHY_BL
  TIME_DT_OCEAN = 300.0D0,                   Time step for calculation of ocean process
  TIME_DT_OCEAN_UNIT = "SEC",                Unit for TIME_DT_OCEAN
  TIME_DT_LAND = 300.0D0,                    Time step for calculation of land process
  TIME_DT_LAND_UNIT = "SEC",                 Unit for TIME_DT_LAND
  TIME_DT_URBAN = 300.0D0,                   Time step for calculation of urban process
  TIME_DT_URBAN_UNIT = "SEC",                Unit for TIME_DT_URBAN
  TIME_DT_ATMOS_RESTART = 21600.D0,          Output interval of restart files for atmospheric vari-
                                            ables
  TIME_DT_ATMOS_RESTART_UNIT = "SEC",        Unit for TIME_DT_ATMOS_RESTART
  TIME_DT_OCEAN_RESTART = 21600.D0,          Output interval of restart files for ocean variables
  TIME_DT_OCEAN_RESTART_UNIT = "SEC",        Unit for TIME_DT_OCEAN_RESTART
  TIME_DT_LAND_RESTART = 21600.D0,           Output interval of restart files for land variables
  TIME_DT_LAND_RESTART_UNIT = "SEC",         Unit for TIME_DT_LAND_RESTART
  TIME_DT_URBAN_RESTART = 21600.D0,          Output interval of restart files for urban variables
  TIME_DT_URBAN_RESTART_UNIT = "SEC",        Unit for TIME_DT_URBAN_RESTART
  TIME_DT_WALLCLOCK_CHECK = 21600.D0,        Interval of checking wallclock
  TIME_DT_WALLCLOCK_CHECK_UNIT = "SEC",      Unit for TIME_DT_WALLCLOCK_CHECK
  TIME_WALLCLOCK_LIMIT = 86400.D0,           Limit of elapse time of wall clock time [sec]
  TIME_WALLCLOCK_SAFE = 0.95D0,              Safety coefficient for elapse time limit
/
```

4.2.5.1 Time Step for Dynamical Processes

(TIME_DT) is the time step for time integration, usually described as Δt. It is used as time

step for tracer advection as well as the basic unit for all physical processes. To avoid numerical instability, `(TIME_DT)` must satisfy the following condition: it is less than the value calculated by dividing grid size by a supposed maximum advection velocity. A time step for dynamical process, i.e. `(TIME_DT_ATMOS_DYN)`, should be given shorter than Δt because the time integration of dynamic variables is constrained not by advection velocity, but by the speed of the acoustic wave. `(TIME_DT_ATMOS_DYN)` depends on the time integration scheme in relation to the stability of calculation. As a criterion, the standard values of `(TIME_DT_ATMOS_DYN)` are calculated by dividing the minimum grid interval by 420 m/s and 840 m/s, in the case `(ATMOS_DYN_TINTEG_SHORT_TYPE="RK4, RK3")`, respectively. Note that `(TIME_DT_ATMOS_DYN)` needs to be a divisor of `(TIME_DT)`. When the ratio of `(TIME_DT)` to `(TIME_DT_ATMOS_DYN)` is too large, the numerical instability sometimes occurs. The ratio of `(TIME_DT)`/`(TIME_DT_ATMOS_DYN)` is recommended to be set two or three. See also Section 4.2.5.2. In stead of setting `(TIME_DT_ATMOS_DYN)` and `(TIME_DT_ATMOS_DYN_UNIT)`, the ratio of `(TIME_DT)`/`(TIME_DT_ATMOS_DYN)` can be specified by `(TIME_NSTEP_ATMOS_DYN)`. Integer is acceptable for `(TIME_NSTEP_ATMOS_DYN)`.

4.2.5.2 Check of CFL Condition

The time step for the advection `(TIME_DT)` must be smaller than the grid spacing divided by velocity, i.e., the Courant-Friedrichs-Lewy (CFL) condition. A non-dimensional number $U\Delta t/\Delta x$ is called the Courant number, where $U, \Delta x$ and Δt are velocity, spatial grid spacing, and time step, respectively. The CFL condition is that the Courant number must be smaller than 1.

SCALE-RM has a functionality to check whether the Courant number exceeds a limit. To enable this function, set `(ATMOS_VARS_CHECKCFL_SOFT)` and/or `(ATMOS_VARS_CHECKCFL_HARD)` in `[PARAM_ATMOS_VARS]`. Their default values are 1.0 and 2.0, respectively. If the Courant number in simulations exceeds `(ATMOS_VARS_CHECKCFL_SOFT)`, the following message is output to the LOG file.

```
INFO [ATMOS_vars_monitor] Courant number = xxx exceeded the soft limit = yyy
```

If it exceeds `(ATMOS_VARS_CHECKCFL_HARD)`, the following message is output to the standard output, and the simulation aborts.

```
ERROR [ATMOS_vars_monitor] Courant number = xxx exceeded the hard limit = yyy
```

4.2.5.3 Time Step for Physical Processes

A time step for the physical process represents the timing of the tendency to update given by the process. Once the model starts, each physical process is called during the setup of the model to obtain the initial tendency. Each tendency is updated at every time step specified process by process. All time steps for the physical process must be a multiple of `(TIME_DT)`.

The surface fluxes are calculated by the surface process for the atmosphere. On the contrary, if a model grid contains several types of land use, such as ocean, urban, and land, ocean, land, and urban models are used and the fluxes are calculated by these models. The grid mean value of fluxes is obtained as the weighted average of fluxes over each instance of land use according to the fraction of land use.

As described above, the initial tendencies of all processes are updated during the setup of the model. Therefore, the output intervals of restart file are required as multiples of time steps for all processes. If not, a restart calculation disagrees with the continuous calculation. When `(TIME_DT_ATMOS_RESTART)`, `(TIME_DT_OCEAN_RESTART)`, `(TIME_DT_LAND_RESTART)`, and `(TIME_DT_URBAN_RESTART)`, are not specified, the restart files are created at the end of the simulation, i.e. at `(TIME_DURATION)`. The details of the restarted simulation are described in Section 4.2.7.

4.2.5.4 Finalization by wall-clock timer

Some batch job system usually have the limit of the execution time. However, it is difficult to estimate the elapse time of the long-termed simulation, and the job sometimes exceeds the time limit. To solve this problem, SCALE-RM has the finalize option by using the self timer.

When the elapse time reached the time specified by (TIME_WALLCLOCK_LIMIT) (in second), the simulation outputs the restart file and finalize the time loop, even if the timestep is not finished. There is the safety factor for (TIME_WALLCLOCK_LIMIT). The default value is 0.9 and specified by (TIME_WALLCLOCK_SAFE).

As described above, the interval of the restart output should be a multiples of time steps for all physical processes and surface submodels. However, the self timer will stop the simulation suddenly. To avoid restart output at a time other than expected timing, you can specify the timing to check the wall clock time. The wall clock will be checked the time interval specified by (TIME_DT_WALLCLOCK_CHECK) and (TIME_DT_WALLCLOCK_CHECK_UNIT). When these parameters are not specified, maximum time interval in the physical process and the surface submodel is set. Note that if you set a very long interval for checking, the timing of the finalization may delay.

In the sample above, the (TIME_WALLCLOCK_LIMIT) and (TIME_WALLCLOCK_SAFE) are set to 24 hours and 0.95, respectively. The wall clock is checked every 6 hours of the simulation time. When the elapse time exceeds 22.8 hours, the restart file will be generated and the simulation will stop.

4.2.6 Setting the history file and output variable

A history file and output variables are configured at [PARAM_FILE_HISTORY_CARTESC], [PARAM_FILE_HISTORY] and [HISTORY_ITEM] in run.conf. The default format of the history file is configured at [PARAM_FILE_HISTORY].

```
&PARAM_FILE_HISTORY_CARTESC
  FILE_HISTORY_CARTESC_PRES_nlayer = -1,    ; Number of pressure levels
                                              (OPTION for interpolation at pressure level)
  FILE_HISTORY_CARTESC_PRES = 0.D0          ; Pressure levels in order from lower to upper [hPa]
                                              (OPTION for interpolation at pressure level)
  FILE_HISTORY_CARTESC_BOUNDARY = .false.,  ; Whether output data in halo or not.
                                              .true.: Output, .false.: Not output.
/
```

```
&PARAM_FILE_HISTORY
 FILE_HISTORY_TITLE = "",                              ;  Brief  description  of  data  (See
                                                       Sec.5.1.1)
 FILE_HISTORY_SOURCE = "",                             ;  Name  of  the  source  software  (See
                                                       Sec.5.1.1)
 FILE_HISTORY_INSTITUTION = "",                        ; Data author (See Sec.5.1.1)
 FILE_HISTORY_TIME_UNITS = "seconds",                  ; Unit of time axis in netCDF
 FILE_HISTORY_DEFAULT_BASENAME = "history_d01",        ; Base name of the output file
 FILE_HISTORY_DEFAULT_POSTFIX_TIMELABEL = .false.,     ; Add the time label to the file name?
 FILE_HISTORY_DEFAULT_ZCOORD = "model",                ; Type of vertical coordinate
 FILE_HISTORY_DEFAULT_TINTERVAL = 3600.D0,             ; Time interval of history output
 FILE_HISTORY_DEFAULT_TUNIT = "SEC",                   ; Unit of DEFAULT_TINTERVAL
 FILE_HISTORY_DEFAULT_TSTATS_OP = "none",              ; Statistics operation
 FILE_HISTORY_DEFAULT_DATATYPE = "REAL4",              ; Output data type: REAL4 or REAL8
 FILE_HISTORY_OUTPUT_STEP0 = .true.,                   ; Output data at initial time (t=0)?
 FILE_HISTORY_OUTPUT_WAIT = 0.D0,                      ; Time to suppress output
 FILE_HISTORY_OUTPUT_WAIT_TUNIT = "SEC",               ; Unit of OUTPUT_WAIT
 FILE_HISTORY_OUTPUT_SWITCH_TINTERVAL = -1.D0,         ; Time interval to switch the file
 FILE_HISTORY_OUTPUT_SWITCH_TUNIT = "SEC",             ; Unit of OUTPUT_SWITCH_TINTERVAL
 FILE_HISTORY_ERROR_PUTMISS = .true.,                  ;  Check  missing  preparation  of  the
                                                       data?
 FILE_HISTORY_AGGREGATE = .false.,                     ; Aggregate to one file with PnetCDF?
 /
```

In default, each process output the history file. When (FILE_HISTORY_AGGREGATE) is set to .true., the distributed output file is aggregated into one file by using parallel NetCDF . The default setting of (FILE_HISTORY_AGGREGATE) is determined by the (FILE_AGGREGATE) in [PARAM_FILE]. Please refer to Section 5.1.1

(FILE_HISTORY_DEFAULT_TINTERVAL) is time interval of history output and its unit is defined by (FILE_HISTORY_DEFAULT_TUNIT). The unit can be selected from "MSEC", "msec", "SEC", "sec", "s", "MIN", "min", "HOUR", "hour", "h", "DAY", "day". When the average value output is selected as (FILE_HISTORY_DEFAULT_TSTATS_OP) to = "mean", the history data averaged over the last period, given as (FILE_HISTORY_DEFAULT_TINTERVAL), is output. As the same way, when "min" or "max" is set to (FILE_HISTORY_DEFAULT_TSTATS_OP), the minimam or maximam value over the last period is output, respectively.

The time interval of history output must be the equal to or a multiple of the time interval of its related scheme. When you want to disable checking this consistency, please set (FILE_HISTORY_ERROR_PUTMISS) to .false..

When (FILE_HISTORY_DEFAULT_POSTFIX_TIMELABEL) is set to .true., the time label is added to the name of output file. The time label is generated from the current time in the simulation, and its format is defined by YYYYMMDD-HHMMSS.msec.

When (FILE_HISTORY_OUTPUT_STEP0) is set to .true., the variables at the time before time integration (initial state) are output to history file. You can suppress history output during the time in the simulation defined by (FILE_HISTORY_OUTPUT_WAIT) and (FILE_HISTORY_OUTPUT_WAIT_TUNIT). If the value is negative, no suppression occurs. (FILE_HISTORY_OUTPUT_SWITCH_TINTERVAL) is time interval of switching output file and its unit is defined by (FILE_HISTORY_OUTPUT_SWITCH_TUNIT). If the value is negative only one file per process is used for history output. If this option is enabled, the time label is added to the file name.

Three vertical coordinates are available to output atmospheric 3-D variables. (FILE_HISTORY_DEFAULT_ZCOORD) = "model" is selected in default. In this case, the variable is output with original coordinate of the model (terrain-following, z* coordinate for SCALE-RM). When (FILE_HISTORY_DEFAULT_ZCOORD) is set to "z", the variables are interpolated to absolute height the number of levels in output data is the same as that of the model level. The height at each level is the same as that of the model level in a grid cell with no topography. The variables are interpolated to the pressure level when (FILE_HISTORY_DEFAULT_ZCOORD) is set to "pressure". In this case, the configurations of (FILE_HISTORY_CARTESC_PRES_nlayer) and

(FILE_HISTORY_CARTESC_PRES) in [PARAM_FILE_HISTORY_CARTESC] are required.

When (FILE_HISTORY_CARTESC_BOUNDARY) in [PARAM_FILE_HISTORY_CARTESC] is .true., the data in halo, which is located outside of a target domain, is also output except in the case of periodic boundary condition. The setting of (FILE_HISTORY_CARTESC_BOUNDARY) is applied to all output variables.

The output variables are set by adding [HISTORY_ITEM]. The list of variables that can be output can be found in the History Variables List on the Reference Manual of SCALE HP (see Section 5.3.2 for details). The output format follows the default setting specified at [PARAM_FILE_HISTORY]. By adding namelists with "(OPTION)", the format for a specific variable can be changed from the default setting.

```
&HISTORY_ITEM
  NAME = "RAIN",                Variable name
  OUTNAME = "",                 (OPTION) same as NAME
  BASENAME = "rain_d01",        (OPTION) same as FILE_HISTORY_DEFAULT_BASENAME
  POSTFIX_TIMELABEL = .false.,  (OPTION) same as FILE_HISTORY_DEFAULT_POSTFIX_TIMELABEL
  ZCOORD = "model",             (OPTION) same as FILE_HISTORY_DEFAULT_ZCOORD
  TINTERVAL = 600.D0,           (OPTION) same as FILE_HISTORY_DEFAULT_TINTERVAL
  TUNIT = "SEC",                (OPTION) same as FILE_HISTORY_DEFAULT_TUNIT
  TSTATS_OP = "mean",           (OPTION) same as FILE_HISTORY_DEFAULT_TSTATS_OP
  DATATYPE = "REAL4",           (OPTION) same as FILE_HISTORY_DEFAULT_DATATYPE
/
```

If the variable requested by [HISTORY_ITEM] is not prepared during the time stepping of the simulation, the execution will stop and error log is written to the log file. This case may occur if there is the spelling miss of the (NAME), or when requested variables are not used in the selected scheme.

Namelists with "(OPTION)" are applied only to the variable (NAME). If the default setting is used for the variable, the namelist with "(OPTION)" can be omitted. For example, let the below setting for [HISTORY_ITEM] be added, maintaining the above setting of [PARAM_FILE_HISTORY]. The snapshot values of U and V are stored as four-byte real values at an interval of 3600 s in the file history_d01.xxxxxx.nc, whereas the value of RAIN averaged over the last 600 seconds is stored each time in the file. The value of T is output as T in a rule same as U and V, and is also output as T_pres with interpolation to pressure coordinate.

```
&HISTORY_ITEM NAME="T" /
&HISTORY_ITEM NAME="U" /
&HISTORY_ITEM NAME="V" /
&HISTORY_ITEM NAME="RAIN", TINTERVAL=600.D0, TSTATS_OP="mean" /
&HISTORY_ITEM NAME="T", OUTNAME="T_pres", ZCOORD="pressure" /
```

4.2.7 How to restart run

The restart function is beneficial to avoid termination of simulations because of the limitation of the job execution time controlled by a computer system. You can divide a long sequential simulation into multiple runs by restarting. The restart files have the same format as the data generated by the initialization run. The function allows outputting multiple restart files at the specified time interval, not only the end of each simulation. The settings for the restart files are configured in [PARAM_RESTART] and [PARAM_TIME] in the configuration file for simulation run . The example below indicates that the simulation restarts by the file restart1_*** and generates restart files

`restart2_***` every six hours.

```
&PARAM_RESTART
 RESTART_IN_BASENAME = "restart1_d01_20070715-000000.000",
                                      Basename of input initial file or restart file.
 RESTART_IN_POSTFIX_TIMELABEL = .false.    Add  initial  date  and  time  after
                                      RESTART_IN_BASENAME?
 RESTART_OUTPUT = .true.,              Output restart file? Default is .false.
 RESTART_OUT_BASENAME = "restart2_d01",   Basename of output restart file.
 RESTART_OUT_POSTFIX_TIMELABEL = .true.   Add  output  date  and  time  after
                                      RESTART_OUT_BASENAME?
 RESTART_OUT_TITLE = "",               Title written in the restart file.
 RESTART_OUT_DTYPE = "DEFAULT",        REAL4 or REAL8 or DEFAULT
/

&PARAM_TIME
 TIME_STARTDATE = 2007, 7, 15, 00, 0, 0,   Start date of restart run
 TIME_STARTMS = 0.D0,                 Start date [ms]
 TIME_DURATION = 12.0D0,              Integration time
 TIME_DURATION_UNIT = "HOUR",         Unit for TIME_DURATION
 ..... *snip* .....
 TIME_DT_ATMOS_RESTART = 21600.D0,    Output interval of restart files for atmospheric vari-
                                      ables
 TIME_DT_ATMOS_RESTART_UNIT = "SEC",  Unit for TIME_DT_ATMOS_RESTART
 TIME_DT_OCEAN_RESTART = 21600.D0,    Output interval of restart files for ocean variables
 TIME_DT_OCEAN_RESTART_UNIT = "SEC",  Unit for TIME_DT_OCEAN_RESTART
 TIME_DT_LAND_RESTART = 21600.D0,     Output interval of restart files for land variables
 TIME_DT_LAND_RESTART_UNIT = "SEC",   Unit for TIME_DT_LAND_RESTART
 TIME_DT_URBAN_RESTART = 21600.D0,    Output interval of restart files for urban variables
 TIME_DT_URBAN_RESTART_UNIT = "SEC",  Unit for TIME_DT_URBAN_RESTART
/
```

The time intervals for output of restart files are specified by (TIME_DT_ATMOS_RESTART), (TIME_DT_OCEAN_RESTART), (TIME_DT_LAND_RESTART), and (TIME_DT_URBAN_RESTART). When they are not specified, the restart files are generated at the end of the simulation, i.e., (TIME_DURATION). The file names of output restart files are specified by (RESTART_OUT_BASENAME).
(RESTART_OUT_POSTFIX_TIMELABEL) indicates whether date and time at output are automatically added to the file name after (RESTART_OUT_BASENAME).
The default setting is (RESTART_OUT_POSTFIX_TIMELABEL) = .true..

The restart files are not compatible for all simulation. The variables included in the restart file are different by the schemes, which chosen in the configuration file. The simple way for preparing restart files while keeping consistency is to use the same setting for the schemes in a series of the simulations.

The other settings are basically the same as the normal run. (RESTART_IN_BASENAME) is the name of the input file, which includes initial state of the atmosphere and surface sub-models. The normal run usually uses `init_***` prepared by `scale-rm_init`, while the restart run uses a restart file output by the previous run. (RESTART_IN_POSTFIX_TIMELABEL) is the same as (RESTART_OUT_POSTFIX_TIMELABEL), but for (RESTART_IN_BASENAME). The default setting is (RESTART_IN_POSTFIX_TIMELABEL) = .false..
In avobe example, setting (RESTART_IN_BASENAME) = "restart1_d01_20070715-000000.000" is equivalent to setting (RESTART_IN_POSTFIX_TIMELABEL) = .true. and (RESTART_IN_BASENAME) = "restart1_d01". (TIME_STARTDATE) and (TIME_DURATION) represent the start date and the integration time for the restart simulation.

For a realistic atmospheric experiment, the boundary data prepared by `scale-rm_init` is needed in addition to the initial data. An example is as follows:

```
&PARAM_ATMOS_BOUNDARY
  ATMOS_BOUNDARY_TYPE = "REAL",                              "REAL":  Real  case  experi-
                                                            ment
  ATMOS_BOUNDARY_IN_BASENAME = "../init/output/boundary_d01",  Head of file name of bound-
                                                            ary data
/
```

In the restart simulation, an appropriate date of the boundary data will be read from the boundary file `boundary_***.nc`.

4.2.8   How to conduct nesting experiment

"Nesting" is the method for settings multiple computational domains overlapping one another in a nested structure. Figure 4.2.5 shows an example of nesting using three domains. An outer domain is configured to be wider with coarser resolution to express larger-scale phenomena, while an inner domain is set narrower with finer resolution to represent smaller-scale phenomena. The result calculated in the outer domain is used for the boundary condition for the inner domain. Hereinafter, the outer domain that sends data is called the "parent domain" and the inner domain which receives data is called the "child domain."

Nesting methods are categorized as follows:

- Execution procedure

  Online nesting
  The computations in the parent and child domains are simultaneously conducted by exchanging information.

  Off-Line nesting
  Following the computation of the parent domain, the initial and boundary conditions of the child domain are formulated, and then the child domain is computed.

- Method of data exchange

  One-way nesting
  The parent domain sends data to the child domain. The latter does not send data to the former. Thus, data flow is one way. The results of the parent domain are not affected by those of the child domain.

  Two-way nesting
  The parent domain sends data to the child domain. The former also receives data from the latter. Thus, the calculations of the domains influence each other. Although this method can be applied in the case of online nesting, it has not yet been implemented in SCALE-RM v5.5.1.

The difference between online and offline nestings lie in the update frequency of providing data from the parent domain to the child domain. In an online nesting experiment, the boundary conditions of the child domain are updated at every time step (Δt) of the parent domain. In an offline nesting experiment, the update interval depends on the output interval of the history file in the calculation of the parent domain.

Regardless of whether the nesting is online or offline, the ratio of the grid interval of the parent domain to that of the child domain ($\mathrm{DX}_{\mathrm{d01}}/\mathrm{DX}_{\mathrm{d02}}$) is not limited in terms of code implementation. However, there is the possibility of degradation in the physical performance of the results of calculations if the ratio is too high. In SCALE-RM , it is recommended that this ratio be equal to or lower than 5.

In this section, the configuration files for the parent and child domains are denoted by `***.d01.conf` and `***.d02.conf`, respectively.

4.2.8.1   Treatment of topography in child domain

In the nesting experiment, the resolutions of the topography were different between the parent and the child domains due to their different spatial resolutions. In the relaxation area of the child domain (refer to Section 4.2.3.5), the atmospheric variables were nudged toward those of the parent domain. If the representations of the topography are different between two domains, the reference data for nudging, calculated in the parent domain, often does not exist. In this case, atmospheric data at the missing levels are estimated by extrapolation. However, this may incur error if the estimation by the extrapolation is not accurate.

In order to avoid such inconsistency due to differences in topographies, the use of "the topography-copy" function in SCALE-RM is recommended. This function copies the topography of the parent
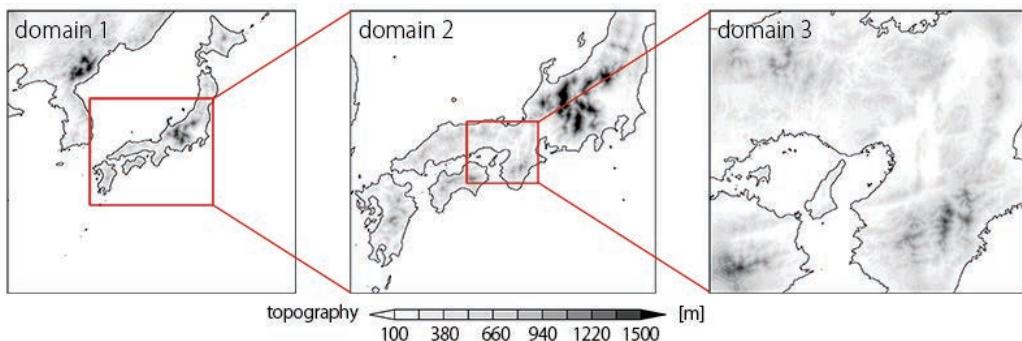
Figure 4.2.5: An example of domain nesting targeting the western area of Japan. Domain 1 is the outermost domain and domain 3 the innermost. The red rectangle and lines represents the domain's location and its relations with other domains. The horizontal grid spacings are 7.5 km, 2.5 km, and 0.5 km for Domains 1, 2, and 3, respectively.

domain onto that of the child domain in the relaxation area. If this function is used, the topography of the relaxation area in the child domain perfectly corresponds to that in the parent domain, as shown in 4.2.6. Furthermore, in order to increase the resolution of topography gradually from that in the parent domain to that in the inner domain, the topography transition area is present on the inside of the relaxation area. In the topography transition area, topography is generated by weighing those of the parent and the child domains. By default, the width of the topography transition area is identical to that of the relaxation area. In the calculation area inside the topography transition area, the topography is given by that of the child domain. If "the making tool for the complete settings of the experiment" is used (refer to 4.2.1), the topography-copy function is automatically applied.

The file `pp.d0*.conf` mentioned in this section can be generated by "the making tool for the complete settings of the experiment" by renaming
`${Tutorial_dir}/real/sample/USER.online-nesting.sh` as `USER.sh`. This may help users understand this setting. The configuration and execution methods are explained below.

How to use topography-copy function

To use the parent topography for the topography-copy function, edit file `pp.d02.conf` for the child domain as follows. Here, the output data of topography in the parent domain is assumed to be saved as file `topo_d01.pe***.nc` in SCALE netCDF file format. As other file formats, GrADS format is supported (Sec. 4.1.2).
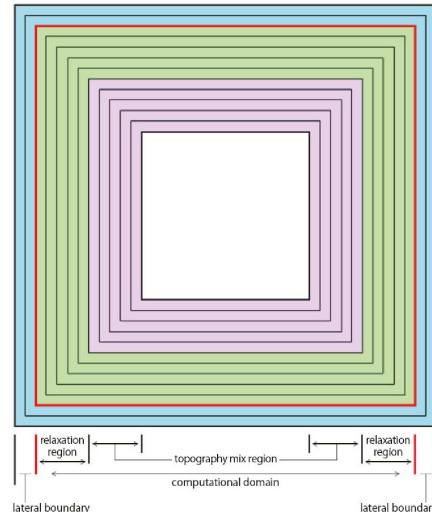
Figure 4.2.6: Horizontal distribution of topography when the topography-copy function is applied. The outermost grids shaded in light blue represent the `HALO` region, and the number of grids depends on the horizontal advection scheme. These grids are the lateral boundary. The area demarcated by the red line is the calculation domain. The green- and rose-colored areas are the relaxation and the topographical transition areas, respectively. The innermost area is where the resolution of topography is the same as that of the original child domain. In the topography transition area, topography is altered gradually from that of the parent domain to the original child domain.

```
&PARAM_CNVTOPO
  ~ .... ~
  CNVTOPO_copy_parent = .true.,        whether the topography-copy function is applied
/

&PARAM_COPYTOPO
  COPYTOPO_IN_BASENAME = "topo_d01",   base name of the file of parent topography data
  COPYTOPO_IN_FILETYPE = "SCALE",      type of the file of parent topography data (SCALE,
                                       GrADS)
  COPYTOPO_TRANSITION_DX = -1,         Width of transition area in x-direction
  COPYTOPO_TRANSITION_DY = -1,         Width of transition area in y-direction
  COPYTOPO_ENTIRE_REGION = .false.,    whether the parent's topography is copied over the entire
                                       child domain
  COPYTOPO_LINEAR_H = .true.,
/
```

If (CNVTOPO_copy_parent) in [PARAM_CNVTOPO] is .true., the topography-copy function is applied. (COPYTOPO_ENTIRE_REGION) is an option whereby the parent's topography can be copied over the entire child domain If this is .true., the topography in the child domain is completely copied from that in the parent domain. (COPYTOPO_LINEAR_H) is the parameter to determine which the topography transition method is applied. If (COPYTOPO_LINEAR_H) = .true., the ratio of the parent's topography to the child's topography is linearly changed. Otherwise, it is changed exponentially. The width of transition area is specified by (COPYTOPO_TRANSITION_DX, COPYTOPO_TRANSITION_DY). If these have negative value, the default setting is applied: the width of transition area equals to that of relaxation area.

Generation of topography

In case of using the topography-copy function, the generation of the topography should start from the parent domain because a child domain requires the catalog file of the parent. If the number of domains is greater than three, the order of execution is as follows:

```
$ mpirun -n [number of processes] ./scale-rm_pp pp.d01.conf
$ mpirun -n [number of processes] ./scale-rm_pp pp.d02.conf
$ mpirun -n [number of processes] ./scale-rm_pp pp.d03.conf
```

4.2.8.2 Offline Nesting Experiment

The following two limitations are imposed for the offline nesting experiment:

- The child domain is completely included in the parent domain.

- The integration time for the child domain is the same as or less than that for the parent domain.

Furthermore, the offline experiment is conducted in the following order:

1. The temporal integration of the parent domain is conducted.

2. The initial and boundary conditions for the child domain are generated using the history or init/restart output of the parent domain.

3. Using the generated initial and boundary data, the temporal integration of the child domain is conducted.

An explanation is provided according to the above workflow.

Time Integration of Parent Domain

In order to prepare the data in parent domain which is used as the boundary data for the child domain, there are some necessary settings. The configuration files for the simulation in the parent domain can be generated by using "the supporting tool for the preparation of configuration file" (See 4.2.1). Rename the sample files
`${Tutorial_dir}/real/sample/USER.offline-nesting-parent.sh` to `USER.sh`, edit it, and run `make`.
    The time integration of the parent domain is carried out as a single-domain computation. However, the following five aspects need to be considered in the configuration.

- All variables required for the computation in the child domain have already been generated with the computation of the parent domain as history or restart output.

- The interval of the history or restart output is sufficiently short.

- In case the starting time of the computation in the child domain is the same as that in the parent domain, history output data at $t = 0$ in the parent domain is required.

If you want to use history file, setting like the following is required.

```
&PARAM_FILE_HISTORY
  FILE_HISTORY_DEFAULT_BASENAME = "history",
  FILE_HISTORY_DEFAULT_TINTERVAL = 900.D0,      Time interval of history data output
  FILE_HISTORY_DEFAULT_TUNIT = "SEC",           Unit of FILE_HISTORY_DEFAULT_TINTERVAL
  FILE_HISTORY_DEFAULT_TSTATS_OP = "none",
  FILE_HISTORY_DEFAULT_DATATYPE = "REAL4",
  FILE_HISTORY_OUTPUT_STEP0 = .true.,           Include the output data at t = 0
/
```

(`FILE_HISTORY_DEFAULT_TINTERVAL`) is the time interval of the history output, which is configured as the update time interval used in the calculation of the child domain. Take care of the free disk space if the time interval of the data output is relatively short. Refer to Section 4.2.6 for details of the items in [`PARAM_FILE_HISTORY`].

If you want to use restart files, the setting is like

```
&PARAM_RESTART
 RESTART_OUTPUT = .true.
 RESTART_OUT_BASENAME = 'restart_d01',
/
&PARAM_TIME
 TIME_DT_ATMOS_RESTART = 900.D0,        Time interval of restart data output
 TIME_DT_ATMOS_RESTART_UNIT = "SEC",
 TIME_DT_OCEAN_RESTART = 900.D0,        Time interval of restart data output
 TIME_DT_OCEAN_RESTART_UNIT = "SEC",
 TIME_DT_LAND_RESTART = 900.D0,         Time interval of restart data output
 TIME_DT_LAND_RESTART_UNIT = "SEC",
 TIME_DT_URBAN_RESTART = 900.D0,        Time interval of restart data output
 TIME_DT_URBASN_RESTART_UNIT = "SEC",
/
```

For details of these parameters, see Section 4.2.7.

All necessary variables for the generation of initial and boundary data for the child domain must be described in [`FILE_HISTORY_ITEM`] in the file the configuration file for simulation run . The necessary variables for offline nesting depends on the setting of the simulation in the child domain, the following is variables in standard real atmospheric simulations.

```
DENS, W (or MOMZ), Umet (or U, MOMX), Vmet (or V, MOMY), PT (or RHOT, T), PRES, QV
LAND_SFC_TEMP, URBAN_SFC_TEMP, OCEAN_SFC_TEMP
OCEAN_SFC_ALB_IR_dir OCEAN_SFC_ALB_IR_dif,
OCEAN_SFC_ALB_NIR_dir OCEAN_SFC_ALB_NIR_dif,
OCEAN_SFC_ALB_VIS_dir OCEAN_SFC_ALB_VIS_dif,
LAND_SFC_ALB_IR_dir, LAND_SFC_ALB_IR_dif,
LAND_SFC_ALB_NIR_dir, LAND_SFC_ALB_NIR_dif,
LAND_SFC_ALB_VIS_dir, LAND_SFC_ALB_VIS_dif,
OCEAN_TEMP, OCEAN_SFC_ZOM, LAND_TEMP, LAND_WATER
```

(Output according to microphysics model used in the parent model)

```
QC, QR, QI, QS, QG
NC, NR, NI, NS, NG
```

Once this configuration is complete, conduct the time integration of the parent domain by `scale-rm`.

Generation of Initial and Boundary Data for Child Domain

To prepare the configuration files for simulations in the child domain can be generated by using "the supporting tool for the preparation of configuration file" (See 4.2.1). Rename the sample files `${Tutorial_dir}/real/sample/USER.offline-nesting-child.sh` to USER.sh, edit it, and run `make`.

In the case that you generate the initial and boundary data using the history data from the simulation for the parent domain, the configuration file for the initialization run is configured as follows:

```
&PARAM_MKINIT_REAL_ATMOS
  NUMBER_OF_FILES = 1,
  BASENAME_ORG = "history_d01",        FILE_HISTORY_DEFAULT_BASENAME    in    the    file
                                       run.d01.conf
  FILETYPE_ORG = "NetCDF",
  BASENAME_BOUNDARY = "boundary_d01",
  BOUNDARY_UPDATE_DT = 900.D0,         time interval of history output (unit:"SEC")
/

&PARAM_MKINIT_REAL_OCEAN
  NUMBER_OF_FILES = 1,
  BASENAME_ORG = "history_d01",        FILE_HISTORY_DEFAULT_BASENAME    in    the    file
                                       run.d01.conf
  FILETYPE_ORG = "NetCDF",
  BASENAME_BOUNDARY = "boundary_d01",
  BOUNDARY_UPDATE_DT = 900.D0,         time interval of history output (unit:"SEC")
/

&PARAM_MKINIT_REAL_LAND
  NUMBER_OF_FILES = 1,
  BASENAME_ORG = "history_d01",        FILE_HISTORY_DEFAULT_BASENAME    in    the    file
                                       run.d01.conf
  FILETYPE_ORG = "NetCDF",
  BASENAME_BOUNDARY = "boundary_d01",
  BOUNDARY_UPDATE_DT = 900.D0,         time interval of history output (unit:"SEC")
/
```

In order to generate the initial and boundary data from the output of SCALE-RM format, (FILETYPE_ORG) is specified as "NetCDF". (BOUNDARY_UPDATE_DT) is set to the same value of (FILE_HISTORY_DEFAULT_TINTERVAL) as in the configuration file in the parent domain (run.d01.conf).

In the case that the restart files are used to generate the initial and boundary data, set (NUMBER_OF_FILES) to the number of the time steps. Additionally, the restart files must be renamed or symbolic links must be created to make file names have incremented number like restart_d01_00000.pe000000.nc as described in Section 4.1.2.

After editing the configuration file, form the initial and boundary data for the child domain by scale-rm_init. If the execution is aborted with the message like the following, it means that the child domain is not completely contained in the parent domain:

```
ERROR [INTERP_domain_compatibility] REQUESTED DOMAIN IS TOO MUCH BROAD
  -- LONGITUDINAL direction over the limit
```

Time Integration in Child Domain

Following the generation of the initial and boundary conditions, compute the time integration in the child domain by scale-rm. This is same as in the usual atmospheric experiment.

If an experiment involving multiple offline nestings is performed, the above procedure is repeated; the results of the above time integration for the child domain are regarded as those for the parent domain, and the initial and boundary data are generated for further inner domain calculations.

4.2.8.3 Online Nesting Experiment

The following two limitations are imposed in carrying out the online nesting experiment:

- The integration time for the child domain is identical to that for the parent domain.

- The time step for the parent domain is a multiple of that for the child domain.

On the other hand, the configurations of vertical layers, map projections, and the physical scheme do not have to be identical in the parent and the child domains. In the online nesting experiment, computations of all domains are conducted simultaneously. In the current version, SCALE-RM supports only one-way nesting. The maximum number of domains allowed is 10.

In the online nesting experiment in SCALE-RM , the time integrations of multiple domains are not serial but parallel. As shown in Fig. 4.2.7, MPI processes are split into several groups; each group manages a domain and computes it, behaving like an independent model. The configuration file launch.conf is needed at execution to boot the multiple domains.
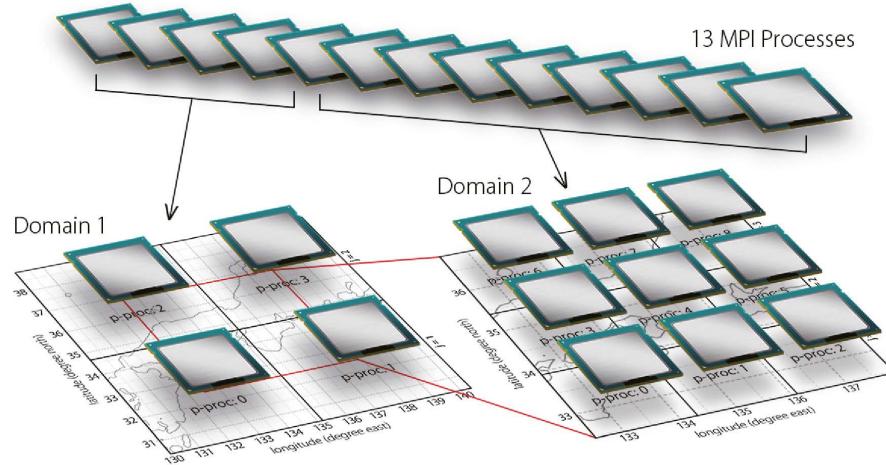
Figure 4.2.7: MPI process distribution in the online nesting experiment. In this example, 13 processes were launched at the beginning. These processes were distributed appropriately; 4-MPI parallel for 2×2 in Domain 1 and 9-MPI parallel for 3×3 in Domain 2 were executed. MPI communication flowed from Domain 1 to Domain 2.

The following explanation is provided for the most simple case of online nesting, two-domain nesting. The experimental set described here can be generated by renaming the sample files ${Tutorial_dir}/real/sample/USER.online-nesting.sh as USER.sh and conducting "the supporting tool for the preparation of configuration file" (refer to Section 4.2.1). The below explanation assumes that the generations of the topography/land-use data and initial/boundary data for each domain have been completed. The procedures for topography generation are described in Section 4.2.8.1.

Configurations for Online Nesting

In the configuration files run.***.conf for the parent and child domains, some nesting settings are added to [PARAM_COMM_CARTESC_NEST]:
— Configuration in run.d01.conf —

```
&PARAM_COMM_CARTESC_NEST
  ONLINE_DOMAIN_NUM = 1,                The domain ID, which is enumerated from the outermost
                                        one as 1.

  ONLINE_IAM_PARENT = .true.,
  ONLINE_IAM_DAUGHTER = .false.,
  ONLINE_BOUNDARY_USE_QHYD = .false.,
  ONLINE_USE_VELZ = .false.,
  ONLINE_AGGRESSIVE_COMM = .false.,
  /
```

— Configuration in `run.d02.conf` —

```
&PARAM_COMM_CARTESC_NEST
  ONLINE_DOMAIN_NUM = 2,                    The domain ID, which is enumerated from the outermost
                                            one as 1.

  ONLINE_IAM_PARENT = .false.,
  ONLINE_IAM_DAUGHTER = .true.,
  ONLINE_BOUNDARY_USE_QHYD = .false.,
  ONLINE_USE_VELZ = .false.,
  ONLINE_AGGRESSIVE_COMM = .false.,
/
```

(ONLINE_DOMAIN_NUM) is the ID number of the domain, which is enumerated from the outermost domain to the innermost. In above example, the ID numbers of the parent and child domains are 1 and 2, respectively.

(ONLINE_IAM_PARENT) and (ONLINE_IAM_DAUGHTER) specify whether each domain has its parent domain and child domain or not. If (ONLINE_IAM_PARENT) = .true. in the Nth domain, the calculation data in the Nth domain is transferred to the child domain with the domain number of $N + 1$. If (ONLINE_IAM_DAUGHTER) = .true., then boundary data in the Nth domain is received from the parent with the domain number of $N - 1$. The outermost domain plays a role only in the parent domain, whereas the innermost domain is involved only in the child domain. Since the intermediate domains are involved in both the parent and child domains, both (ONLINE_IAM_PARENT) and (ONLINE_IAM_DAUGHTER) are .true.. Table 4.2.3 gives the configuration for an N-domain nesting experiment.

Table 4.2.3: A configuration for N-domain nesting

domain	ONLINE_DOMAIN_NUM	ONLINE_IAM_PARENT	ONLINE_IAM_CHILD
the outermost domain	1	.true.	.false.
intermediate domains	$2 - (N - 1)$	.true.	.true.
the innermost domain	N	.false.	.true.

(ONLINE_BOUNDARY_USE_QHYD) specifies whether water condensation (hydrometeors) is used for the boundary condition. When the boundary condition is generated from external input data, water condensations are not usually employed. However, in the SCALE-RM nesting experiment, water condensation calculated in the parent domain can be used for the boundary condition of the child domain because the two domains usually use the same physical schemes. By receiving hydrometeors' condition at the lateral boundary, the delay in the generation of clouds and rain near the inflow boundary of the nested domain is expected to be suppressed. (ONLINE_USE_VELZ) specifies whether vertical velocity is used for the boundary condition in the child domain. (ONLINE_AGGRESSIVE_COMM) is a setting related to MPI communication when passing data from the parent domain to the child domain. The parent domain passes boundary conditions to the child domain at each time step ((TIME_DT)) of the parent domain. If (ONLINE_AGGRESSIVE_COMM) is .true., regardless of whether the child domain has received the data or not, the parent domain passes the data at the parent domain's timing and moves to the next time step of calculation; whereas if .false., the parent domain waits until the child domain has received the data of the previous step, and then passes the data of the current time step. In the case of .true., if the calculation time vastly differs between the parent and child domains, the memory usage will be extensive and the computation may stop due to hardware limitations.

Configuration of Launcher

The online nesting experiment requires the configuration of the launch file `launch.conf` other than `run.***.conf`.

```
&PARAM_LAUNCHER
  NUM_DOMAIN = 2,                                 number of domains
  PRC_DOMAINS = 4, 16,                            MPI processes used for each domain (as many do-
                                                  mains as necessary)
  CONF_FILES = run.d01.conf, run.d02.conf,        The configuration files for each domain (as many
                                                  domains as necessary)
  LOG_SPLIT = .false.,                            log-output for mpi splitting?
  COLOR_REORDER = .true.,                         coloring reorder for mpi splitting?
/
```

(CONF_FILES) must correspond to (PRC_DOMAINS) in order. The above case means that the run is executed by the 4-MPI parallel for the parent domain and the 16-MPI parallel for the child domain; each number of processes in the launch file must correspond to the total number of MPI processes (PRC_NUM_X×PRC_NUM_Y) specified in each configuration file run.***.conf.

Log of MPI communicator splitting will be output, when (LOG_SPLIT) is .true.. The default value of (LOG_SPLIT) is .false..

Option of (COLOR_REORDER) is the switch for job rearrangement in MPI communicator groups following the process size of the job. The job which has the largest process size is arranged at the front row; this would be reasonable to gain the efficient inter-node communication.

At execution, the total number of MPI processes is given, which different from that in the case of single-domain execution. For example, 20 processes are specified in the above case.

```
$ mpirun  -n  [number of processes]  ./scale-rm  launch.conf
```

When multiple domain calculations are executed at the same time, the different file names must be used for input/output files among domains to avoid any confusion. For example, the configuration files prepared by the "the making tool for the complete settings of the experiment" use history_d01.pe***.nc, history_d02.pe***.nc for the file name of history output.

The calculation may sometimes abort, outputting the message below. This is the error message, meaning that the domain of computation of the child is larger than that of the parent domain. If such a message appears, retry creating the topography, the land-use data, and the initial/boundary data, and confirm again whether the configurations are correct:

```
ERROR [COMM_CARTESC_NEST_domain_relate] region of daughter domain is larger than that
 of parent
```

Guideline for Distribution of MPI Processes

As shown in Fig. 4.2.7, no MPI process is shared between the multiple domains in the online nesting experiment. In other words, each MPI process takes charge of a part of a specific domain. Therefore, the user should determine how many MPI processes to allocate to each domain. When this allocation is not appropriate, a long waiting time is incurred. To avoid this situation, it is reasonable to allocate processes so that the magnitude of time integrations for each process is as similar as possible among processes[*2]. Here, the magnitude of time integration, i.e., computational effort, is defined as the product of the number of grids and time steps.

Let us consider N-domain nesting. The total number of grids in the x, y, and z directions in n-th domain are denoted by IMAXG_n, JMAXG_n, and KMAX_n, respectively. DT_n is the time interval (TIME_DT) in the n-th domain. Using the time-step of the outermost domain (n=1) DT_1 as a benchmark, the necessary number of time steps in the n-th domain is estimated as:

$$TSTEP\_n = DT\_1/DT\_n$$

The calculation for the n-th domain is derived by multiplying the number of grids as

$$OPR\_n = IMAXG\_n \times JMAXG\_n \times KMAX\_n \times TSTEP\_n$$

[*2] More accurately, floating-point operations should be estimated.

Therefore, the standard number of processes allocated to the n-th domain (`MPI_n`) is estimated as

$$\texttt{MPI\_n} = \texttt{MPI\_total} \times \frac{\texttt{OPR\_n}}{\sum_{m=1}^{N} \texttt{OPR\_m}}, \tag{4.2.15}$$

where `MPI_total` is the total number of MPI processes.

The number of processes that are distributed along the x and y directions (`PRC_NUM_X, PRC_NUM_Y`) out of `MPI_n` can be arbitrarily decided. It is recommended to configure them so that the difference between the number of grid along X direction(`IMAX = IMAXG/PRC_NUM_X`) and that along Y direction(`JMAX = JMAXG/PRC_NUM_Y`) is as small as possible. This is because such a configuration can reduce the area of the halo. As a result, high computational performance can be obtained[*3].

In the above explanation, only the number of grids and time steps are considered. However, in actual calculations such as nesting simulation in real atmospheric experiment, the time interval for each physical process, and intra-domain and inter-domain communications affect the elapsed time. In the online nesting configuration, because the calculation in the innermost domain is largest in general, it is reasonable to distribute MPI processes so that waiting time caused by MPI communications is minimized in the innermost domain. In case of large-scale computations, long integration and many ensemble simulations, it is recommended to tune for the distribution of processes following the above rough estimation.

[*3] Note that in case of hybrid parallelization used together with thread parallelism, it is necessary to take a larger number of grids along the y-axis than the x-axis to minimize computational imbalance between threads.

Chapter 4.3

Setting the dynamical core

4.3.1 Dynamical Core for Cartesian C-grid

In this section, the dynamical core for the Cartesian C-grid is described. The Cartesian C-grid is employed in SCALE-RM . In the Cartesian C-grid, scalar quantities, such as density, thermodynamics variable and vapor, is defined at the cell center, while components of vector quantities, such as the momentums and fluxes, are defined at staggered point. See the description document of SCALE-RM for more details.

4.3.1.1 Setting Integration Numerical Method

The numerical method for time integration in the dynamical process is specified in `(ATMOS_DYN_TYPE)` in `[PARAM_ATMOS]` in the configuration file.

```
&PARAM_ATMOS
 ATMOS_DYN_TYPE = "HEVE",    ; Choose from Table 4.3.1.
/
```

Time step depends on the sound speed in the case of using the explicit method, while it does not in the case of using the implicit method. In most real atmospheric simulations, vertical grid spacing is much smaller than the horizontal ones. Thus, fully explicit scheme, that is "HEVE", requires a quite small time step, which depends on vertical grid spacing and sound speed. Therefore, "HEVI" is often used for the real atmospheric simulations.

Table 4.3.1: Options of methods for time integration in dynamical process

Scheme name	Description of scheme	Note
HEVE	Fully explicit method	
HEVI	Horizontally explicit and vertically implicit methods	Recommended for real experiment

4.3.1.2 Setting Temporal and Spatial Schemes

The temporal and spatial schemes are configured in `[PARAM_ATMOS_DYN]`. An example of setting, which is recommended for real atmospheric simulations, is shown below.

```
&PARAM_ATMOS_DYN
 ATMOS_DYN_TINTEG_SHORT_TYPE = RK4,            ; Choose from temporal schemes in Table
                                               4.3.2
 ATMOS_DYN_TINTEG_TRACER_TYPE = RK3WS2002,     ; Choose from temporal schemes
 ATMOS_DYN_FVM_FLUX_TYPE = UD3,                ; Choose from temporal spatial schemes in
                                               Table 4.3.2
 ATMOS_DYN_FVM_FLUX_TRACER_TYPE = UD3KOREN1993, ; Choose from spatial schemes
 ATMOS_DYN_FLAG_FCT_TRACER = .false.,          ; Use FCT scheme (.true.) or not (.false.)
 ATMOS_DYN_NUMERICAL_DIFF_COEF = 0.D0,
 ATMOS_DYN_NUMERICAL_DIFF_COEF_TRACER = 0.D0,
 ATMOS_DYN_wdamp_height = 15.D3,               ; height [m] of the bottom of sponge layer
                                               (for Rayleigh damping)

 /
```

The other options for temporal and spatial schemes are shown in Table 4.3.2. Note that the time step should be set according to the used schemes in order to ensure numerical stability. An criteria to determine the time step is described in Section 4.2.5.

Table 4.3.2: Setting temporal and spatial schemes

Scheme name	Description of scheme	
Temporal scheme		
RK3	Heun-type 3 stage and 3rd-order Runge–Kutta scheme	
RK3WS2002	3 stage and generraly 2nd-order Runge–Kutta scheme in Wicker and Skamarock (2002)	
RK4	4 stage and 4th-order Runge–Kutta scheme	
RK7s6o	7 stage and 6th-order Runge–Kutta scheme in Lawson (1967) (supported only for HEVE)	
RK11s8o	11 stage and 8th-order Runge–Kutta scheme in Cooper and Verner (1972) (supported only for HEVE)	
Spatial scheme		Minimum number of halos
CD2	2nd-order central flux	1
CD4	4th-order central flux	2
CD6	6th-order central flux	3
CD8	8th-order central flux	3
UD3	3rd-order upwind flux	2
UD5	5th-order upwind flux	3
UD7	7th-order upwind flux	3
UD3KOREN1993	3rd-order upwind flux + Koren (1993)'s filter	2

The default setting for advection scheme used for the prognostic variables in dynamics, spcified by (ATMOS_DYN_FVM_FLUX_TYPE)), is the 4th-order central flux (CD4) in the SCALE-RM . When using CD4 in a simulation with a steep terrain, an artificial grid-scale vertical flow is often seen at the peak of mountains. This grid-scale flow may be reduced by using UD3. So, the use of UD3 is recommended for experiments with steep terrains.

4.3.1.3 Numerical Diffusions

The numerical stability depends on schemes for the dynamical process used in simulations (Sec. 4.3.1.2). The stability would be improved by applying numerical diffusion. SCALE-RM has the hyper-diffusion and divergence dumping as the explicit numerical diffusion.

The setting for them is the following:

```
&PARAM_ATMOS_DYN
ATMOS_DYN_NUMERICAL_DIFF_LAPLACIAN_NUM = 2,
ATMOS_DYN_NUMERICAL_DIFF_COEF = 1.D-4,
ATMOS_DYN_NUMERICAL_DIFF_COEF_TRACER = 0.D0,
ATMOS_DYN_DIVDMP_COEF = 0.D0,
/
```

The hyper-diffusion reduces the high frequency component of a target variable; it is mainly used to remove numerical noise. The hyper-diffusion of a variable ϕ is defined as

$$\nu \Delta^n \rho(\phi - \phi_0), \tag{4.3.1}$$

where ν is diffusion coefficient, ϕ_0 is the reference state (See Section 4.3.1.7), and Δ is the Laplacian operator,

$$\Delta^n = \nabla^{2n} = \frac{\partial^{2n}}{\partial x^{2n}} + \frac{\partial^{2n}}{\partial y^{2n}} + \frac{\partial^{2n}}{\partial z^{2n}}. \tag{4.3.2}$$

The order of the Laplacian operator is specified by
(ATMOS_DYN_NUMERICAL_DIFF_LAPLACIAN_NUM). (ATMOS_DYN_NUMERICAL_DIFF_COEF) and
(ATMOS_DYN_NUMERICAL_DIFF_COEF_TRACER) is a non-dimensional coefficient of the hyper-diffusion.
(ATMOS_DYN_NUMERICAL_DIFF_COEF) is a coefficient for the dynamical prognostic variables, such as density, momentum and potential temperature,
and (ATMOS_DYN_NUMERICAL_DIFF_COEF_TRACER) is a coefficient for the tracer variables, such as specific humidity, hydrometeors, and turbulent kinetic energy. The two-grid scale noise is dumped to $1/e$ in a one time step if the coefficient is unity. The dumping is stronger for a larger coefficient. The hyper-diffusion itself may cause numerical instability if the coefficient is larger than 1.
(ATMOS_DYN_NUMERICAL_DIFF_COEF) can be set to zero when using the upwind schemes, such as UD3, UD5, because they have implicit numerical diffusion.

The divergence dumping can also be available to improve numerical stability. The divergence dumping reduces the three-dimensional divergence component; it is mainly used to remove sound waves. Its coefficient can be set by (ATMOS_DYN_DIVDMP_COEF).

4.3.1.4 Positive Definite

For tracer advection, guaranteeing a non-negative value is required in most cases.
The UD3KOREN1993 scheme guarantees a non-negative value, whereas other schemes do not. When schemes other than UD3KOREN1993 are used the FCT filter can be used to guarantee the non-negative value. The advection scheme is specified by (ATMOS_DYN_FVM_FLUX_TRACER_TYPE), and switch for the FCT filter is (ATMOS_DYN_FLAG_FCT_TRACER)=.true..

4.3.1.5 Halos

The necessary number of halos grid depends on the spatial difference scheme as shown in Table 4.3.2. Set (IHALO) and (JHALO) in [PARAM_ATMOS_GRID_CARTESC_INDEX] for the number of halos grid for the x- and y-directions, respectively. By default, the number of the grid is 2, which is suitable for "UD3", "UD3KOREN1993", and "CD4". For example, the configuration of the halo for the fifth-order upwind difference scheme is as follows:

```
&PARAM_ATMOS_GRID_CARTESC_INDEX
 IHALO = 3,
 JHALO = 3,
/
```

4.3.1.6 Setting for Coriolis Force

In this subsection, treatments of the Coriolis force in SCALE-RM is explained. The Coriolis parameter is zero as the default, so that you have to set (some) parameter(s) to introduce the Coriolis force in the simulation. There are two types of setting for the Coriolis parameter: f-/β-plane and sphere. The type can be specified by (CORIOLIS_type) in [PARAM_CORIOLIS].

f-/β-plane

If (CORIOLIS_type) is set to "PLANE", the Coriolis parameter f is $f = f_0 + \beta(y - y_0)$. When $f_0 = 0$ and $\beta = 0$, which is default, no Coriolis force is taken into account.

The plane for $\beta = 0$ is called f-plane, otherwise it is called β-plane. The parameters of f_0, β and y_0 is set with the parameters of [PARAM_CORIOLIS] as follows:

```
&PARAM_CORIOLIS
  CORIOLIS_type = 'PLANE', ! PLANE or SPHERE
  CORIOLIS_f0 = 1.0D-5, ! f_0
  CORIOLIS_beta = 0.0D0, ! β
  CORIOLIS_y0 = 0.0D0, ! y_0
/
```

The default values of the (CORIOLIS_f0), (CORIOLIS_beta), and (CORIOLIS_y0) are 0.0, 0.0, and y at the domain center, respectively.

If you want to add the geostrophic pressure gradient force that is in balance with the Coriolis force accompanied by the geostrophic wind, you need to modify the user specific file mod_user.f90 (see Section 5.2.1). The test case of scale-rm/test/case/inertial_oscillation/20km is an example of a simulation on the f-plane with the geostrophic pressure gradient force.

Sphere

On the sphere, the Coriolis parameter depends on the latitude as $f = 2\Omega \sin(\phi)$, where Ω and ϕ are angular velocity of the sphere and latitude, respectively. In this case, you have to set (CORIOLIS_type) = "SPHERE". The angular velocity of the sphere is set by (CONST_OHM) parameter of [PARAM_CONST] (see Section 4.5.3). The latitude of the individual grids is determined depending on the map projection, which is explained in Section 4.2.4.

Lateral Boundary Condition for Coriolis Force

If there exists geostrophic wind, the periodic boundary conditions cannot be applied in its perpendicular direction, since the wind is not periodic. For the f-plane, the double periodic boundary conditions can be applied with no geostrophic wind. For the β-plane or sphere, the periodic boundary condition cannot be used in the y-direction, since the Coriolis parameter differs at the southern and northern boundaries. In the absent of meridional geostrophic wind, the periodic boundary in the x-direction is allowed for the all the settings (i.e., the f-plane, β-plane, and sphere).

The nudge lateral boundary conditions at the south and north boundaries might be used for f- and β-plane experiment. For the details of the nudging boundary, see Sections 4.2.3.5. The test case of scale-rm/test/case/rossby_wave/beta-plane is an example of a simulation on the β-plane with the south and north nudging boundaries.

4.3.1.7 Setting for Reference State

As explained in Section 4.3.1.3, the reference state is used in the calculation of numerical diffusion in the dynamical processes. It is also used to calculate the pressure gradients in the momentum equations. Since the reference state is defined under the hydrostatic balance, the deviation from the reference state can be used to calculate the pressure gradients.

The settings for the reference state are the following:

```
&PARAM_ATMOS_REFSTATE
ATMOS_REFSTATE_IN_BASENAME = "",              ! input file name
ATMOS_REFSTATE_OUT_BASENAME = "",             ! output file name
ATMOS_REFSTATE_OUT_TITLE = "SCALE-RM RefState, ! title in the output file
ATMOS_REFSTATE_OUT_DTYPE = "DEFAULT",         ! data type in the output file
ATMOS_REFSTATE_TYPE = "UNIFORM",              ! type of reference state
ATMOS_REFSTATE_TEMP_SFC = 300.0D0,            ! surface potential temperature
ATMOS_REFSTATE_RH = 0.0D0,                    ! relative humidity
ATMOS_REFSTATE_POTT_UNIFORM = 300.0D0,        ! potential temperature
ATMOS_REFSTATE_UPDATE_DT = -1.0D0,            ! update interval [sec]
/
```

If (ATMOS_REFSTATE_IN_BASENAME) is specified, the reference state is read from the file. Otherwise, the reference state is generated at the initial time of a simulation with the type specified by (ATMOS_REFSTATE_TYPE). It can be "ISA", "UNIFORM", "ZERO", or "INIT":

ISA The international standard atmosphere where the surface potential temperature, relative humidity, and the surface pressure are specified by (ATMOS_REFSTATE_TEMP_SFC), (ATMOS_REFSTATE_RH), and (CONST_Pstd) (See Section 4.5.3), respectively.

UNIFORM The constant potential temperature and relative humidity profile specified by (ATMOS_REFSTATE_POTT_UNIFORM) and (ATMOS_REFSTATE_RH), respectively.

ZERO The profile with all variables set to zero.

INIT The horizontally averaged initial state.

When (ATMOS_REFSTATE_TYPE) is "INIT", the reference state can be updated during the simulation. The update interval is specified by (ATMOS_REFSTATE_UPDATE_DT).
The unit of (ATMOS_REFSTATE_UPDATE_DT) is seconds. The updated reference state is the horizontally averaged value at the update time. If (ATMOS_REFSTATE_UPDATE_DT) is set to a negative value, the reference state is not updated during the simulation.

For a restart simulation, you need to pay attention to the setting of the reference state. In order for the result of a restart simulation to be the same as that of the continuous simulation, it is necessary to set so that the reference state of both are the same. If you use an "INIT"-type and update the reference state ((ATMOS_REFSTATE_UPDATE_DT) > 0), the reference states of the two simulations will be identical by setting the update interval ((ATMOS_REFSTATE_UPDATE_DT)) to a divisor of the restart interval. If you use an "INIT"-type and do not update the reference state ((ATMOS_REFSTATE_UPDATE_DT) < 0), you can use the same reference state by outputting the state to a file specified by (ATMOS_REFSTATE_OUT_BASENAME) in the run before the restart and inputting the state from the file by specifying (ATMOS_REFSTATE_IN_BASENAME) in the run after the restart.

Chapter 4.4

Setting the physical process

4.4.1 Cloud Micro-Physics

The cloud micro-physics scheme is configured in (`ATMOS_PHY_MP_TYPE`) in [`PARAM_ATMOS`] in files `init.conf` and `run.conf`, respectively. Note that it is necessary to specify the same scheme for (`ATMOS_PHY_MP_TYPE`) in the configuration files for both init and run executions. The update interval for the cloud micro-physics scheme is specified in [`PARAM_TIME`]. Refer to Section 4.2.5 for the detailed configuration of calling timing. The following example shows the configuration for cases involving a six-class one-moment bulk scheme that contains ice phase clouds:

```
&PARAM_ATMOS
  ATMOS_PHY_MP_TYPE = "TOMITA08",   ; Choose from Table 4.4.1.
/
```

Table 4.4.1: List of cloud micro-physics scheme types

Schemes	Description of scheme	Reference
OFF	Do not calculate phase change of water by cloud micro-physics.	
KESSLER	Three-class one-moment bulk scheme	Kessler (1969)
TOMITA08	Six-class one-moment bulk scheme	Tomita (2008)
SN14	Six-class two-moment bulk scheme	Seiki and Nakajima (2014)
SUZUKI10	Spectral bin scheme (consideration of ice cloud can be specified as option)	Suzuki et al. (2010)

Four typical schemes are prepared:

1. **One-moment bulk scheme without ice Kessler (1969)**
   This scheme assumes that the particle size distribution function is expressed only by mass concentration. Considering two categories of water in cloud and rain, the ratios of the densities of cloud and rain to total air density are prognostically predicted.

2. **One-moment bulk scheme with ice Tomita (2008)**
   This scheme makes the same assumption as that in Kessler (1969) for the particle size distribution function, but with five categories of water: cloud, rain, ice, snow, and graupel.

3. **Two-moment bulk scheme with ice Seiki and Nakajima (2014)**
   In this scheme, the particle size distribution is expressed by the numerical concentration of particles and their mass concentration.

4. **One-moment bin scheme Suzuki et al. (2010)**
   This scheme explicitly expresses particle size distribution by discretizing it using an appropriate number of degrees of freedom for each category. There are six categories: cloud, rain, ice, snow, graupel, and hail. The accuracy of expressing the size distribution depends on the degrees of freedom.

The degrees of sophistication increases from 1 to 4, as does computational cost.

If SUZUKI10 is selected, in addition to the specification of (ATMOS_PHY_MP_TYPE), the following configuration needs to be added to both configuration files of init and run executions:

```
&PARAM_ATMOS_PHY_MP_SUZUKI10_bin
  nbin = 33,                         ; The number of bins
  ICEFLG = 1,                        ; Option for consideration of ice cloud: 0(not considered),
                                     1(considered)
  kphase = 0,                        ; Type of collection kernel function for collision/coagulation
                                     processes: 0 is hydro-dynamic kernel, 1 is Golovin type kernel
                                     (Golovin (1963)), and 2 is Long type kernel (Long (1974)).
                                     Pleas see the description document of SCALE-RM for more
                                     details.
/
```

In this case, [PARAM_ATMOS_PHY_MP_SUZUKI10_bin] in the init configuration file must also be same as in the run configuration file. A necessary file `micpara.dat` is automatically generated. If file `micpara.dat` already exists, it is used for the calculation. When changing `nbin` as described in the first line, this file is regenerated. If `nbin` in file `run.conf` is different from that in file `micpara.dat`, the following error message is output and the simulation program is terminated instantaneously without calculation:

```
ERROR [ATMOS_PHY_MP_suzuki10_setup] nbin in inc_tracer and nbin in micpara.dat is
different check!
```

To avoid this error, it is necessary to delete the old `micpara.dat` beforehand and regenerate it. The regeneration is automatically done at the execution of SCALE-RM with SUZUKI10.

4.4.2 Cumulus Parameterization

The cumulus parameterization is specified in (ATMOS_PHY_CP_TYPE) in [PARAM_ATMOS] in files `init.conf` and `run.conf`. The timing of the calling of the cumulus parameterization is specified in [PARAM_TIME]. Refer to Section 4.2.5 for the detailed configuration of the calling timing.

```
&PARAM_ATMOS
  ATMOS_PHY_CP_TYPE = "KF",    ; Select the scheme shown in Table 4.4.2
/
```

Table 4.4.2: List of cumulus parameterization types

Schemes	Description of scheme	Reference
OFF	No use of the cumulus parameterization	
KF	Kain-Fritsch convective parameterization	Kain and Fritsch (1990); Kain (2004)

In this version of SCALE-RM , only KF is supported for cumulus parameterization. KF is a mass flux conservation type of cumulus parameterization scheme, representing one convection in

the sub-grid scale. The cumulus parameterization is recommended to be used when grid spacing is more than about 5 km to avoid calculation of unnatural strong deep convections. The cumulus parameterization and cloud micro-physics individually output precipitation named `RAIN_CP` and `RAIN_MP`, respectively. `RAIN` and `PREC` are the sum of precipitation from cumulus parameterization and cloud micro-physics schemes (i.e., `RAIN = RAIN_CP + RAIN_MP` and `PREC = RAIN_CP + RAIN_MP`). Note that `KF` calculates changes in the water vapor and hydrometeors such as cloud water and cloud ice in the atmosphere. The changes in hydrometeors is calculated further in micro physics schemes. The number concentration of cloud water, ice, and so on is not considered in `KF`. Therefore, the changes in the number concentration associated with the changes in hydrometeors in `KF` is estimated by a prescribed function and then provided to double-moment micro-physics schemes.

Configuration for `Kain-Fritsch` scheme

Following tuning parameters are available in `KF`:

```
&PARAM_ATMOS_PHY_CP_KF
  ATMOS_PHY_CP_kf_trigger_type = 1,    ; Trigger function type: 1=Kain, 3=Narita-Ohmori
  ATMOS_PHY_CP_kf_dlcape = 0.1,        ; Cape decrease rate
  ATMOS_PHY_CP_kf_dlifetime = 1800,    ; Lifetime scale of deep convection [sec]
  ATMOS_PHY_CP_kf_slifetime = 2400,    ; Lifetime scale of shallow convection [sec]
  ATMOS_PHY_CP_kf_DEPTH_USL = 300,     ; Depth of updraft source layer at the beginning of search [hPa]
  ATMOS_PHY_CP_kf_prec_type = 1,       ; Precipitation type: 1=Ogura-Cho, 2=Kessler
  ATMOS_PHY_CP_kf_rate = 0.03,         ; Ratio of cloud water and precipitation for Ogura-Cho precipitation function
  ATMOS_PHY_CP_kf_thres = 1.E-3,       ; Autoconversion rate for Kessler precipitation function
  ATMOS_PHY_CP_kf_LOG = false,         ; Output warning messages
/
```

Users can select trigger function from two options:

1. Kain Type Kain (2004)
   Default trigger function in SCALE-RM.

2. Narita and Ohmori Type Narita and Ohmori (2007)
   A trigger function which seems to be more suitable for Japan area.

Users can select precipitation function from two options:

1. Ogura-Cho Type Ogura and Cho (1973)
   Default precipitation function in SCALE-RM. For this type, additional tuning parameter of `(ATMOS_PHY_CP_kf_rate)` is available.

2. Kessler Type Kessler (1969)
   Kessler type simple precipitation function. For this type, additional tuning parameter of `(ATMOS_PHY_CP_kf_thres)` is available.

The time interval of calling KF specified by `(TIME_DT_ATMOS_PHY_CP)` in `[PARAM_TIME]` is also a tuning parameter, which is effective to the precipitation amount. As the first setting of `(TIME_DT_ATMOS_PHY_CP)`, 300s is recommended. KF will output a warning message of "go off top/bottom of updraft source layer", if `(PALAM_ATMOS_PHY_CP_kf_LOG)` is true. The updraft source layer should be thicker than the threshold layer depth (default is 50 hPa). However, the calculation is not aborted for these conditions.

4.4.3 Sub-Grid Scale Turbulence Scheme

The sub-grid scale turbulence model is for representing energy cascade to the sub-grid scale due to the advection terms in Large-eddy simulations (LESs).

The turbulence scheme is specified in `(ATMOS_PHY_TB_TYPE)` in `[PARAM_ATMOS]` in files `init.conf` and `run.conf`. The timing of the calling of the turbulence scheme is specified in `[PARAM_TIME]`. Refer to Section 4.2.5 for the detailed configuration of the calling timing.

```
&PARAM_ATMOS
 ATMOS_PHY_TB_TYPE = "SMAGORINSKY",    ; Select the scheme shown in Table 4.4.3
/
```

Table 4.4.3: List of turbulence scheme types

Schemes	Description of scheme	Reference
OFF	Do not calculate the turbulence process	
SMAGORINSKY	Smagorinsky—Lilly-type sub-grid model	Smagorinsky (1963); Lilly (1962); Brown et al. (1994); Scotti et al. (1993)
D1980	Deardorff-type sub-grid model	Deardorff (1980)

The `SMAGORINSKY` scheme can be also used for the horizontal eddy viscosity in RANS simulations. The planetary boundary layer parameterization (Section 4.4.4) is a scheme considering only vertical mixing. If you want to consider the horizontal eddy viscosity in RANS simulations, use the sub-grid scale turbulence model for horizontal mixing. When using the sub-grid scale turbulence model in conjunction with planetary boundary layer parameterization, the vertical eddy viscosity of the `SMAGORINSKY` scheme is disabled.

4.4.4 Planetary Boundary Layer Scheme

The planetary boundary layer (PBL) parameterization is a scheme for vertical mixing by turbulence in the PBL. It is for Reynolds-Averaged Navier-Stokes equations (RANS) simulations.

The planetary boundary layer parameterization scheme is specified in `(ATMOS_PHY_BL_TYPE)` in `[PARAM_ATMOS]` in files `init.conf` and `run.conf`. The timing of the calling of PBL scheme is specified in `[PARAM_TIME]`. Refer to Section 4.2.5 for the detailed configuration of the calling timing.

```
&PARAM_ATMOS
 ATMOS_PHY_BL_TYPE = "MYNN",    ; Select the scheme shown in Table 4.4.4
/
```

Table 4.4.4: List of planetary boundary layer scheme types

Schemes	Description of scheme	Reference
OFF	Do not calculate the PBL process	
MYNN	MYNN level 2.5 boundary scheme	Mellor and Yamada (1982); Nakanishi and Niino (2004, 2009)

The PLB scheme calculate only vertical mixing. If you want to consider the horizontal eddy viscosity in RANS simulations, a sub-grid scale turbulence model can be used for the horizontal mixing. See Section 4.4.3 for the horizontal mixing.

4.4.5 Radiation scheme

The radiation scheme is specified in (ATMOS_PHY_RD_TYPE) in [PARAM_ATMOS] in files `init.conf` and `run.conf`. The timing of the calling of the radiation scheme is specified in [PARAM_TIME]. Refer to Section 4.2.5 for the detailed configuration of calling timing.

```
&PARAM_ATMOS
 ATMOS_PHY_RD_TYPE = "MSTRNX",    ; Select the radiation scheme shown in Table 4.4.5
 /
```

Table 4.4.5: List of radiation scheme types

Schemes	Explanation of scheme	Reference
OFF or NONE	Do not calculate the radiation process	
OFFLINE	Use prescribed radiative data given from a file	
MSTRNX	mstrnX (A k-distribution-based broad-band radiation transfer model)	Sekiguchi and Nakajima (2008)

Configuration for OFFLINE

When (ATMOS_PHY_RD_TYPE) is OFFLINE in [PARAM_ATMOS], the file name and information of the data are specified in [PARAM_ATMOS_PHY_RD_OFFLINE].

```
&PARAM_ATMOS_PHY_RD_OFFLINE
 ATMOS_PHY_RD_OFFLINE_BASENAME = "",                   ; base name of external data file
 ATMOS_PHY_RD_OFFLINE_AXISTYPE = "XYZ",                ; order of spatial dimensions of the
                                                         data. 'XYZ' or 'ZXY'
 ATMOS_PHY_RD_OFFLINE_ENABLE_PERIODIC_YEAR = .false.,  ; whether annually cyclic data
 ATMOS_PHY_RD_OFFLINE_ENABLE_PERIODIC_MONTH = .false., ; whether monthly cyclic data
 ATMOS_PHY_RD_OFFLINE_ENABLE_PERIODIC_DAY = .false.,   ; whether daily cyclic data
 ATMOS_PHY_RD_OFFLINE_STEP_FIXED = 0,                  ; step number when data at a cer-
                                                         tain time step is used.  Set the
                                                         value less than 1 for temporal var-
                                                         ied data.
 ATMOS_PHY_RD_OFFLINE_CHECK_COORDINATES = .true.,      ; whether coordinate variables are
                                                         to be checked
 ATMOS_PHY_RD_OFFLINE_STEP_LIMIT = 0,                  ; maximum limit of steps.   The
                                                         data at the time step exceed this
                                                         limit would not be read.  0 for no
                                                         limit
 ATMOS_PHY_RD_OFFLINE_DIFFUSE_RATE = 0.5D0,            ; diffuse rate (diffuse solar ra-
                                                         diation/global solar radiation) of
                                                         short wave used when short-wave
                                                         direct flux data is not given
 /
```

The file format of external data file is netCDF format with the same coordinate variables as that of initial/boundary data files. Variables required as the external data are shown in Table 4.4.6.

Table 4.4.6: Radiative data as external file input

Variable name	Description	# of dimensions
RFLX_LW_up	Upward long-wave radiative flux	3D (spatial) + 1D (time)
RFLX_LW_dn	Downward long-wave radiative flux	3D (spatial) + 1D (time)
RFLX_SW_up	Upward short-wave radiative flux	3D (spatial) + 1D (time)
RFLX_SW_dn	Downward short-wave radiative flux	3D (spatial) + 1D (time)
SFLX_LW_up	Upward long-wave radiative flux at the surface	2D (spatial) + 1D (time)
SFLX_LW_dn	Downward long-wave radiative flux at the surface	2D (spatial) + 1D (time)
SFLX_SW_up	Upward short-wave radiative flux at the surface	2D (spatial) + 1D (time)
SFLX_SW_dn	Downward short-wave radiative flux at the surface	2D (spatial) + 1D (time)
SFLX_SW_dn_dir	Downward short-wave direct radiative flux at the surface	2D (spatial) + 1D (time) optional

Configuration for MSTRNX

The solar radiation is calculated by using date, time, longitude, and latitude according to the model configuration of calculation. For the ideal experiment, they can be arbitrarily given as fixed values of time and location over the domain. The solar constant can also be changed. These are configured in [PARAM_ATMOS_SOLARINS] as follows:

```
&PARAM_ATMOS_SOLARINS
  ATMOS_SOLARINS_constant = 1360.250117      Solar constant [W/m2]
  ATMOS_SOLARINS_set_ve = .false.            Whether settings of the ideal vernal equinox are
                                             used
  ATMOS_SOLARINS_set_ideal = .false.         Whether values are fixed for obliquity and eccen-
                                             tricity
  ATMOS_SOLARINS_obliquity = 0.0             obliquity   [deg.]      in    the    case    that
                                             ATMOS_SOLARINS_set_ideal=.true.
  ATMOS_SOLARINS_eccentricity = 0.0          eccentricity    in     the     case     that
                                             ATMOS_SOLARINS_set_ideal=.true.
  ATMOS_SOLARINS_fixedlatlon = .false.       Whether values are fixed for latitude and longi-
                                             tude at radiation calculation
  ATMOS_SOLARINS_lon = 135.221               Longitude   [deg.]      in    the    case    that
                                             ATMOS_SOLARINS_fixedlatlon=.true.
  ATMOS_SOLARINS_lat = 34.653                Latitude    [deg.]      in    the    case    that
                                             ATMOS_SOLARINS_fixedlatlon=.true.
  ATMOS_SOLARINS_fixeddate = .false.         Whether values are fixed for date and time at ra-
                                             diation calculation
  ATMOS_SOLARINS_date = -1,-1,-1,-1,-1,-1,   Date and time [Y,M,D,H,M,S] in the case that
                                             ATMOS_SOLARINS_fixeddate=.true.
/
```

The position of the earth on the orbital period is calculated based on the vernal equinox. However, the Gregorian calendar has leap years, and the vernal equinox days are not the same each year. Therefore, we set 7:35 on March 20, 2000, as the reference time to match the calendar with the vernal equinox passing point. We set the solar year to 365.2425 days. Since this value is larger than the average solar year of 365.24219, it causes a deviation of 3 hours in the revolution orbit in about 400 years, but it is almost negligible. When (ATMOS_SOLARINS_set_ideal)

is `.true.`, the solar insolation is calculated using obliquity(deg.) and eccentricity specified at `(ATMOS_SOLARINS_obliquity)`,
`(ATMOS_SOLARINS_eccentricity)`, respectively. These configurations are useful for the ideal simulations and the simulations of planets other than the Earth. When `(ATMOS_SOLARINS_fixedlatlon)` is `.true.`, the solar insolation is calculated using same longitude and latitude specified at `(ATMOS_SOLARINS_lon, ATMOS_SOLARINS_lat)` over the whole domain. The default values of them are `(MAPPROJECTION_basepoint_lon, MAPPROJECTION_basepoint_lat)`, which are set in `[PARAM_MAPPROJECTION]`. Refer to Section 4.2.4 for the explanation of `[PARAM_MAPPROJECTION]`. When `(ATMOS_SOLARINS_fixeddate)` is `.true.`, the solar insolation is calculated according to date and time (Y,M,D,H,M,S) specified at `(ATMOS_SOLARINS_date)`. The date and time are not fixed if the negative values are specified. For example, when configuring `(ATMOS_SOLARINS_date)` to 2000,3,20,-1,-1,-1, the date is fixed to March 20, 2000 (the vernal equinox) and diurnal variations of the solar insolation are considered in the simulation. When `(ATMOS_SOLARINS_set_ve)` is `.true.`, the multiple configurations of ideal vernal equinox is automatically set. This option sets obliquity and eccentricity to zero, both longitude and latitude to zero deg., and the date and time to 12UTC, March 20, 2000. If the values are specified by using `(ATMOS_SOLARINS_set_ideal, ATMOS_SOLARINS_fixedlatlon, ATMOS_SOLARINS_fixeddate)` as described above, those settings are given priority.

Depending on the experimental design, the top of the model is often too low, such as $10 \sim 20$ km, compared to the height of the atmosphere. To remedy this situation, another top height used only for radiation calculation is set. The top height for radiation depends on the parameter file of the radiation scheme. For example, when `MSTRNX` is used, the default parameter table used for `MSTRNX` assumes that it is 100 km. For the calculation of radiation at levels higher than the top of the model, several layers are prepared. The additional layers are 10 by default; If the top of the model is 22 km, 10 additional layers with a grid spacing of 7.8 km are added for radiation calculation. These are configured in `[PARAM_ATMOS_PHY_RD_MSTRN]`.

`MSTRNX` requires a parameter table for radiation calculation. By default, the wavelength between solar radiation and infrared radiation is divided into 29 bands/111 channels; nine types of cloud and aerosol particles with eight particle bins are prepared in the table. Three kinds of parameter files are prepared in the directory `scale-rm/test/data/rad/`.

```
scale-rm/test/data/rad/PARAG.29      ; absorption parameter for gas
scale-rm/test/data/rad/PARAPC.29     ; absorption and scattering param. for particles
scale-rm/test/data/rad/VARDATA.RM29  ; particle parameter for cloud and aerosol
```

These files are specified in `[PARAM_ATMOS_PHY_RD_MSTRN]` as follows:

```
&PARAM_ATMOS_PHY_RD_MSTRN
  ATMOS_PHY_RD_MSTRN_KADD = 10                              Number of layers between
                                                           model top and TOA for radi-
                                                           ation

  ATMOS_PHY_RD_MSTRN_TOA = 100.0                            Height of TOA for radiation
                                                           [km] (depending on parameter
                                                           file used)

  ATMOS_PHY_RD_MSTRN_nband = 29                             Number of bins for wave-
                                                           length (depending on param-
                                                           eter file used)

  ATMOS_PHY_RD_MSTRN_nptype = 9                             Number of aerosol species
                                                           (depending on parameter file
                                                           used)

  ATMOS_PHY_RD_MSTRN_nradius = 8                            Number of particle bins for
                                                           aerosol (depending on param-
                                                           eter file used)

  ATMOS_PHY_RD_MSTRN_GASPARA_IN_FILENAME = "PARAG.29"       Input file for absorption pa-
                                                           rameter by gas

  ATMOS_PHY_RD_MSTRN_AEROPARA_IN_FILENAME = "PARAPC.29"     Input file for absorption and
                                                           scattering parameter by cloud
                                                           and aerosol

  ATMOS_PHY_RD_MSTRN_HYGROPARA_IN_FILENAME = "VARDATA.RM29" Input file for particle parame-
                                                           ter of cloud and aerosol
  ATMOS_PHY_RD_MSTRN_ONLY_QCI = .false.                     Whether only cloud water and
                                                           ice are considered (rain, snow,
                                                           graupel are ignored)


/
```

The parameter files were updated in version 5.2. It is recommended to use new parameter files in the latest version of SCALE-RM . The previous parameter files, provided in version 5.1 or earlier, are located in the directory scale-rm/test/data/rad/OpenCLASTR. The number of particle type and the number of particle bins are different from those in the new parameter files. So if you want to use these files, (ATMOS_PHY_RD_MSTRN_nptype, ATMOS_PHY_RD_MSTRN_nradius) should be specified in
[PARAM_ATMOS_PHY_RD_MSTRN] as follows.

```
  ATMOS_PHY_RD_MSTRN_nptype = 11
  ATMOS_PHY_RD_MSTRN_nradius = 6
```

It is necessary to provide vertical profiles of temperature, pressure, and gas concentration, such as carbon dioxide and ozone, in additional layers for radiation calculation. There are two methods for this. The profiles are input as climatologies or prepared by users in ASCII format.

In the case of providing climatologies, SCALE-RM provides the database form CIRA86[*1)] (Committee on Space Research; NASA National Space Science Data Center, 2006) for temperature and pressure, and MIPAS2001 (Remedios et al., 2007) for gas species. The climatology profiles are calculated from the databases according to date, time, latitude, and longitude. If the fixed date and location are specified in [PARAM_ATMOS_SOLARINS], the calculation of profile follows those settings. The input files are also provided in the directory scale-rm/test/data/rad/.

```
  scale-rm/test/data/rad/cira.nc       ; CIRA86 data (NetCDF format)
  scale-rm/test/data/rad/MIPAS/day.atm ; MIPAS2011 data for mid-lat. (ASCII format)
```

[*1)] http://catalogue.ceda.ac.uk/uuid/4996e5b2f53ce0b1f2072adadaeda262

```
scale-rm/test/data/rad/MIPAS/equ.atm ;   for tropics (ASCII format)
scale-rm/test/data/rad/MIPAS/sum.atm ;   for summer-side high-lat. (ASCII format)
scale-rm/test/data/rad/MIPAS/win.atm ;   for winter-side high-lat. (ASCII format)
```

The file and directory names are specified in [PARAM_ATMOS_PHY_RD_PROFILE]. The example of configuration of putting five files above to the current directory is as follows:

```
&PARAM_ATMOS_PHY_RD_PROFILE
 ATMOS_PHY_RD_PROFILE_use_climatology = .true.        Whether climatologies of CIRA86
                                                      and MIPAS2001 are used
 ATMOS_PHY_RD_PROFILE_CIRA86_IN_FILENAME = "cira.nc"  File name of CIRA86
 ATMOS_PHY_RD_PROFILE_MIPAS2001_IN_BASENAME = "."     Directory name which contains
                                                      MIPAS2001 files
/
```

Gases considered in radiation calculation are water vapor(H_2O), carbon dioxide(CO_2), ozone(O_3), Nauru's oxide(N_2O), carbon monoxide(CO), methane(CH_4), oxigen(O_2), and chlorofluorocarbons(CFCs). These concentrations are able to be set to zero in [PARAM_ATMOS_PHY_RD_PROFILE].

```
&PARAM_ATMOS_PHY_RD_PROFILE
 ATMOS_PHY_RD_PROFILE_USE_H2O = .true.     When false, H2O concentration is zero.
 ATMOS_PHY_RD_PROFILE_USE_CO2 = .true.     When false, CO2 concentration is zero.
 ATMOS_PHY_RD_PROFILE_USE_O3 = .true.      When false, O3 concentration is zero.
 ATMOS_PHY_RD_PROFILE_USE_N2O = .true.     When false, N2O concentration is zero.
 ATMOS_PHY_RD_PROFILE_USE_CO = .true.      When false, CO concentration is zero.
 ATMOS_PHY_RD_PROFILE_USE_CH4 = .true.     When false, CH4 concentration is zero.
 ATMOS_PHY_RD_PROFILE_USE_O2 = .true.      When false, O2 concentration is zero.
 ATMOS_PHY_RD_PROFILE_USE_CFC = .true.     When false, CFC concentration is zero.
/
```

In case of using user-defined profiles, users must prepare height [m], pressure [Pa], temperature [K], water vapor [kg/kg], and ozone concentration [kg/kg] in ASCII format. The concentrations of gases other than water vapor and ozone are set to zero, and temporal variation is not considered. An example of user-defined files is provided in

```
scale-rm/test/data/rad/rad_o3_profs.txt
```

To use the user-defined profiles, it is required to set (ATMOS_PHY_RD_PROFILE_use_climatology) = .false. in [PARAM_ATMOS_PHY_RD_PROFILE], and to specify the file and directory names in (ATMOS_PHY_RD_PROFILE_USER_IN_FILENAME).

```
&PARAM_ATMOS_PHY_RD_PROFILE
 ATMOS_PHY_RD_PROFILE_use_climatology = .false.   Whether climatologies of CIRA86 and MI-
                                                  PAS2001 are used
 ATMOS_PHY_RD_PROFILE_USER_IN_FILENAME = ""       User-defined file in the case of not using
                                                  climatologies (ASCII format)
/
```

The number of layers and their heights in this user-defined file can be given independently of the default model configuration. During execution, the values in the model layers are interpolated

from the given profiles. Note that if the top height considered TOA in the radiation calculation is higher than in the input profile, an extrapolation is adopted there.

4.4.6 Surface flux scheme

The surface flux scheme calculates momentum, heat and moisture fluxes at the bottom atmospheric boundary. Type of the scheme is configured in (ATMOS_PHY_SF_TYPE) in [PARAM_ATMOS].

```
&PARAM_ATMOS
 ATMOS_PHY_SF_TYPE = "NONE",    ; Select the surface flux scheme shown in Table 4.4.7
/
```

If ocean, land, and urban models are used, (ATMOS_PHY_SF_TYPE) must be COUPLE or NONE; NONE is ultimately automatically replaced to COUPLE. Using these models, the surface fluxes are individually calculated in these models, and then, the merged grid value is calculated as area-weighted average of them.

Table 4.4.7: List for the atmospheric bottom boundary types

Schemes	Description of scheme
NONE	Do not calculate surface flux, but NONE is replaced to COUPLE according to settings of ocean, land, and urban models
OFF	Do not calculate surface flux
CONST	Use constant values in calculation of surface flux
BULK	Calculate the surface flux by a bulk model
COUPLE	Merge surface fluxes from ocean, land, and urban models

The time step to call the surface flux scheme is configured in (PARAM_DT_ATMOS_PHY_SF) and (PARAM_DT_ATMOS_PHY_SF_UNIT) in [PARAM_TIME]. Refer to Section 4.2.5 for the detailed configuration of the time step.

Constant scheme

If (ATMOS_PHY_SF_TYPE) is CONST, the surface flux is determined as configured in [PARAM_ATMOS_PHY_SF_CONST].

```
&PARAM_ATMOS_PHY_SF_CONST
 ATMOS_PHY_SF_FLG_MOM_FLUX = 0          ; 0: Bulk coefficient is constant
                                        ; 1: Friction velocity is constant
 ATMOS_PHY_SF_U_minM = 0.0E0            ; Lower limit of absolute velocity [m/s]
 ATMOS_PHY_SF_Const_Cm = 0.0011E0       ; Constant bulk coefficient for momentum
                                        ; (valid if ATMOS_PHY_SF_FLG_MOM_FLUX = 0)
 ATMOS_PHY_SF_Cm_min = 1.0E-5           ; Lower limit of bulk coefficient for momentum
                                        ; (valid if ATMOS_PHY_SF_FLG_MOM_FLUX = 1)
 ATMOS_PHY_SF_Const_Ustar = 0.25E0      ; Constant friction velocity [m/s]
                                        ; (valid if ATMOS_PHY_SF_FLG_MOM_FLUX = 1)
 ATMOS_PHY_SF_Const_SH = 15.E0          ; Constant sensible heat flux at the surface [W/m^2]
 ATMOS_PHY_SF_FLG_SH_DIURNAL = .false.  ; Whether diurnal variation is considered for sensible
                                        ; heat flux [logical]
 ATMOS_PHY_SF_Const_FREQ = 24.E0        ; Daily cycle if diurnal variation is enabled [hour]
 ATMOS_PHY_SF_Const_LH = 115.E0         ; Constant latent heat flux at the surface [W/m^2]
/
```

If (ATMOS_PHY_SF_FLAG_MOM_FLUX) is 0, a constant bulk coefficient is used to calculate the surface momentum flux; its value is specified by (ATMOS_PHY_SF_Const_Cm).

If (`ATMOS_PHY_SF_FLAG_MOM_FLUX`) is 1, a constant friction velocity is used to calculate the bulk coefficient; its value is specified by (`ATMOS_PHY_SF_Const_Ustar`). In the constant friction velocity case, the lower limit of the coefficient is set by (`ATMOS_PHY_SF_Cm_min`). The lower limit of absolute velocity at the lowermost layer used in the calculation is set by (`ATMOS_PHY_SF_U_minM`) in both cases.

If (`ATMOS_PHY_SF_FLG_SH_DIURNAL`) = `.false.`, sensible heat flux is constant; its value is specified by (`ATMOS_PHY_SF_Const_SH`). If (`ATMOS_PHY_SF_FLG_SH_DIURNAL`) = `.true.`, the sensible heat flux has temporal sinusoidal variation whose amplitude and time period are (`ATMOS_PHY_SF_Const_SH`) and (`ATMOS_PHY_SF_Const_FREQ`), respectively, as `ATMOS_PHY_SF_Const_SH` $\times \sin(2\pi t/3600/$`ATMOS_PHY_SF_Const_FREQ`$)$, where t is the integrated time [sec] from start of calculation.

Latent heat flux is constant and its value is specified by (`ATMOS_PHY_SF_Const_LH`).

Bulk scheme

If (`ATMOS_PHY_SF_TYPE`) is `BULK`, the surface flux is calculated by the bulk model using the pre-scribed surface state, such as surface temperature, roughness length, and albedo. The values of the surface state are read from the initial data file. The surface temperature, roughness length and albedos for short- and long-wave radiation can be set respectively with (`ATMOS_PHY_SF_DEFAULT_SFC_TEMP`), (`ATMOS_PHY_SF_DEFAULT_SFC_Z0`), (`ATMOS_PHY_SF_DEFAULT_SFC_ALBEDO_SW`), and (`ATMOS_PHY_SF_DEFAULT_SFC_ALBEDO_LW`) in [`PARAM_ATMOS_PHY_SF_VARS`] at generating the initial data with the `scale-rm_init`.

For calculation of latent heat flux, evaporation efficiency can be given in a range of 0 to 1. This flexibility enables the ideal experiment not only for ocean surfaces but also for land. If this value is 0, the surface is assumed to be completely dry, resulting no latent heat flux. If the value is 1, it is assumed to be completely wet like ocean surface. The evaporation efficiency is specified in (`ATMOS_PHY_SF_BULK_beta`) in [`PARAM_ATMOS_PHY_SF_BULK`].

```
&PARAM_ATMOS_PHY_SF_BULK
 ATMOS_PHY_SF_BULK_beta = 1.0,    ; Evaporation efficiency (range of 0–1)
/
```

Bulk coefficient

If (`ATMOS_PHY_SF_TYPE`) is `BULK` or `COUPLE`, the bulk coefficients are calculated based on the Monin-Obukhov similarity. The parameters for calculation of the coefficients are configured in [`PARAM_BULKFLUX`].

```
&PARAM_BULKFLUX
 BULKFLUX_TYPE = "B91W01",            ; Select the bulk coefficient scheme shown in Table 4.4.8
 BULKFLUX_Uabs_min = 1.0D-2,          ; Minimum absolute velocity speed in the surface flux
                                        calculation [m/s]
 BULKFLUX_surfdiag_neutral = .true.,  ; Switch to assume neutral condition in calculation of
                                        surface diagnostics
 BULKFLUX_WSCF = 1.2D0,               ; Empirical scaling factor for w* (Beljaars, 1994)
 BULKFLUX_Wstar_min = 1.0E-4,         ; Minimum value of w* [m/s]
 BULKFLUX_NK2018 = .true.,            ; Switch to use scheme of Nishizawa and Kitamura
                                        (2018)
/
```

The scheme to calculate the bulk coefficients is specified by (`BULKFLUX_TYPE`). The supported schemes are listed in Table 4.4.8.

The coefficients strongly depend on the wind speed. Because the velocity in simulation does not contain sub-grid scale component, the surface flux may be underestimated especially under calm

Table 4.4.8: List of bulk coefficient scheme types

Schemes	Description of scheme	Reference
B91W01	Bulk method by the universal function (Default)	Beljaars and Holtslag (1991); Wilson (2001); Nishizawa and Kitamura (2018)
U95	Louis-type bulk method (improved version of Louis (1979))	Uno et al. (1995)

condition. Therefore, a lower limit can be introduced for the absolute velocity at the lowermost layer used in the calculation. The lower limit is specified by (BULKFLUX_Uabs_min).

The surface diagnostic variables, such as 10-m velocities and 2-m temperature and humidity, are calculated so that they are consistent with the bulk coefficients. It means that these variables significantly depend on the surface conditions and static stability. Thus, when grid cells having different surface conditions are adjacently distributed such as near coasts, the values of the diagnostics may have a large gap at the boundaries between these grid cells. In addition, when the static stability greatly changes around sunrise or sunset, the values of the diagnostics may change drastically in a short time. These large gaps can be reduced by assuming neutral static stability in the calculation of the surface diagnostic variables. To use the assumption of neutral stability, set (BULKFLUX_surfdiag_neutral) to .true..

For the B91W01 scheme, a couple of additional parameters are available. (BULKFLUX_WSCF) is an empirical scaling factor for the free convection velocity scale w^* introduced by Beljaars (1994) (β in his paper). By default, it is determined depending on the horizontal grid spacing Δx as $1.2 \times \min(\Delta x/1000, 1)$. In simulations with the grid spacing larger than 1 km, the factor is 1.2. For smaller grid spacing, it decreases in proportional to the grid spacing, c.f., $\lim_{\Delta x \to 0} w^* = 0$. Note that, the minimum of w^* is specified by (BULKFLUX_Wstar_min) and its default value is 1.0E-4. (BULKFLUX_NK2018) is a switch to use the more appropriate formulation of the similarity for finite volume models proposed by Nishizawa and Kitamura (2018).

4.4.7 Ocean Model

The ocean process consists of two main parts, i.e., the update of the state of the near-surface ocean and the calculation of the flux at the interface of atmosphere and ocean. The timing of calling these scheme is configured in [PARAM_TIME]. Refer to Section 4.2.5 for the detailed configuration of the calling timing.

The schemes of the ocean submodel is configured by the parameter (OCEAN_DYN_TYPE), (OCEAN_SFC_TYPE), (OCEAN_ICE_TYPE), (OCEAN_ALB_TYPE), and (OCEAN_RGN_TYPE) in [PARAM_OCEAN] in the configuration files:

```
&PARAM_OCEAN
  OCEAN_DYN_TYPE = "SLAB",          ; Select the ocean dynamics type shown in Table 4.4.9
  OCEAN_SFC_TYPE = "FIXED-TEMP",    ; Select the ocean surface type shown in Table 4.4.10
  OCEAN_ICE_TYPE = "SIMPLE",        ; Select the ocean ice physics type shown in Table 4.4.11
  OCEAN_ALB_TYPE = "NAKAJIMA00",    ; Select the ocean albedo type shown in Table 4.4.12
  OCEAN_RGN_TYPE = "MOON07",        ; Select the ocean roughness type shown in Table 4.4.13
/
```

Setting vertical grids for ocean model

The number of vertical layers for ocean model is specified by (OKMAX) in [PARAM_OCEAN_GRID_CARTESC_INDEX]. The vertical grid intervals are specified by (ODZ) in [PARAM_OCEAN_GRID_CARTESC]. The unit is

Table 4.4.9: List of subsurface ocean scheme types (`OCEAN_DYN_TYPE`).

Schemes	Description of scheme
NONE or OFF	Disable the ocean submodel
INIT	Fixed to the initial condition
OFFILNE	Update by external data
SLAB	Slab ocean model

Table 4.4.10: List of ocean surface scheme types (`OCEAN_SFC_TYPE`). There is one option in current version.

Schemes	Description of scheme
FIXED-TEMP	Calculate flux without diagnosing surface skin temperature

[m].

```
&PARAM_OCEAN_GRID_CARTESC_INDEX
 OKMAX = 1,                          ; number of vertical layers
/

&PARAM_OCEAN_GRID_CARTESC
 ODZ = 10.D0,                        ; grid interval along the vertical direction
/
```

You can specify an array in `(ODZ)` for the number of vertical layers specified by `(OKMAX)`. The order of the array is from the sea surface to the deep ocean.

4.4.7.1 Fixed to initial condition

When `(OCEAN_DYN_TYPE)` is `INIT`, the ocean condition is fixed to the initial condition. In this case, the number of vertical layer must be 1. The layer depth can be arbitrary as long as it is positive value.

4.4.7.2 Subsurface ocean scheme

At a surface grid point whose ocean fraction exists (= land fraction is less than 1.0), the physical quantities must be calculated by the ocean submodel. The setting of land-ocean fraction is controlled by `[PARAM_LANDUSE]`. When `(OCEAN_DYN_TYPE)` is set to `NONE` or `OFF` although the grid points with the ocean fraction exist, the following error message is displayed to the log file:

```
ERROR [CPL_vars_setup] Ocean fraction exists, but ocean component has not been called.
 Please check this inconsistency. STOP.
```

If `(OCEAN_DYN_TYPE)` is set to `SLAB`, the subsurface ocean is treated as a slab layer. The temperature of the slab layer changes over time by the heat flux from the surface. The depth of the slab layer, which controls the heat capacity of the slab ocean, can be specified as the depth of the first ocean layer. Note that the number of ocean layer must be 1.

In the slab ocean model, you can apply a relaxation (i.e., nudging) for sea surface temperature (SST) by using external data. The parameter of the nudging can be specified in `[PARAM_OCEAN_DYN_SLAB]`.

Table 4.4.11: List of sea ice scheme types (`OCEAN_ICE_TYPE`).

Schemes	Description of scheme
NONE	Disable the sea ice model
INIT	Fixed to the initial condition
SIMPLE	Simple sea ice model

Table 4.4.12: List of ocean surface albedo scheme types (`OCEAN_ALB_TYPE`).

Schemes	Description of scheme	Reference
INIT	Fixed to the initial condition	
CONST	Use constant value	
NAKAJIMA00	Calculate albedo for short-wave by solar zenith angle	Nakajima et al. (2000)

```
&PARAM_OCEAN_DYN_SLAB
 OCEAN_DYN_SLAB_nudging = .false.,                        ; Use nudging for ocean vari-
                                                            ables?
 OCEAN_DYN_SLAB_nudging_tau = 0.0_DP,                     ; Relaxation time for nudging
 OCEAN_DYN_SLAB_nudging_tau_unit = "SEC",                 ; Relaxation time unit
 OCEAN_DYN_SLAB_nudging_basename = "",                    ; Base name of input data
 OCEAN_DYN_SLAB_nudging_enable_periodic_year = .false.,   ; Annually cyclic data?
 OCEAN_DYN_SLAB_nudging_enable_periodic_month = .false.,  ; Monthly cyclic data?
 OCEAN_DYN_SLAB_nudging_enable_periodic_day = .false.,    ; Daily cyclic data?
 OCEAN_DYN_SLAB_nudging_step_fixed = 0,                   ; Option for using specific step
                                                            number of the data
 OCEAN_DYN_SLAB_nudging_defval = UNDEF,                   ; Default value of the variables
 OCEAN_DYN_SLAB_nudging_check_coordinates = .true.,       ; Check coordinate of variables
 OCEAN_DYN_SLAB_nudging_step_limit = 0,                   ; Maximum limit of the time
                                                            steps of the data

 /
```

When `(OCEAN_DYN_SLAB_nudging_tau)` is 0, the value of sea surface temperature is totally replaced by the external file. When `(OCEAN_DYN_SLAB_nudging_step_fixed)` is less than 1, the value of current time is calculated by temporal interpolation from the input data. When the specific step number is set for `(OCEAN_DYN_SLAB_nudging_step_fixed)`, the data of that step is always used without temporal interpolation. When the number larger than 0 is set to `(OCEAN_DYN_SLAB_nudging_step_limit)`, the data at the time step exceed this limit would not be read. The last read data is used for nudging. When `(OCEAN_DYN_SLAB_nudging_step_limit)` is 0, no limit is set.

If `(OCEAN_DYN_TYPE)` is set to `"OFFLINE"`, no dynamical nor physical process of subsurface ocean are calculated. The temporal changes of sea surface temperature are provided by the external file. This is the same as `(OCEAN_TYPE)` = `"FILE"` in the old version of SCALE , and also same as the case of `(OCEAN_DYN_SLAB_nudging_tau)` = 0 in the slab ocean scheme.

Table 4.4.13: List of ocean surface roughness scheme types (`OCEAN_RGN_TYPE`).

Schemes	Description of scheme	Reference
MOON07	Based on empirical formula with time development	Moon et al. (2007)
INIT	Fixed to the initial condition	
CONST	Use constant value	
MILLER92	Based on empirical formula without time development	Miller et al. (1992)

```
&PARAM_OCEAN_DYN_OFFLINE
  OCEAN_DYN_OFFLINE_basename = "",                        ; Base name of input data
  OCEAN_DYN_OFFLINE_enable_periodic_year = .false.,       ; Annually cyclic data?
  OCEAN_DYN_OFFLINE_enable_periodic_month = .false.,      ; Monthly cyclic data?
  OCEAN_DYN_OFFLINE_enable_periodic_day = .false.,        ; Daily cyclic data?
  OCEAN_DYN_OFFLINE_step_fixed = 0,                       ; Option for using specific step number
                                                            of the data
  OCEAN_DYN_OFFLINE_defval = UNDEF,                       ; Default value of the variables
  OCEAN_DYN_OFFLINE_check_coordinates = .true.,           ; Check coordinate of variables
  OCEAN_DYN_OFFLINE_step_limit = 0,                       ; Maximum limit of the time steps of
                                                            the data

/
```

Each parameter for the external file in offline mode follows the parameter for the nudging settings in the slab ocean scheme.

4.4.7.3 Ocean surface process

The ocean surface process contains the following subprocess:

- Calculation of open ocean (ice-free) surface

  - Calculation of the ocean surface albedo
  - Calculation of the ocean surface roughness length
  - Calculation of the heat/evaporation/emission flux between the atmosphere and the ocean

- Calculation of sea ice surface

  - Calculation of the ice surface albedo
  - Calculation of the ice surface roughness length
  - Calculation of the thermal conductance between the sea ice and the subsurface ocean
  - Calculation of the heat/evaporation/emission flux between the atmosphere and the ice
  - Calculation of the heat and water flux between the ice and the subsurface ocean

The albedo of the open ocean surface is configured by the scheme selected by (OCEAN_ALB_TYPE). If (OCEAN_ALB_TYPE) is set to "CONST", the albedo values over the open ocean are constant and set by (PARAM_OCEAN_PHY_ALBEDO_const). If (OCEAN_ALB_TYPE) is set to "NAKAJIMA00", the albedo for the short-wave is calculated from the solar zenith angle, while the albedo parameters set by (PARAM_OCEAN_PHY_ALBEDO_const) are used for the long-wave (IR).

```
&PARAM_OCEAN_PHY_ALBEDO_const
  OCEAN_PHY_ALBEDO_IR_dir = 0.05D0,      ; Ocean surface albedo for the long-wave (IR), direct light
  OCEAN_PHY_ALBEDO_IR_dif = 0.05D0,      ; Ocean surface albedo for the long-wave (IR), diffuse
                                           light
  OCEAN_PHY_ALBEDO_NIR_dir = 0.07D0,     ; Ocean surface albedo for the short-wave (near-IR), di-
                                           rect light
  OCEAN_PHY_ALBEDO_NIR_dif = 0.06D0,     ; Ocean surface albedo for the short-wave (near-IR), dif-
                                           fuse light
  OCEAN_PHY_ALBEDO_VIS_dir = 0.07D0,     ; Ocean surface albedo for the short-wave (visible), direct
                                           light
  OCEAN_PHY_ALBEDO_VIS_dif = 0.06D0,     ; Ocean surface albedo for the short-wave (visible), diffuse
                                           light
/
```

The albedos over the sea ice surface are constant regardless of (OCEAN_ALB_TYPE). The values are set by (PARAM_OCEAN_PHY_ALBEDO_seaice).

```
&PARAM_OCEAN_PHY_ALBEDO_seaice
  OCEAN_PHY_ALBEDO_seaice_IR_dir = 0.05D0,     ; Sea ice surface albedo for the long-wave (IR),
                                                 direct light
  OCEAN_PHY_ALBEDO_seaice_IR_dif = 0.05D0,     ; Sea ice surface albedo for the long-wave (IR),
                                                 diffuse light
  OCEAN_PHY_ALBEDO_seaice_NIR_dir = 0.60D0,    ; Sea ice surface albedo for the short-wave (near-
                                                 IR), direct light
  OCEAN_PHY_ALBEDO_seaice_NIR_dif = 0.60D0,    ; Sea ice surface albedo for the short-wave (near-
                                                 IR), diffuse light
  OCEAN_PHY_ALBEDO_seaice_VIS_dir = 0.80D0,    ; Sea ice surface albedo for the short-wave (visi-
                                                 ble), direct light
  OCEAN_PHY_ALBEDO_seaice_VIS_dif = 0.80D0,    ; Sea ice surface albedo for the short-wave (visi-
                                                 ble), diffuse light
/
```

The roughness length of ocean surface is calculated by the scheme selected by (OCEAN_RGN_TYPE). If (OCEAN_RGN_TYPE) is set to "CONST", the parameters in (PARAM_OCEAN_PHY_ROUGHNESS_const) are used.

```
&PARAM_OCEAN_PHY_ROUGHNESS_const
  OCEAN_PHY_ROUGHNESS_Z0M = 1.0D-5,    ; Ocean surface roughness length for momentum [m]
  OCEAN_PHY_ROUGHNESS_Z0H = 1.0D-5,    ; Ocean surface roughness length for heat [m]
  OCEAN_PHY_ROUGHNESS_Z0E = 1.0D-5,    ; Ocean surface roughness length for moisture [m]
/
```

When (OCEAN_RGN_TYPE) is set to "MOON07" or "MILLER92", the roughness length for momentum, heat, and water vapor are calculated in the selected scheme. You can specify some limiters by setting (PARAM_OCEAN_PHY_ROUGHNESS).

```
&PARAM_OCEAN_PHY_ROUGHNESS
  OCEAN_PHY_ROUGHNESS_visck = 1.5D-5,        ; Kinematic viscosity [m2/s]
  OCEAN_PHY_ROUGHNESS_Ustar_min = 1.0D-3,    ; Minimum limit of friction velocity [m/s]
  OCEAN_PHY_ROUGHNESS_ZOM_min = 1.0D-5,      ; Minimum limit of surface roughness length for
                                               momentum [m]
  OCEAN_PHY_ROUGHNESS_ZOH_min = 1.0D-5,      ; Minimum limit of roughness length for heat [m]
  OCEAN_PHY_ROUGHNESS_ZOE_min = 1.0D-5,      ; Minimum limit of roughness length for moisture
                                               [m]
/
```

The roughness lengths of the sea ice surface are constant regardress of (OCEAN_RGN_TYPE).
The values are set by (PARAM_OCEAN_PHY_ROUGHNESS_seaice). The minimum limit of roughness
lengths specified by (PARAM_OCEAN_PHY_ROUGHNESS) are also applied to the value over the sea ice
surface.

```
&PARAM_OCEAN_PHY_ROUGHNESS_seaice
  OCEAN_PHY_ROUGHNESS_seaice_ZOM = 2.0D-2,   ; Sea ice surface roughness length for momentum
                                               [m]
  OCEAN_PHY_ROUGHNESS_seaice_ZOH = 2.0D-3,   ; Sea ice surface roughness length for heat [m]
  OCEAN_PHY_ROUGHNESS_seaice_ZOE = 2.0D-3,   ; Sea ice surface roughness length for moisture [m]
/
```

By using the albedo and the roughness length, the surface fluxes between atmosphere and ocean,
and fluxes between atmosphere and ice are calculated by the scheme selected by (OCEAN_SFC_TYPE).
The bulk scheme specified by (BULKFLUX_TYPE) in [PARAM_BULKFLUX] is used for this calculation.
Refer to Section 4.4.6 for more detail of bulk scheme.

Ocean ice process

When (OCEAN_ICE_TYPE) is set to "SIMPLE", the ocean ice process comes to be considered. The
thermal conductance between the sea ice and the subsurface ocean is calculated by using the
parameter specified in [PARAM_OCEAN_PHY_TC_seaice].

```
&PARAM_OCEAN_PHY_TC_seaice
  OCEAN_PHY_thermalcond_max = 10.D0,     ; Maximum  thermal  conductivity  per  depth
                                           [J/m2/s/K]
  OCEAN_PHY_thermalcond_seaice = 2.D0,   ; Thermal conductivity of sea ice [J/m/s/K]
/
```

The parameters of sea ice process is configured by [PARAM_OCEAN_PHY_ICE].

```
&PARAM_OCEAN_PHY_ICE
  OCEAN_PHY_ICE_density = 1000.D0,          ; Density of sea ice [kg/m3]
  OCEAN_PHY_ICE_mass_critical = 1600.D0,    ; Ice amount for fraction = 1 [kg/m2]
  OCEAN_PHY_ICE_mass_limit = 50000.D0,      ; Maximum limit of ice amount [kg/m2]
  OCEAN_PHY_ICE_fraction_limit = 1.D0,      ; Maximum limit of ice fraction [1]
/
```

In SCALE , the mass amount of the sea ice is the prognostic variable, and the ice fraction is
diagnosed by following equation:

$$ICE\_fraction = \sqrt{\frac{ICE\_mass}{OCEAN\_PHY\_ICE\_mass\_critical}}.$$

4.4.8   Land Model

Similar to the ocean model, the land model consists of two main parts, i.e., an update of the state of the land surface, and calculation of fluxes at the interface of atmosphere and land. The timing of the calling of these scheme is configured in [PARAM_TIME] (Refer to Section 4.2.5 for the details).

Land Scheme

The land scheme that updates the state of land, e.g., land surface temperature, soil temperature, and soil moisture, is configured by (LAND_DYN_TYPE) in [PARAM_LAND] in the files `init.conf` and `run.conf`:

```
&PARAM_LAND
 LAND_DYN_TYPE = "NONE",   ; Select the land type shown in Table 4.4.14
 LAND_SFC_TYPE = "SKIN",   ; (For BUCKET) Select the land surface type shown in Table 4.4.15
/
```

Table 4.4.14: List of land scheme types

Schemes	Description of scheme
NONE or OFF	Do not use land model
BUCKET	Heat diffusion/bucket model
INIT	Fixed to the initial condition

If the land type is included in the land use setup by [PARAM_LANDUSE], neither NONE nor OFF can be given to (LAND_DYN_TYPE). If this condition is not satisfied, the program immediately terminates without computation, outputting the following message to the LOG file:

```
ERROR [CPL_vars_setup] Land fraction exists, but land component has not been called.
 Please check this inconsistency. STOP.
```

If (LAND_DYN_TYPE) except "NONE" or "OFF" is specified, the input data of land use distribution and a parameter table including information such as roughness length and albedo for each land-use category are required. An example of the parameter table is provided in the file `scale-rm/test/data/land/param.bucket.conf`.

Setting vertical grids for land model

The number of vertical layers for land model is specified by (LKMAX) in [PARAM_LAND_GRID_CARTESC_INDEX]. The vertical grid intervals are specified by (LDZ) in [PARAM_LAND_GRID_CARTESC]. The unit is [m].

```
&PARAM_LAND_GRID_CARTESC_INDEX
 LKMAX = 7,                                      ; number of vertical layers
/

&PARAM_LAND_GRID_CARTESC
 LDZ = 0.05, 0.15, 0.30, 0.50, 1.00, 2.00, 4.00,   ; grid interval along the vertical direction
/
```

You can specify an array in (LDZ) for the number of vertical layers specified by (LKMAX). The order of the array is from the ground surface to the underground.

Fluxes between atmosphere and land surface

The surface fluxes between atmosphere and land surface is calculated in the scheme selected by (LAND_SFC_TYPE). The bulk scheme specified in (BULKFLUX_TYPE) in [PARAM_BULKFLUX] is used for this calculation. Refer to Section 4.4.6 for more detail of bulk scheme.

4.4.8.1 Fixed to initial condition

If (LAND_DYN_TYPE) is INIT, the land condtion is fixed to the initial condition. In this case, the number of vertical layer ((LKMAX)) must be 1. The layer depth can be arbitrary as long as it is positive value.

4.4.8.2 BUCKET land scheme

land surface skin temperature

The definition of the land surface skin temperature is specified by (LAND_SFC_TYPE) in [PARAM_LAND]. The default setting is SKIN.

Table 4.4.15: List of land surface scheme types for BUCKET land scheme

Schemes	Description of scheme
SKIN	Determine surface temperature at which energy fluxes balance
FIXED-TEMP	Use the soil temperature at the uppermost land level as surface temperature

When (LAND_SFC_TYPE) is SKIN, the surface temperature is determined so that the heat budget of the energy fluxes at the ground surface is balanced. In practice, the surface temperature is calculated iteratively to reduce residual from the heat balance based on Tomita (2009). This surface temperature is different from the soil temperature at the uppermost layer of a land model.

When (LAND_SFC_TYPE) is FIXED-TEMP, the surface temperature is assumed to be the same as the soil temperature at the uppermost layer of a land model; the soil temperature is determined by the land model physics. Heat energy fluxes are calculated by the given surface temperature. The residuals of heat balance are provided to the land model as ground heat flux.

land nudging

If (LAND_DYN_TYPE) is BUCKET, you can apply relaxation (nudging) of land variables by using external data. The parameter of the nudging can be specified in run.conf.

```
&PARAM_LAND_DYN_BUCKET
  LAND_DYN_BUCKET_nudging = .false.,                            ; use nudging for land vari-
                                                                 ables?
  LAND_DYN_BUCKET_nudging_tau = 0.0_DP,                         ; relaxation time for nudging
  LAND_DYN_BUCKET_nudging_tau_unit = "SEC",                     ; relaxation time unit
  LAND_DYN_BUCKET_nudging_basename = "",                        ; base name of input data
  LAND_DYN_BUCKET_nudging_enable_periodic_year = .false.,       ; annually cyclic data?
  LAND_DYN_BUCKET_nudging_enable_periodic_month = .false.,      ; monthly cyclic data?
  LAND_DYN_BUCKET_nudging_enable_periodic_day = .false.,        ; dayly cyclic data?
  LAND_DYN_BUCKET_nudging_step_fixed = 0,                       ; Option for using specific step
                                                                 number of the data
  LAND_DYN_BUCKET_nudging_defval = UNDEF,                       ; default value of the variables
  LAND_DYN_BUCKET_nudging_check_coordinates = .true.,          ; check coordinate of variables
  LAND_DYN_BUCKET_nudging_step_limit = 0,                       ; maximum limit of the time
                                                                 steps of the data
/
```

When (LAND_DYN_BUCKET_nudging_tau) is 0, the value of land variables is totally replaced by the external file. This configuration is the same as (LAND_TYPE) = "FILE" in the old version. When (LAND_DYN_BUCKET_nudging_step_fixed) is less than 1, the certain time of the input data is used and temporal interpolation is applied to calculate the value of current time. When the specific step number is set for (LAND_DYN_BUCKET_nudging_step_fixed), the data of that step is always used without interpolation. When the number larger than 0 is set to (LAND_DYN_BUCKET_nudging_step_limit), The data at the time step exceed this limit would not be read. When (LAND_DYN_BUCKET_nudging_step_limit) is 0, no limit is set.

4.4.9 Urban model

The urban scheme calculates the energy and water fluxes between the atmosphere and urban surface/canopy. To calculate them, it also updates the urban state, e.g., urban surface temperature, moisture, and so on. The timing of updating (calling) the urban scheme is configured at (TIME_DT_URBAN) and (TIME_DT_URBAN_UNIT) in [PARAM_TIME] (Refer to Section 4.2.5 for the details).

The urban scheme used is specified in (URBAN_DYN_TYPE) in [PARAM_URBAN] in init.conf and run.conf, as follows:

```
&PARAM_URBAN
  URBAN_DYN_TYPE = "NONE",    ; Select a scheme from Table 4.4.16
/
```

Table 4.4.16: List of urban scheme types

Schemes	Description of scheme	reference
NONE or OFF	Do not use an urban scheme	
LAND	Urban areas are calculated by a land model	
KUSAKA01	Single-layer canopy model	Kusaka et al. (2001)

If urban land-use is included in the calculation domain, neither NONE nor OFF can be chosen as (URBAN_TYPE). When they are selected under that situation, the program immediately stops without calculation, outputting the following message to LOG file:

```
ERROR [CPL_vars_setup] Urban fraction exists, but urban component has not been called.
 Please check this inconsistency. STOP.
```

If `LAND` type is selected as `(URBAN_DYN_TYPE)`, the land model specified in `[PARAM_LAND]` is used to calculate the surface fluxes and soil variables for urban areas. Since the land model implemented is only the slab model in the current version, Choosing `LAND` type means applying the slab model to the urban areas. In this case, parameters for an urban area should be provided in the land parameter table described in Section 4.4.8.

`KUSAKA01` scheme is a single-layer urban canopy model. Since the scheme considers the energy exchange between the top of urban canopy and atmosphere, the building height, i.e., `(ZR)` in `[PARAM_URBAN_DATA]`, must be located at least the two meters lower than the face level of the 1st atmospheric layer.

4.4.9.1 `KUSAKA01` urban scheme

Settings of layers at facets of urban structures: roof, wall, and road

`KUSAKA01` considers two-dimentional symmetrical street canyons as urban geometry. The boundaries between facets of urban structures and atmosphere consist of roof and wall surfaces of a building and road surface. The temperature of these surfaces are calculated by heat diffusion. The number of layers used to calculate the heat diffusion is specified by `(UKMAX)` in `[PARAM_URBAN_GRID_CARTESC_INDEX]`. The thickness of each layer is specified by `(UDZ)` in `[PARAM_URBAN_GRID_CARTESC]`. The unit is [m].

```
&PARAM_URBAN_GRID_CARTESC_INDEX
  UKMAX = 5,                              ; Number of layers at facets of urban structures
/

&PARAM_URBAN_GRID_CARTESC
  UDZ = 0.01, 0.01, 0.03, 0.05, 0.10,   ; Thickness of each facet layers
/
```

You can specify an array in `(UDZ)` for the number of layers specified by `(UKMAX)`. The order of the array is from the facet to inside of urban structures. The settings are applied to the surfaces of roof, wall, and road.

Urban parameters

There are many parameters characterizing urban morphology in the `KUSAKA01` scheme. Regarding the available urban parameters, please refer to the namelist of `[PARAM_URBAN_DATA]`; you can find it from the link in the "NAMELIST Parameters" of Section 5.3.2. The sample file for the `[PARAM_URBAN_DATA]` is provided at `scale/scale-rm/test/data/urban/param.kusaka01.dat`.

The configuration for urban parameters is specified in `[PARAM_URBAN_DYN_KUSAKA01]`.

```
&PARAM_URBAN_DYN_KUSAKA01
  DTS_MAX = 0.1,                                      ; Limit of temperature change per 1 step
                                                        = DTS_MAX * DT [K/step]
  BOUND = 1,                                          ; Boundary condition at the innermost
                                                     level of building, roof, and load
                                                       1: Zero-flux, 2: Temperature const.
  URBAN_DYN_KUSAKA01_PARAM_IN_FILENAME = "",         ; File name of urban parameter table
  URBAN_DYN_KUSAKA01_GRIDDED_ZOM_IN_FILENAME = "",   ; File name of gridded Z0m data
  URBAN_DYN_KUSAKA01_GRIDDED_AH_IN_FILENAME = "",    ; File name of gridded AH data
  URBAN_DYN_KUSAKA01_GRIDDED_AHL_IN_FILENAME = "",   ; File name of gridded AHL data
/
```

The values of urban parameters used in a calculation are determined through the following three steps.

1) The default values given in the SCALE source code are applied to all grid cells in the calculation domain.

2) If any file is given to (URBAN_DYN_KUSAKA01_PARAM_IN_FILENAME) of [PARAM_URBAN_DYN_KUSAKA01], the value of urban parameter(s) specified in [PARAM_URBAN_DATA] in the file is read. Then, the default value(s) is replaced with the given value(s). The replaced data is also applied to all grid cells in the calculation domain.

3) For the roughness length for momentum (Z0m) and anthropogenic sensible heat (AH) and latent heat (AHL), if the gridded data file is given, the gridded data is read from the file. Then, the value specified by 1) or 2) is replaced with the gridded data. That is, it is possible to use different values for each grid for Z0m, AH, and AHL. The gridded data file for Z0m, AH, and AHL is specified by (URBAN_DYN_KUSAKA01_GRIDDED_ZOM_IN_FILENAME), (URBAN_DYN_KUSAKA01_GRIDDED_AH_IN_FILENAME), and (URBAN_DYN_KUSAKA01_GRIDDED_AHL_IN_FILENAME), respectively.

Users need to prepare beforehand the gridded data for (URBAN_DYN_KUSAKA01_GRIDDED_(ZOM|AH|AHL)_IN_FILENAME) with the SCALE-netCDF format according to the experimental settings. These SCALE-netCDF data can be converted from the simple binary format data using `scale_init`.

How to prepare gridded urban data with SCALE-netCDF
`scale_init` is used to prepare the gridded urban data; the detailed explanations are given in Section 4.1.3. Here, a practical usage is demonstrated with examples of configuration file and namelist file.

To create the SCALE-netCDF file according to the experimental settings from a certain binary format data, add [PARAM_CONVERT] and [PARAM_CNVUSER] to `init.conf`. The following shows an example of `init.conf` in the case of preparing AH gridded data.

```
&PARAM_CONVERT
 CONVERT_USER = .true.,
/

&PARAM_CNVUSER
 CNVUSER_FILE_TYPE = "GrADS",
 CNVUSER_NSTEPS = 24,                           ; Set 1 for Z0M and 24 for AH and AHL
 CNVUSER_GrADS_FILENAME = "namelist.grads.ah",
 CNVUSER_GrADS_VARNAME = "AH",                  ; (name) of (GrADS_ITEM) in namelist file
 CNVUSER_GrADS_LONNAME = "lon",                 ; (name) of (GrADS_ITEM) in namelist file
 CNVUSER_GrADS_LATNAME = "lat",                 ; (name) of (GrADS_ITEM) in namelist file
 CNVUSER_OUT_BASENAME = "urb_ah.d01",
 CNVUSER_OUT_VARNAME = "URBAN_AH",              ; URBAN_AH, URBAN_AHL, or URBAN_ZOM
 CNVUSER_OUT_VARDESC = "Anthropogenic sensible heat flux",
 CNVUSER_OUT_VARUNIT = "W/m2",                  ; Unit
 CNVUSER_OUT_DTYPE = "REAL8"
 CNVUSER_OUT_DT = 3600D0,
/
```

The following shows an example of namelist file, corresponding to "ctl" file of GrADS .

```
#
# Dimension
#
&GrADS_DIMS
 nx = 361,
 ny = 181,
 nz = 1,
/


#
# Variables
#
&GrADS_ITEM name='lon', dtype='linear', swpoint=0.0d0, dd=1.0d0 /
&GrADS_ITEM name='lat', dtype='linear', swpoint=-90.0d0, dd=1.0d0 /
&GrADS_ITEM name='AH', dtype='map', fname='urb_ah', startrec=1, totalrec=1, \
        bintype='real4', yrev=.false., missval=-999.0E+0 /
```

Although the initial time and the time interval in the output file are specified by (TIME_STARTDATE) in [PARAM_TIME] and (CNVUSER_OUT_DT), respectively, as described in Sec. 4.1.3, the urban scheme ignores these settings when reading the data. That is, regardless of the time coordinate in the SCALE-netCDF file, the gridded AH and AHL data are assumed to contain the data for 24 hours in the order from 1 to 24 (local time: LT) with the interval of one hour. The local time here is defined by $UTC + $ (MAPPROJECTION_basepoint_lon)$/15.0$ using the longitude of the reference point. Note that the unit of the time axis of the AH and AHL data array may change in the next version or later. The Z0M data is read only the first data because Z0M is assumed to be unchanged during the simulation (namely, (CNVUSER_NSTEPS)=1). The other settings are as described in Sec. 4.1.3.

Chapter 4.5

Common settings

4.5.1 Log File

4.5.1.1 Log File Output

SCALE-RM can output log files when you execute `scale-rm`, `scale-rm_init`, and `scale-rm_pp`. On a default setting, log messages from the process of number zero is written into "`LOG.pe000000`" for `scale-rm`, "`init_LOG.pe000000`" for `scale-rm_init`", and `pp_LOG.pe000000`" for `scale-rm_pp`. Users can modify the settings of log file outputs by editing configuration files as follows:

```
&PARAM_IO
 IO_LOG_BASENAME = 'LOG',          ; Output name of log file
 IO_LOG_ALLNODE = .false.,         ; Output log files for all processes?
 IO_LOG_SUPPRESS = .false.,        ; If .true., log output is suppressed
 IO_LOG_NML_SUPPRESS = .false.,    ; If .true., namelist parameter output is suppressed
 IO_NML_FILENAME = '',             ; If specified, namelist parameter is output to specified file.
                                     Otherwise to the log file.
 IO_STEP_TO_STDOUT = -1,           ; If positive, the time step information is output to standard
                                     output.
/
```

The name of the log file is controlled by (`IO_LOG_BASENAME`) in [`PARAM_IO`]. In the case of above default setting, the name of log file for the master process is "`LOG.pe000000`". (`IO_LOG_ALLNODE`) controls whether to output log files for all processes or not. When (`IO_LOG_ALLNODE`) is `.true.`, log files for all processes will be created, otherwise the log file is output only from master process, namely rank 0.

If (`IO_LOG_SUPPRESS`) is set to `.true.`, no log files are created and the almost all log messages are not output. Only the information about elapsed time is sent to the standard output (STDOUT), even if (`IO_LOG_SUPPRESS`) is set to `.true.`.

The NAMELIST parameters are output unless (`IO_LOG_NML_SUPPRESS`) is set to `.true.`. The parameters are output to the log file as default. They can be output different file by setting (`IO_NML_FILENAME`). Note that the file specified by (`IO_NML_FILENAME`) can be used as input configuration file for later runs.

The time step information is output to the log file. Details of the information is explained in the next section. If (`IO_STEP_TO_STDOUT`) is set as > 0, the time step information is output to the STDOUT, too. All the time step information is output to the log file. For the STDOUT, the step interval of output is specified by the number of (`IO_STEP_TO_STDOUT`).

4.5.1.2 Execution Time Information in Log File

You can find the lines with following format in the log file when you execute `scale-rm`:

```
+++++ TIME: 0000/01/01 00:06:36 + 0.600 STEP: 1984/ 432000 WCLOCK: 2000.2
```

This line presents messages about status of computation as follows:

- Now, carrying out 6m36.6s of time integration from the initial time of "0000/01/01 00:00:00 + 0.000".

- This is 1984th step in whole time steps of 432000.

- 2000.2s was taken in a wall clock time (cpu time) .

Furthermore, the required time for this computation can be estimated from these information. In this case, 121 hours ($= 2000.2 \times 432000 \div 1984$) is the estimated elapsed time.

Messages in the log file are output in following format.

```
type [subroutine name]  message
     messages
      ...
```

type : message type that can take one of the following.

- INFO: General information about job execution
- WARN: Considerable event about job execution
- ERROR: Fatal error that involves stop of execution

subroutine name : the subroutine name writing the message.

message : the main body of the message.

The sample of error message is as follows.

```
  ERROR [ATMOS_PHY_MP_negative_fixer] large negative is found. rank = 1
  k,i,j,value(QHYD,QV) = 17 8 1 1.7347234759768071E-018 0.0000000000000000
  k,i,j,value(QHYD,QV) = 19 8 1 -5.4717591620764856E-003 0.0000000000000000
  ...
```

4.5.2 Setting the monitor file

A monitor file and output variables are configured at [PARAM_MONITOR] and [MONITOR_ITEM] in `run.conf`. The default format of the monitor file is configured at [PARAM_MONITOR].

```
&PARAM_MONITOR
 MONITOR_OUT_BASENAME = "monitor",   ; Base name of the output file
 MONITOR_USEDEVIATION = .true.,      ; Use deviation from first step?
 MONITOR_GLOBAL_SUM = .true.,        ; Use global summation?
 MONITOR_STEP_INTERVAL = 1,          ; Step interval of monitor output
/
```

```
&MONITOR_ITEM
 NAME = "ENGT",    Variable name. List of variables is shown in Table 4.5.1, 4.5.2, 4.5.3, and 4.5.4
/
```

Monitor component outputs domain total of physical quantity such as dry air mass, water vapor, total energy, surface precipitation flux, and so on. These output is useful to check mass and energy budget. The output value is global domain total if (MONITOR_GLOBAL_SUM) is .true., while it is total in the individual MPI process. The output value is the difference from the initial value for the snapshot-type quantity, if (MONITOR_USEDEVIATION) is .true. For the tendency-type quantity, the output value is temporally accumulated value.

The monitor file is ASCII format, and the name of the file is set by (MONITOR_OUT_BASENAME). It is (MONITOR_OUT_BASENAME).peall if (MONITOR_GLOBAL_SUM) is .true., nothewise it is (MONITOR_OUT_BASENAME).peXXXXXX, where XXXXXX is the process number.

The time interval of monitor output is specified in the (MONITOR_STEP_INTERVAL) as the multiple of time step (Δt).

For example, let the below setting for [MONITOR_ITEM] be added with (MONITOR_STEP_INTERVAL) = 10 and (MONITOR_USEDEVATION) = .true.,

```
&MONITOR_ITEM NAME="DENS" /
&MONITOR_ITEM NAME="QTOT" /
&MONITOR_ITEM NAME="EVAP" /
&MONITOR_ITEM NAME="PREC" /
```

The monitor file is output as follows;

```
DENS QTOT EVAP PREC
STEP= 1 (MAIN) 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
STEP= 11 (MAIN) -2.27510244E+11 6.67446186E+11 9.39963392E+10 2.98914905E+11
STEP= 21 (MAIN) -3.04179976E+11 1.16811060E+12 1.64602175E+11 7.56753096E+11
STEP= 31 (MAIN) -7.55688670E+11 1.42784177E+12 2.25452889E+11 1.42932656E+12
STEP= 41 (MAIN) -9.45082752E+11 1.56057082E+12 2.82959478E+11 2.19673659E+12
STEP= 51 (MAIN) -1.02869018E+12 1.66179511E+12 3.45854371E+11 2.98295445E+12
STEP= 61 (MAIN) -1.69997222E+12 1.74413176E+12 4.20139948E+11 3.78414734E+12
STEP= 71 (MAIN) -1.72816474E+12 1.81512719E+12 5.04055360E+11 4.59740827E+12
STEP= 81 (MAIN) -1.58692434E+12 1.88174470E+12 5.93665632E+11 5.41341475E+12
STEP= 91 (MAIN) -1.71362764E+12 1.94867974E+12 6.86327009E+11 6.22069061E+12
STEP= 101 (MAIN) -2.04231630E+12 1.99886166E+12 7.80859828E+11 7.03479603E+12
```

Table 4.5.1: Variables in the atmospheric model available for monitor output

Values	Description	Unit	Type
DENS	Air mass	kg	snapshot
MOMZ	Momentum in z-direction	kg m/s	snapshot
MOMX	Momentum in x-direction	kg m/s	snapshot
MOMY	Momentum in y-direction	kg m/s	snapshot
RHOT	Potential temperature	kg K	snapshot
TRACER*	Tracers in prognostic variable	unit $\times$ kg	snapshot
QDRY	Dry air mass	kg	snapshot
QTOT	Water mass	kg	snapshot
EVAP	Evaporation at the surface	kg	tendency
PREC	Precipitation	kg	tendency
ENGT	Total energy (ENGP + ENGK + ENGI)	J	snapshot
ENGP	Potential energy ($\rho * g * z$)	J	snapshot
ENGK	Kinetic energy ($\rho * (W^2 + U^2 + V^2)/2$)	J	snapshot
ENGI	Internal energy ($\rho * C_v * T$)	J	snapshot
ENGFLXT	Total energy flux convergence (SH + LH + SFC_RD - TOM_RD)	J	tendency
ENGSFC_SH	Surface sensible heat flux	J	tendency
ENGSFC_LH	Surface latent heat flux	J	tendency
ENGSFC_EVAP	Internal energy of surface evapolation	J	tendency
ENGSFC_PREC	Internal energy of precipitation	J	tendency
ENGSFC_RD	Surface net radiation flux (SFC_LW_up+SFC_SW_up-SFC_LW_dn-SFC_SW_dn)	J	tendency
ENGTOM_RD	Top-of-model net radiation flux (TOM_LW_up+TOM_SW_up-TOM_LW_dn-TOM_SW_dn)	J	tendency
ENGSFC_LW_up	Surface longwave upward flux	J	tendency
ENGSFC_LW_dn	Surface longwave downward flux	J	tendency
ENGSFC_SW_up	Surface shortwave upward flux	J	tendency
ENGSFC_SW_dn	Surface shortwave downward flux	J	tendency
ENGTOM_LW_up	Top-of-model longwave upward flux	J	tendency
ENGTOM_LW_dn	Top-of-model longwave downward flux	J	tendency
ENGTOM_SW_up	Top-of-model shortwave upward flux	J	tendency
ENGTOM_SW_dn	Top-of-model shortwave downward flux	J	tendency
MASSTND_DAMP	Mass tendency by nudging	kg	tendency
MASSFLX_WEST	Mass flux at western boundary	kg	tendency
MASSFLX_EAST	Mass flux at eastern boundary	kg	tendency
MASSFLX_SOUTH	Mass flux at southern boundary	kg	tendency
MASSFLX_NORTH	Mass flux at northern boundary	kg	tendency
QTOTTND_DAMP	Water mass tendency by nudging	kg	tendency
QTOTFLX_WEST	Water mass flux at western boundary	kg	tendency
QTOTFLX_EAST	Water mass flux at eastern boundary	kg	tendency
QTOTFLX_SOUTH	Water mass flux at southern boundary	kg	tendency
QTOTFLX_NORTH	Water mass flux at northern boundary	kg	tendency
QTOTTND_NF	Water mass tendency by negative fixer	kg	tendency
ENGITND_NF	internal energy tendency by negative fixer	kg	tendency
QTOTFLX_TB_WEST	Water mass flux at western boundary by turbulence	kg	tendency
QTOTFLX_TB_EAST	Water mass flux at eastern boundary by turbulence	kg	tendency
QTOTFLX_TB_SOUTH	Water mass flux at southern boundary by turbulence	kg	tendency
QTOTFLX_TB_NORTH	Water mass flux at northern boundary by turbulence	kg	tendency

Table 4.5.2: Variables in the ocean model available for monitor output

Values	Description	Unit	Type
OCN_TEMP	Sea water temperature	K m^3	snapshot
OCN_ICE_TEMP	Sea ice temperature	K m^3	snapshot
OCN_ICE_MASS	Sea ice mas	kg	snapshot
OCN_MASFLX_TOP	Surface mass flux (top of open ocean and sea ice)	kg	snapshot
OCN_MASFLX_MID	Sea surface mass flux (top of open ocean and bottom of sea ice)	kg	snapshot
OCN_MAS_SUPL	Mass supply to conserve sea water	kg	tendency
OCN_MASCNV	Total mass convergence	kg	tendency
OCN_WTR_MASCNV	Sea water mass convergence	kg	tendency
OCN_ICE_MASCNV	Sea ice mass convergence	kg	tendency
OCN_WTR_ENGI	Sea water internal energy	J	snapshot
OCN_ICE_ENGI	Sea ice internal energy	J	snapshot
OCN_GHFLX_TOP	Surface ground heat flux	J	tendency
OCN_GHFLX_MID	Sea surface ground heat flux	J	tendency
OCN_ENGIFLX_TOP	Surface internal energy flux	J	tendency
OCN_ENGIFLX_MID	Sea surface internal energy flux	J	tendency
OCN_ENGI_SUPL	Internal energy supply	J	tendency
OCN_ENGICNV	Total internal energy convergence	J	tendency
OCN_WTR_ENGICNV	Sea water internal energy convergence	J	tendency
OCN_ICE_ENGICNV	Sea ice internal energy convergence	J	tendency

Table 4.5.3: Variables in the land model available for monitor output

Values	Description	Unit	Type
LND_TEMP	Land temperature	K m^3	snapshot
LND_WATER	Land water	kg	snapshot
LND_ICE	Land ice	kg	snapshot
LND_MASSFC	Surface water mss flux	kg	tendency
LND_ROFF	Runoff water	kg	tendency
LND_MASFLX	Total mass change	kg	tendency
LND_ENGI	Total internal energy	J	snapshot
LND_WTR_ENGI	Water internal energy	J	snapshot
LND_ICE_ENGI	Ice internal energy	J	snapshot
LND_ENGSFC_GH	Surface ground heat flux	J	tendency
LND_ENGSFC_EI	Surface internal energy flux	J	tendency
LND_ROFF_EI	Internal energy of runoff water	J	tendency
LND_ENGFLX	Total internal energy change	J	tendency

Table 4.5.4: Variables in the urban model available for monitor output

Values	Description	Unit	Type
URB_TRL	Roof temperature	K m^3	snapshot
URB_TBL	Wall temperature	K m^3	snapshot
URB_TGL	Road temperature	K m^3	snapshot
URB_TR	Roof surface temperature	K m^2	snapshot
URB_TB	Wall surface temperature	K m^2	snapshot
URB_TG	Road surface temperature	K m^2	snapshot
URB_TC	Canopy temperature	K m^2	snapshot
URB_UC	Canopy wind speede	m^3/s	snapshot
URB_QC	Canopy humidity	kg/m	snapshot
URB_RAINR	Roof water	kg	snapshot
URB_RAINB	Wall water	kg	snapshot
URB_RAING	Road water	kg	snapshot
URB_ROFF	Runoff water	kg	tendency

4.5.3 Setting Physical Constants

The some physical constants are able to modify in [PARAM_CONST] in the configuration file.

```
&PARAM_CONST
 CONST_RADIUS = 6.37122D+6,       ; Radius of the planet [m]
 CONST_OHM = 7.2920D-5,           ; Angular velocity of the planet [1/s]
 CONST_GRAV = 9.80665D0,          ; Standard acceleration of gravity [m/s2]
 CONST_Rdry = 287.04D0,           ; Specific gas constant (dry air) [J/kg/K]
 CONST_CPdry = 1004.64D0,         ; Specific heat (dry air,constant pressure) [J/kg/K]
 CONST_LAPS = 6.5D-3,             ; Lapse rate of ISA [K/m]
 CONST_Pstd = 101325.D0,          ; Standard pressure [Pa]
 CONST_PRE00 = 100000.D0,         ; Pressure reference [Pa]
 CONST_Tstd = 288.15D0,           ; Standard temperature (15C) [K]
 CONST_THERMODYN_TYPE = 'EXACT',  ; Internal energy type
 CONST_SmallPlanetFactor = 1.D0,  ; Factor for small planet [1]
/
```

When (CONST_THERMODYN_TYPE) is 'EXACT', temperature dependencies of the latent heat are considered. When (CONST_THERMODYN_TYPE) is 'SIMPLE', specific heat of water categories are set to that of dry air, and temperature dependencies of the latent heat are ignored. The (CONST_RADIUS) is multiplied by the (CONST_SmallPlanetFactor). At the same time, (CONST_OHM) is multiplied by the inverse of the (CONST_SmallPlanetFactor).

4.5.4 Setting Calendar

The calendar type can be specified in [PARAM_CALENDAR] in the configuration file. Gregorian calendar is used in default.

```
&PARAM_CALENDAR
 CALENDAR_360DAYS = .false.   ; Whether 12x30 days calendar is used?
 CALENDAR_365DAYS = .false.   ; Whether leap year is considered?
/
```

The setting of the calendar affects calculation of solar zenith angle. It is calculated so that length of one year and the full circle of the ecliptic match. Note that the external data in the different calendar should not be read.

When (CALENDAR_360DAYS) is .true., the calendar that one year is 12 months and one month is 30 days, is set. When (CALENDAR_365DAYS) is .true., the Gregorian calendar is used without leap year.

4.5.5 Setting Random Number Generator

A parameter of the random number generator is set in [PARAM_RANDOM] in the configuration file.

```
&PARAM_RANDOM
 RANDOM_FIX = .false.   ; Random seed is fixed?
/
```

The intrinsic function of random number generator is used in the scale library. Note that the generated number is pseudorandom. The seed of the random number is determined by current datetime, cpu time, and the process id. When (RANDOM_FIX) is .true., the seed is fixed by specific number. This option is useful to reproduce the simulation results, which use the random perturbation for the initial field.

4.5.6 Setting Performance Profiler

Parameters of the performance profiler are set in [PARAM_PROF] in the configuration file.

```
&PARAM_PROF
 PROF_rap_level = 2          ; Rap output level
 PROF_mpi_barrier = .false.  ; Add barrier command of MPI in every rap?
/
```

To measure the elapse time, utility functions (PROF_rapstart,PROF_rapend) are inserted in the source code. These measurement sections are also used for detailed performance profiling. The results of the rap time are displayed at the end of the log file. If (IO_LOG_ALLNODE) in [PARAM_IO] is .true., the result of each process is reported to each log file individually. If (IO_LOG_SUPPRESS) in [PARAM_IO] is .true., the result is sent to the standard output. Each measurement section has the output level. The section, which has the output level larger than the (PROF_rap_level) are not measured.

When (PROF_mpi_barrier) is .true., barrier command of MPI is called before and after getting current time. This option is useful to separate the computation time and communication time. The computation time often shows large imbalance among the processes.

4.5.7 Setting Statistics Monitor

Parameters of the statistics monitor are set in [PARAM_STATISTICS] in the configuration file.

```
&PARAM_STATISTICS
 STATISTICS_checktotal = .false.       ; Calculate and report variable totals to logfile?
 STATISTICS_use_globalcomm = .false.   ; Calculate total with global communication?
/
```

When (STATISTICS_checktotal) is .true., domain total of some variables are calculated and reported to the log file, for debugging. When (STATISTICS_use_globalcomm) is .true., global domain total is calculated by using global communication. This option can slow the simulation time. When the option is .false., the total is calculated in the spatial region allocated for each process.

Chapter 4.6

Post-processing

4.6.1   SCALE NetCDF Operator (SNO)

NOTICE

SNO is a post-processing tool for the SCALE-netCDF file generated by the SCALE version 5.3 or later. SNO is not available for the older version's SCALE-netCDF files because of the less information in the global attributes and the attributes of the axis data.

Unless (FILE_AGGREGATE)=.true. in [PARAM_FILE] with the Parallel netCDF (PnetCDF) (Sec. 5.1.1.5), the output file of SCALE-RM , i.e., SCALE-netCDF file, is decomposed into the multiple files according to the horizontal domain decomposition with the multiple MPI processes; each process outputs a file. Although this is an efficient way from the viewpoint of the file I/O throughput during the simulation, it may not be useful in the following points. 1) When the number of MPI processes is large, it is not easy to handle such a lot of files. 2) Even if we use the same experimental settings across multiple experiments except for the number of MPI processes, we cannot share the init/boundary/restart/history files between the experiments. 3) Many analytical and visualization tools do not support such the spatially decomposed files.

SNO has the following features to reduce the annoyance of handling these decomposed files. The format in the parentheses indicates the supported format which SNO output.

- Combine the decomposed multiple files into a single file (SCALE-netCDF).

- Convert a single file or multiple files into multiple files with different number of decomposition (SCALE-netCDF).

- Regrid data from the model-level to the different model-level coordinate (SCALE-netCDF).

- Regrid data from the model-level to the actual height-level coordinate (SCALE-netCDF).

- Regrid data from the model-level to the pressure-level coordinate (SCALE-netCDF).

- Regrid data from the model grid to the geodesic (latitude-longitude) grid coordinate (NetCDF, but NOT SCALE-netCDF !).

- Average history data over multiple steps (SCALE-netCDF).

4.6.1.1   Basic usage

The settings for SNO are specified in [PARAM_SNO].

```
&PARAM_SNO
  basename_in = "",           ; Path and basename of the input file(s)
  basename_out = "",          ; Basename of the output file
  vars = "",                  ; Name of variables to be processed
  debug = .false.,            ; Output verbose log for debugging?
  nprocs_x_out = 1,           ; Decomposition number of files in x-direction
                                 If output_grads = true, it must be set to 1.
  nprocs_y_out = 1,           ; Decomposition number of files in y-direction
                                 If output_grads = true, it must be set to 1.
  output_single = .false.,    ; (SCALE-netCDF ) Whether output to a single netCDF file when
                                using multiple MPI processes?
  dirpath_out = "",           ; (GrADS ) Directory path of a output file
  output_grads = .false.,     ; (GrADS ) Output with grads format?
/
```

(basename_in) is always required. The information of the decomposed files, such as the total number of files and the 2-D topology, is read from the first file (*.pe000000.nc).

If (vars) is not specified, all of variables in the input file are processed.

The default value of (nprocs_x_out) and (nprocs_y_out) is 1; it means that multiple files are combined to a single file. If (output_single) = .true., the input data is combined in a single file regardless of the settings of (nprocs_x_out) and (nprocs_y_out).

The number of MPI processes to execute SNO must be the same with nprocs_x_out $\times$ nprocs_y_out, and (nprocs_x_out) and (nprocs_y_out) must be a divisor of the decomposition number of input files in x- and y-directions, respectively.

When the output format is SCALE-netCDF , (basename_out) is required.

If (output_grads) is set to .true., data is output in a single file with binary format, which is readable by GrADS . If (dirpath_out) is empty, the directory path for output is set to the current directory. When the output format is GrADS , (nprocs_x_out) and (nprocs_y_out) must be 1.

In addition to [PARAM_SNO], SNO depends on the following namelist parameters:

- [PARAM_IO]: Log file (Sec. 4.5.1)

- [PARAM_PROF]: Performance Profiler (Sec. 4.5.6)

- [PARAM_CONST]: Physical Constants (Sec. 4.5.3)

- [PARAM_CALENDAR]: Calendar (Sec. 4.5.4)

4.6.1.2 Examples of configuration: conversion of data format and decomposition numbers

Convert multiple SCALE-netCDF files to a single NetCDF file (single MPI process)

```
&PARAM_SNO
  basename_in = 'input/history_d02',
  basename_out = 'output/history_d02_new',
/
```

This example converts history files named `history_d02.pe######.nc` in the directory `./input`, where `######` represents the MPI process number. Any options about the number of output file and variables are not specified in this example. Thus, all the variables in the input files are combined and output to a single file. The converted file is output to `./output` directory with the new name `history_d02_new.pe######.nc`.

Generally, a single netCDF file is readable by GrADS without any external metadata file. However, GrADS interface is limited and cannot understand the SCALE-netCDF format, which contains the associated coordinates and the map projections. Then, a control file is necessary for

GrADS to read SCALE-netCDF file. The following is an example of the output control file.

```
SET ^history_d02_new.pe000000.nc
TITLE SCALE-RM data output
DTYPE netcdf
UNDEF -0.99999E+31
XDEF 88 LINEAR 134.12 0.027
YDEF 80 LINEAR 33.76 0.027
ZDEF 35 LEVELS
 80.841 248.821 429.882 625.045 835.409 1062.158 1306.565 1570.008 1853.969
 2160.047 2489.963 2845.574 3228.882 3642.044 4087.384 4567.409 5084.820 5642.530
 6243.676 6891.642 7590.075 8342.904 9154.367 10029.028 10971.815 11988.030 13083.390
14264.060 15536.685 16908.430 18387.010 19980.750 21698.615 23550.275 25546.155
TDEF 25 LINEAR 00:00Z01MAY2010 1HR
PDEF 80 80 LCC 34.65 135.22 40 40 30.00 40.00 135.22 2500.00 2500.00
VARS 3
U=>U 35 t,z,y,x velocity u
PREC=>PREC 0 t,y,x surface precipitation flux
OCEAN_SFC_TEMP=>OCEAN_SFC_TEMP 0 t,y,x ocean surface skin temperature
ENDVARS
```

When the simple map projections is used such as mercator method, GrADS can read the single netCDF file using "sdfopen" command without preparing a control file.

Convert multiple SCALE-netCDF files to a single NetCDF file (multiple MPI processes)

```
&PARAM_SNO
 basename_in = 'input/history_d02',
 basename_out = 'output/history_d02',
 nprocs_x_out = 4,
 nprocs_y_out = 6,
 output_single = .true.,
/
```

This example outputs a single file containing all variables as well as the above example. On the other hand, in this example, the number of MPI processes must be set to 24 ($= 4 \times 6$).

Convert multiple SCALE-netCDF files to multiple SCALE-netCDF files with different number of docomposition

```
&PARAM_SNO
 basename_in = 'input/history_d02',
 basename_out = 'output/history_d02',
 nprocs_x_out = 4,
 nprocs_y_out = 6,
/
```

To output decomposed files, there is a restriction that each file must have the same number of grids (excluding the halo grid). To satisfy this restriction, (nprocs_x_out) must be a divisor of the total grid number (IMAXG) of the entire domain in X direction (see Sec. 4.2.3). (nprocs_y_out) also has the same restriction as (nprocs_x_out).

Let's assume that the number of input files is 4 ([x,y]=[2,2]) and each file has 30 grid cells both in x- and y-directions except for halo. That is, the number of total grid cells of the entire domain is 60 × 60. Since the number of output file is 24 ([x,y]=[4,6]), individual output files have 15 and 10 grid cells in x- and y-directions, respectively.

You can get information of the grid and decomposition in the input file by checking the global attribute in SCALE-netCDF file. For example, you can obtain the header information by using "ncdump" command as follows:

```
$  ncdump -h history_d02.pe000000.nc
```

You will find the global attributes at the end of dumped information.

```
  ......
 // global attributes:
  ......
 :scale_cartesC_prc_rank_x = 0 ;
 :scale_cartesC_prc_rank_y = 0 ;
 :scale_cartesC_prc_num_x = 2 ;
 :scale_cartesC_prc_num_y = 2 ;
  ......
 :scale_atmos_grid_cartesC_index_imaxg = 60 ;
 :scale_atmos_grid_cartesC_index_jmaxg = 60 ;
  ......
```

scale_cartesC_prc_num_x and scale_cartesC_prc_num_y are the sizes of x- and y-directions in the 2-D file topology, respectively. scale_cartesC_prc_rank_x and scale_cartesC_prc_rank_y are the x- and y-positions of this file in the 2-D map, respectively. The rank number starts from 0. scale_atmos_grid_cartesC_index_imaxg and scale_atmos_grid_cartesC_index_jmaxg are the total number of grid cells in x- and y-directions, respectively. These numbers do not contain halo grid. For other attribute information, refer to the table 5.1.1 in Sec. 5.1.1.1.

Convert multiple SCALE-netCDF files to GrADS format file

```
&PARAM_SNO
  basename_in = 'input/history_d02',
  dirpath_out = 'output',
  output_grads = .true.,
  vars = "U", "PREC", "LAND_TEMP",
/
```

If (output_grads) is .true., all of the decomposed data is combined spatially and output into a single file of GrADS binary format. The conversion is applied to the variables specified by (vars). Since the GrADS format file cannot contain the variables with the different vertical layers, SNO outputs individual variables to a different file. The individual file is named as the variable name. The control files are also generated. The converted file is output to ./output directory. Note that (basename_out) is ignored in this conversion. Instead of (basename_out), you can use the output path, i.e., (dirpath_out).

4.6.1.3 Examples of configuration: plugin functions

You can apply the operators, such as time averaging or horizontal/vertical remapping, before outputting the combined/decomposed files.

Temporally averaging

The settings for time averaging are specified by [PARAM_SNOPLGIN_TIMEAVE].

```
&PARAM_SNO
 basename_in = 'input/history_d02',
 basename_out = 'output/history_d02',
 nprocs_x_out = 2,
 nprocs_y_out = 2,
/

&PARAM_SNOPLGIN_TIMEAVE
 SNOPLGIN_timeave_type = 'NUMBER',
 SNOPLGIN_timeave_interval = 4,
/
```

When (SNOPLGIN_timeave_type) is set to 'NUMBER', the data is averaged every equal time interval. The time interval is specified by (SNOPLGIN_timeave_interval). In this example, the variables is averaged every 4 output-steps. The data is decomposed into 2×2, and then is output to 4 files.

The other example is as follows:

```
&PARAM_SNO
 basename_in = 'input/history_d02',
 basename_out = 'output/history_d02',
/

&PARAM_SNOPLGIN_TIMEAVE
 SNOPLGIN_timeave_type = 'MONTHLY',
/
```

Both the file aggregation and the time averaging are applied in this example.
When (SNOPLGIN_timeave_type) is set to 'DAILY', 'MONTHLY', or 'ANNUAL', SNO try to daily, monthly or annual average of the variables, respectively. The date and time of the data are read from the file. If you use different calendar from the default in the simulation, the same setting of [PARAM_CALENDAR] should be added to the configuration file of SNO (Please see Sec. 4.5.4).

Regrid data to the lat-lon coordinate

The settings for regrid of data on the lat-lon coordinate are specified by [PARAM_SNOPLGIN_HGRIDOPE].

```
&PARAM_SNO
 basename_in = 'input/history_d02',
 basename_out = 'output/history_d02',
/

&PARAM_SNOPLGIN_HGRIDOPE
 SNOPLGIN_hgridope_type = 'LATLON',
 SNOPLGIN_hgridope_lat_start = 30.0,
 SNOPLGIN_hgridope_lat_end = 40.0,
 SNOPLGIN_hgridope_dlat = 0.5,
 SNOPLGIN_hgridope_lon_start = 130.0,
 SNOPLGIN_hgridope_lon_end = 140.0,
 SNOPLGIN_hgridope_dlon = 0.5,
/
```

When (SNOPLGIN_hgridope_type) is set to 'LATLON', horizontal remapping to the latitude-longitude grid system is applied. This plugin operator can be available only when the output file is single. The other options in [PARAM_SNOPLGIN_HGRIDOPE] are used to set the domain boundary and grid interval of the output data. The size of longitude grid point nlon is calculated as follows:

$$\text{nlon} = \frac{\text{SNOPLGIN\_hgridope\_lon\_end} - \text{SNOPLGIN\_hgridope\_lon\_start}}{\text{SNOPLGIN\_hgridope\_dlon}} + 1.$$

The result of this calculation is rounded to an integer. Thus, the easternmost grid point may be different from (SNOPLGIN_hgridope_lon_end). The number of latitude grid points is calculated in the same manner as that of the longitude grid points.

You can set the lat-lon domain larger than the original domain size used in the simulation. In remapping process, extrapolation is not allowed. Therefore, a missing value is set at the grid, which has no interpolated value.

Regrid data to different model-level coordinate

The settings for regrid of data on other vertical coordinate are specified by [PARAM_SNOPLGIN_VGRIDOPE]. This plugin operator is available even when the output file is multiple.

```
&PARAM_SNO
 basename_in = 'input/history_d02',
 basename_out = 'output/history_d02',
/

&PARAM_SNOPLGIN_VGRIDOPE
 SNOPLGIN_vgridope_type = 'model',
 SNOPLGIN_vgridope_lev_num = 5,
 SNOPLGIN_vgridope_lev_data = 80.0, 160.0, 400.0, 1000.0, 2000.0,
/
```

When (SNOPLGIN_vgridope_type) in [PARAM_SNOPLGIN_VGRIDOPE] is set to 'model', vertical remapping to the different model-coordinate system is applied. (SNOPLGIN_vgridope_lev_num) is the number of vertical layers, and (SNOPLGIN_vgridope_lev_data) specifies the vertical levels at which you want to convert data. The unit of level is [m].

Regrid data to actual height-level coordinate

```
&PARAM_SNO
 basename_in = 'input/history_d02',
 basename_out = 'output/history_d02',
/

&PARAM_SNOPLGIN_VGRIDOPE
 SNOPLGIN_vgridope_type = 'z',
 SNOPLGIN_vgridope_lev_num = 3,
 SNOPLGIN_vgridope_lev_data = 100.0, 300.0, 500.0,
/
```

When (SNOPLGIN_vgridope_type) in [PARAM_SNOPLGIN_VGRIDOPE] is set to 'z', vertical remapping to the actual height-coordinate system is applied. (SNOPLGIN_vgridope_lev_num)

is the number of vertical layers, and (SNOPLGIN_vgridope_lev_data) specifies the vertical levels at which you want to convert data. The unit of level is [m].

Regrid data to pressure-level coordinate

```
&PARAM_SNO
 basename_in = 'input/history_d02',
 basename_out = 'output/history_d02',
/

&PARAM_SNOPLGIN_VGRIDOPE
 SNOPLGIN_vgridope_type = 'pressure',
 SNOPLGIN_vgridope_lev_num = 3,
 SNOPLGIN_vgridope_lev_data = 850.e+2, 500.e+2, 200.e+2,
/
```

When (SNOPLGIN_vgridope_type) in [PARAM_SNOPLGIN_VGRIDOPE] is set to 'pressure', vertical remapping to the pressure-coordinate system is applied. (SNOPLGIN_vgridope_lev_num) is the number of vertical layers, and (SNOPLGIN_vgridope_lev_data) specifies the vertical levels at which you want to convert data. The unit of level is [Pa]. Note that pressure data (PRES) must be included in the input file.

Part 5

Detailed explanation for advanced use

Chapter 5.1

File IO of SCALE

5.1.1 What is SCALE-netCDF File?

In this section, SCALE-netCDF file, which SCALE directly reads from and writes in, is explained. SCALE employs netCDF (network Common Data Format) as data file format. NetCDF is a software developed by Unidata (http://www.unidata.ucar.edu/), and it enables us to generate files with self-describing and machine-independent data format; for example, the former has a merit to describe variables along with their axis variables in a file and the latter has a merit to treat data without worry which endian is used. Based on the above virtue, SCALE has particular convention (SCALE-netCDF convention). It is mostly following the CF conventions (http://cfconventions.org).

5.1.1.1 Global Attributes

SCALE-netCDF file contains information about data contained in the file, such as those with regards to spatial decomposition as "Global Attributes" (Table 5.1.1).

5.1.1.2 Data in the Halo Region

Whether the file contains the data in the halo region depends on the type of file and its configuration. Note that the halo region that we define here means the halo in the whole calculation region, not the halo in each of local regions.

For the initial (or restart) and boundary data files, the halo data is contained if the lateral boundary conditions are not periodic ((PRC_PERIODIC_X), (PRC_PERIODIC_Y) = .false. in [PARAM_PRC_CARTESC]) or the single file I/O (Section 5.1.1.5) is used ((FILE_AGGREGATE)=.true. in [PARAM_FILE]), otherwise it is not.

On the other hand, for the history data file, the halo data is contained only when the lateral boundary conditions are not periodic and (FILE_HISTORY_CARTESC_BOUNDARY)=.true. in [PARAM_FILE_HISTORY_CARTESC], otherwise it is not. Refer Section 4.2.6 for detail.

5.1.1.3 Axis Variables

SCALE-netCDF file contains axis data. All the axis variables have "long_name" and "units" attributes, which describe description and unit of the variable, respectively. In addition, x, y, xh, and yh variables have attributes for the total number of grids in the whole domain ("size_global"), the start index in the total grid of data in the file ("start_global"), the number of the halo grids at the begin and end in the whole data ("halo_global"), and the number of the halo grids of data in the file ("halo_local").

Table 5.1.2 shows list of the axis data. The coordinate variables have their own dimension; their variable names are the same as their dimension names. The variables with lower-case name are mainly used in the file, while those with the upper-case name describe axes in simulation. Figure

Table 5.1.1: Global attributes in SCALE-netCDF file

Name	Description	Remarks
title	Brief description of data	Value of (FILE_History_TITLE) in [PARAM_FILE_HISTORY]
source	Name of the source software	Value of (FILE_History_SOURCE) in [PARAM_FILE_HISTORY] for history files and (H_SOURCE) in [PARAM_IO] for other files
institution	Data author	Value of (FILE_History_INSTITUTION) in [PARAM_FILE_HISTORY] for history files and (H_INSTITUTE) in [PARAM_IO] for other files
rankid	Rank id of MPI process	PRC_myrank in the model
Conventions	CF convention version	"CF-1.6" for version 5.3
grid_name	Grid type	"cartesC" for SCALE-RM
scale_cartesC_prc_rank_[xy]	Mapping index of the 2D decomposition	Equal to PRC_2Drank(PRC_myrank, i) variable in the model (i=1 for x and 2 for y)
scale_cartesC_prc_num_[xy]	Number of the 2D decomposition	(PRC_NUM_X), (PRC_NUM_Y) in the model
scale_cartesC_prc_periodic_[zxy]	Whether the boundary condition is periodic	.false. and .true. They correspond to (PRC_PERIODIC_X), (PRC_PERIODIC_Y) in the model
scale_atmos_grid_cartesC_index_[ij]maxg	Number of the grids in the global domain	(IMAXG), (JMAXG) in the model
scale_atmos_grid_cartesC_index_kmax	Number of the vertical layer for the atmospheric model	(KMAX) in the model
scale_ocean_grid_cartesC_index_kmax	Number of the vertical layer for the ocean model	(OKMAX) in the model
scale_land_grid_cartesC_index_kmax	Number of the vertical layer for the land model	(LKMAX) in the model
scale_urban_grid_cartesC_index_kmax	Number of the vertical layer for the urban model	(UKMAX) in the model
scale_atmos_grid_cartesC_index_[kij]halo	Number of halo grids	(KHALO), (IHALO), (JHALO) in the model
Calendar	Calendar type	(PARAM_CALENDAR) in the model
time_units	Unit of time	
time_start	Start time	

Refer Section 4.2.6 for (History_TITLE, History_SOURCE, History_INSTITUTION), Section 4.2.3 for (PRC_NUM_X, PRC_NUM_Y, PRC_PERIODIC_X, PRC_PERIODIC_Y) and (KMAX, IMAXG, JMAXG), and Section 4.5.4 for (PARAM_CALENDAR).

and show the horizontal and vertical locations of coordinate variables, respectively. Refer to them with Table at the same time.

The area and volume data of the grids are also in the file as "cell_area**" and "cell_volume**", respectively. The "cell_measures" attribute of each variables specifies the corresponding area or volume data.

The map projection data is contained as a non-dimensional variable whose name is specified by "grid_mapping" attribute of the variables.

The information of relationship with the staggered grids is contained as an attribute and a non-dimensional variable, based on SGRID conventions (https://github.com/sgrid/sgrid). The non-dimensional variable's name is specified by the "grid" attribute of each variable.

In the file, there are also the surface elevation data and land mask data as "topo" and "lsmask", respectively.

Table 5.1.2: Axis data in SCALE-netCDF .

Coordinate variables	
name	description
Horizontal axis & time axis: Common	
x	full level position in x-direction of the data in the file
x_bnds	full level cell boundary in x-direction of the data in the file
xh	half level position in x-direction of the data in the file
xh_bnds	half level cell boundary in x-direction of the data in the file
y	full level position in y-direction of the data in the file
y_bnds	full level cell boundary in y-direction of the data in the file
yh	half level position in y-direction of the data in the file
yh_bnds	half level cell boundary in y-direction of the data in the file
time	time information
time_bnds	time boundary information
CX	full level grid position in x-direction for the local region (inc. halo grids)
FX	half level grid position in x-direction for the local region (inc. halo grids)
CDX	full level grid spacing in x-direction (inc. halo grids)
FDX	half level grid spacing in x-direction (inc. halo grids)
CY	full level grid position in y-direction for the local region (inc. halo grids)
FY	half level grid position in y-direction for the local region (inc. halo grids)
CDY	full level grid spacing in y-direction (inc. halo grids)
FDY	half level grid spacing in y-direction (inc. halo grids)
CXG	full level grid position in x-direction for the whole region (inc. halo grids)
FXG	half level grid position in x-direction for the whole region (inc. halo grids)
CYG	full level grid position in y-direction for the whole region (inc. halo grids)
FYG	half level grid position in y-direction for the whole region (inc. halo grids)
Vertical axis : Atmosphere	
z	full level position in z-direction in the file
z_bnds	full level cell boundary in z-direction in the file
zh	half level position in z-direction in the file
zh_bnds	half level cell boundary in z-direction in the file
CZ	full level grid position in z-direction (inc. halo grids)
FZ	half level grid position in z-direction (inc. halo grids)
CDZ	full level grid spacing in z-direction (inc. halo grids)
FDZ	half level grid spacing in z-direction (inc. halo grids)
Vertical axis : Ocean	
oz	full level position in z-direction of the ocean data in the file
oz_bnds	full level cell boundary in z-direction of the ocean data in the file
ozh	half level position in z-direction of the ocean data in the file
ozh_bnds	half level cell boundary in z-direction of the ocean data in the file
OCZ	full level grid position in z-direction of the ocean model

OFZ	half level grid position in z-direction of the ocean model
OCDZ	full level grid spacing in z-direction of the ocean model
Vertical axis : Land	
lz	full level position in z-direction of the land data in the file
lz_bnds	full level cell boundary in z-direction of the land data in the file
lzh	half level position in z-direction of the land data in the file
lzh_bnds	half level cell boundary in z-direction of the land data in the file
LCZ	full level grid position in z-direction of the land model
LFZ	half level grid position in z-direction of the land model
LCDZ	full level grid spacing in z-direction of the land model
Vertical axis : Urban canopy	
uz	full level position in z-direction in the file
uz_bnds	full level cell boundary in z-direction in the file
uzh	half level position in z-direction in the file
uzh_bnds	half level cell boundary in z-direction in the file
UCZ	full level grid position in z-direction
UFZ	half level grid position in z-direction
UCDZ	full level grid spacing in z-direction

Other axis variables (1D)

name	description
CBFZ	buffer factor at CZ
FBFZ	buffer factor at FZ
CBFX	buffer factor at CX for the local region
FBFX	buffer factor at FX for the local region
CBFY	buffer factor at CY for the local region
FBFY	buffer factor at FY for the local region
CBFXG	buffer factor at CXG for the whole region
FBFXG	buffer factor at FXG for the whole region
CBFYG	buffer factor at CYG for the whole region
FBFYG	buffer factor at FYG for the whole region

Other axis variables (2D)

name	description
lon	longitude at (y, x)
lon_uy	longitude at (y, xh)
lon_xv	longitude at (yh, x)
lon_uv	longitude at (yh, xh)
lat	latitude at (y, x)
lat_uy	latitude at (y, xh)
lat_xv	latitude at (yh, x)
lat_uv	latitude at (yh, xh)

Other axis variables (3D)

name	description
height	heigth at (z, y, x) in history file or at (y, x, z) in restart/initial file
height_xyw	height at (zh, y, x) in history file or at (y, x, zh) in restart/initial file
height_xvz	height at (z, yh, x) in history file or at (yh, x, z) in restart/initial file
height_uyz	height at (z, yh, xh) in history file or at (hy, x, z) in restart/initial file
height_xvw	height at (zh, yh, x) in history file or at (yh, x, zh) in restart/initial file
height_uyw	height at (zh, y, xh) in history file or at (y, xh, zh) in restart/initial file
height_uvz	height at (z, yh, xh) in history file or at (yh, xh, z) in restart/initial file
height_uvw	height at (zh, yh, xh) in history file or at (yh, xh, zh) in restart/initial file

Table 5.1.3: Functionality of each netCDF versions.

	multiple time interval	Compression	Read netCDF 3 file	Read netCDF 4 file	Read single file
netCDF 3	NG	NG	OK	NG	NG*
netCDF 4	OK	OK	OK	OK	NG*
pnetCDF	NG	NG	NG*	NG	OK

(*) it is OK if the number of total processes is 1.

5.1.1.4 Data Variables

Data variables have attributes of the undefined value "_FillValue" and the missing value "missing_value" as well as "long_name" and "untis". Data structure in the initial (restart) data and boundary data files is identical to the array in the model, that is the z-x-y order. On the other hand, it in the history data file is the x-y-z order. This is true for 3D axis variables. See also Table 5.1.2.

5.1.1.5 Single File I/O

As default, all the data files are output by each process, namely file I/O is separated. If you compile SCALE-RM with pnetCDF by specifying an environmental variable as SCALE_ENABLE_PNETCDF=T, the data from all processes can be combined into a single file (see Section 2.2.1). To do this, (FILE_AGGREGATE) in [PARAM_FILE] should be set to .true.. Instead, you can switch on/off of the single I/O for individual file types: history, init/restart, topography, and land use files. For the history file, set (FILE_HISTORY_AGGREGATE)=.true. in [PARAM_FILE_HISTORY]. For the init/restart file, set (RESTART_(IN|OUT)_AGGREGATE)=.true. in [PARAM_RESTART], or ((MODELNAME)_RESTART_(IN|OUT)_AGGREGATE)=.true. in [PARAM_(MODELNAME)_VARS] where (MODELNAME) is "ATMOS", "OCEAN", "LAND", or "URBAN". For the topography and land use files, set (TOPOGRAPHY_(IN|OUT)_AGGREGATE)=.true. in [PARAM_TOPOGRAPHY], and (LANDUSE_(IN|OUT)_AGGREGATE)=.true. in [PARAM_LANDUSE], respectively.

5.1.1.6 Restrictions with NetCDF 3

If SCALE is compiled with netCDF version 3, the following restrictions exist.

- Variables with different time intervals cannot stored in the same file.

- Compression of data cannot be used.

- File of netCDF 4 format cannot be read.

If you want to output variables with different time intervals, set (BASENAME) in [HISTORY_ITEM] to output them to different files.

Note that the restrictions of the multiple time interval and compression are placed when the single file I/O is used, because pnetCDF is based on netCDF 3 file format.

## Horizontal grid positions for all the components
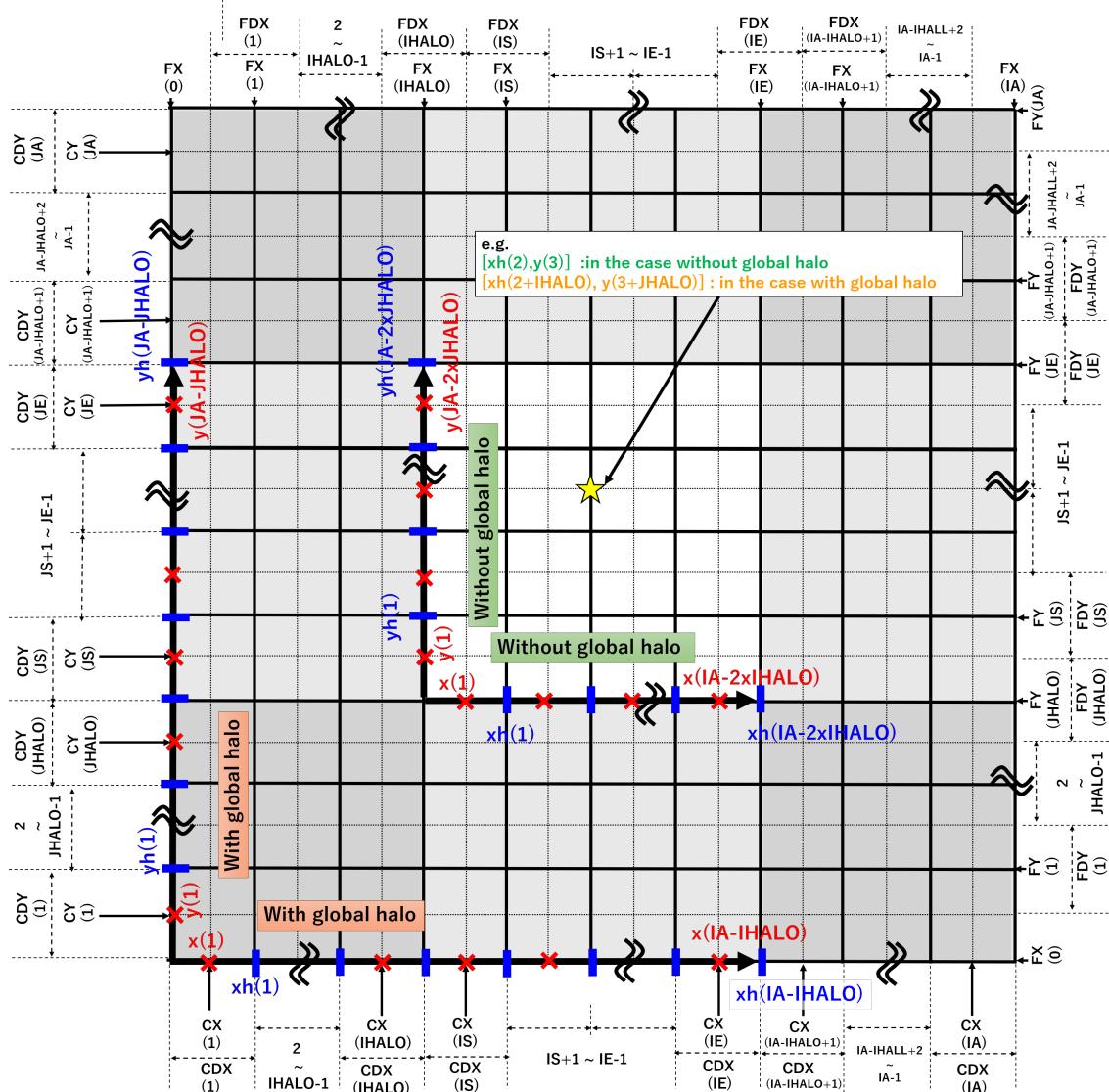
Figure 5.1.1: Horizontal coordinate in SCALE-netCDF file

# Vertical grid positions for each component and physical meanings
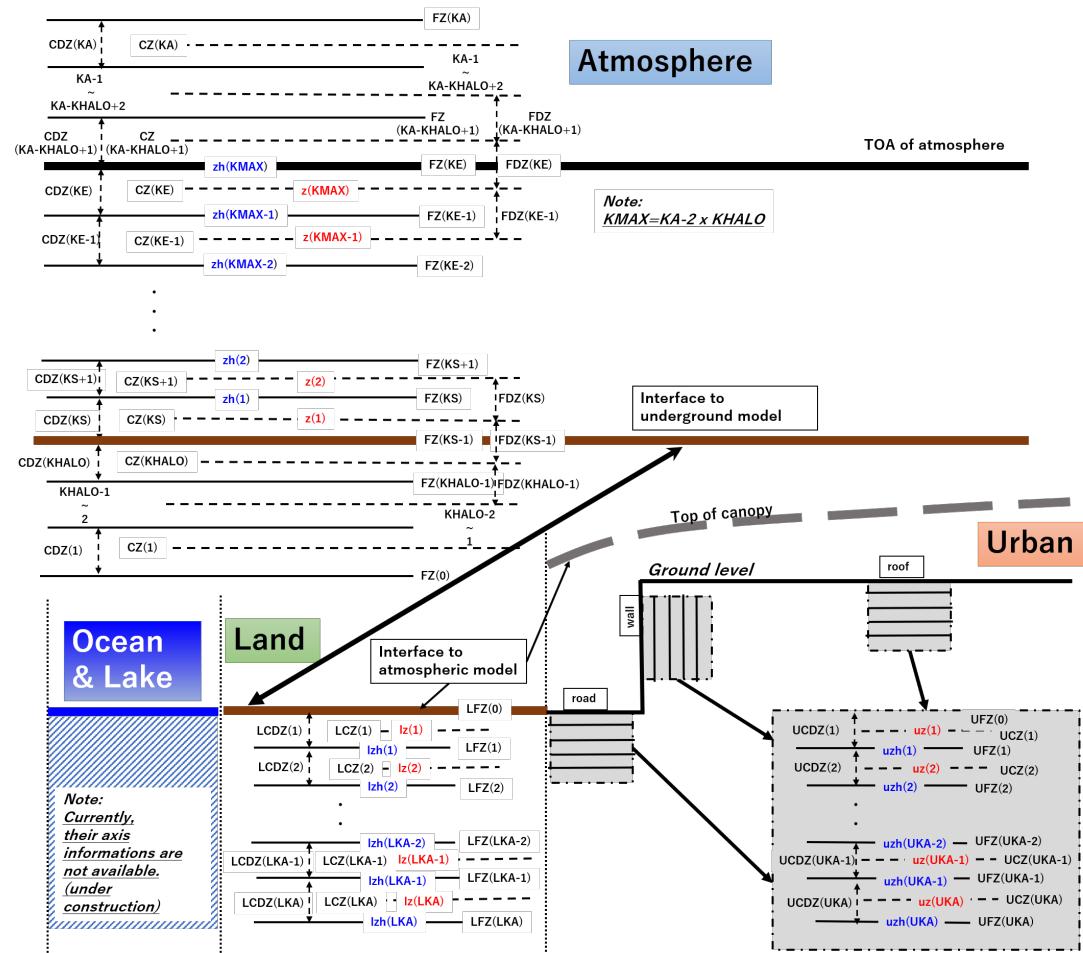
Figure 5.1.2: Vertical coordinate in SCALE-netCDF file

Chapter 5.2

How to use user defined program

5.2.1 Module for user defined settings

SCALE-RM prepares many options, which users can specify as namelist parameters, to meet users' demand for their calculations. However, when there is no option you desire, you can modify model variables directly as you want by programming using the module for users, namely `mod_user`. This section describes what is `mod_user` module and how to use it.

5.2.1.1 What is mod_user module?

The default `mod_user` module is provided in `scale-5.5.1/scale/scale-rm/src/user/mod_user.F90`. Prepare your own `mod_user.F90` and then compile it instead of the default one.

The module must contain the following subroutines:

```
subroutine USER_tracer_setup
subroutine USER_setup
subroutine USER_mkinit
subroutine USER_update
subroutine USER_calc_tendency
```

This is the sequence of execution of the scale-rm.

```
Initial setup
  IO setup
  MPI setup
  Grid settings
  Setup of administrator for dynamics and physics schemes
  Tracer setup
  USER tracer setup
  Setup topography, land
  Setup of vars and drivers for dynamics and physics schemes
  USER setup
Main routine
  Time advance
  Ocean/Land/Urban/Atmos update
  User update
  Output restart
  Calculation of tendency in Atmos/Ocean/Land/Urban
  Calculation of tendency
  History output
```

The timings for calling individual subroutines of the `mod_user` are indicated by blue color. The `USER_mkinit` is called in the init program `scale-rm_init`.

Since the subroutines in `mod_user` are basically called after handling each process, you can replace any settings and variables as you need. You can also add several tracers, such as passive tracers, in `USER_tracer_setup`. There are several examples of `mod_user.F90` in the test cases (`scale-5.5.1/scale-rm/test/case`).

5.2.1.2 Compile

To compile SCALE-RM with your `mod_user.F90`, you can use the Makefile in the test cases as follows.

```
$ cd scale-5.5.1/scale-rm/test/case
$ mkdir -p your_dir/exp_name
$ cd your_dir/exp_name
$ cp ../../advection/500m/Makefile .
```

Copy your `mod_user.F90` to this directory.

```
$ make
```

Chapter 5.3

How to use scale library

5.3.1   User Program Using SCALE

SCALE is a collection of subroutines. The subroutines can be used in an arbitrary program. The library file is created as "`scalelib.a`" under `scale-5.5.1/lib/` after compiling.

The following is a template when you use SCALE in your program.

```
program template
 use scalelib
 implicit none

 call SCALE_init

 ! user instructions

 call SCALE_finalize

 stop
end program template
```

The following is a pseudo program calculating convective available potential energy (CAPE) from atmospheric quantities read from a file. You needs to use necessary modules and prepare required input variables before main part. Note that this is only a part of the program.

```
use scale_const, only: &
 Rdry => CONST_Rdry, Rvap => CONST_Rvap, CPdry => CONST_CPdry
use scale_atmos_hydrometeor, only: &
 CPvap => CP_VAPOR, CL => CP_WATER
use scale_file, only: &
 FILE_open, FILE_read, FILE_close
use scale_atmos_adiabat, only: &
 ATMOS_ADIABAT_setup, ATMOS_ADIABAT_cape
 :
 real(8) :: z(kmax,imax,jmax), zh(0:kmax,imax,jmax)
 real(8) :: temp(kmax,imax,jmax), pres(kmax,imax,jmax), dens(kmax,imax,jmax)
 real(8) :: qv(kmax,imax,jmax), qc(kmax,imax,jmax), qdry(kmax,imax,jmax)
 real(8) :: rtot(kmax,imax,jmax), cptot(kmax,imax,jmax)
 real(8) :: cape(imax,jmax), cin(imax,jmax)
 real(8) :: lcl(imax,jmax), lfc(imax,jmax), lnb(imax,jmax)
 :
 call FILE_open( basename, fid ) ! open file
 call FILE_read( fid, 'height', z(:,:,:) ) ! read full-level height data
 call FILE_read( fid, 'height_xyw', zh(:,:,:) ) ! read half-level height data
 call FILE_read( fid, 'T', temp(:,:,:) ) ! read temperature data
 : ! read PRES, DENS, QV, QC
 call FILE_close( fid )

 ! calculate some variables required to calculate CAPE
 qdry(:,:,:) = 1.0D0 - qv(:,:,:) - qc(:,:,:) ! mass ratio of dry air
 rtot(:,:,:) = qdry(:,:,:) * Rdry + qv(:,:,:) * Rvap ! gas constant
 cptot(:,:,:) = qdry(:,:,:) * CPdry + qv(:,:,:) * CPvap + ql(:,:,:) * CL ! heat capacity

 call ATMOS_ADIABAT_setup
 call ATMOS_ADIABAT_cape( kmax, 1, kmax, imax, 1, imax, jmax, 1, jmax, & ! array size
                          k0, & ! vertical index from which a parcel is started
                          dens(:,:,:), temp(:,:,:), pres(:,:,:), & ! input
                          qv(:,:,:), qc(:,:,:), qdry(:,:,:), & ! input
                          rtot(:,:,:), cptot(:,:,:), & ! input
                          z(:,:,:), zh(:,:,:), & ! input
                          cape(:,:), cin(:,:), lcl(:,:), lfc(:,:), lnb(:,:) ) ! output
```

In the reference manual (see Section 5.3.2), you can find list of the subroutine which is available and details of the subroutines. There are sample programs for analysis of SCALE-RM history file at the `scale-5.5.1/scalelib/test/analysis` directory.

5.3.1.1 Compile

Before compiling your program using SCALE , you need to compile SCALE .
```
$ cd scale-5.5.1/scalelib/src
$ make
```
To compile your program, you need to link `libscale.a` located at `scale-5.5.1/lib`. You also have to tell the path of the module files to the compiler. The module path is `scale-5.5.1/include`. The option to specify the module path depends on compiler. You can find the option in the sysdep file as `MODDIROPT` variable (refer Section 2.2.1).
```
 $ ${FC} your-program ${MODDIROPT} scale-top-dir/include \\
          `nc-config --cflags` -Lscale-top-dir/lib -lscale `nc-config --libs`
```
You can also use the Makefile in the samples to compile your program as follows.
```
$ cd scale-5.5.1/scalelib/test/analysis
$ mkdir your_dir
```

```
$ cd your_dir
$ cp ../horizontal_mean/Makefile .
```
Copy your program file to this directory.
Edit Makefile (BINNAME = your_program_name).
```
$ make
```

5.3.2 Reference manual

Reference manual for the SCALE subroutines is available on
https://scale.riken.jp/archives/5.5.1/index.html. This reference manual is generated by
doxgen (http://www.doxygen.org/).

The reference has the following information:

- Subroutines

- NAMELIST parameters

- History (output) variables

5.3.2.1 Subroutines

The subroutine information contains the description, arguments, and call graph. The source code
of the subroutine can be also seen. Users can find subroutines through "Module List" or "File
List", which are linked at the top page or the top menu. The module list has brief description of
each module.

The prefix of module name is "scale_" for SCALE and "mod_" for SCALE-RM , respectively.
The file name is the module name with suffix of ".F90". The name of the subroutines consists of
a module name except the prefix and the name explaining function.
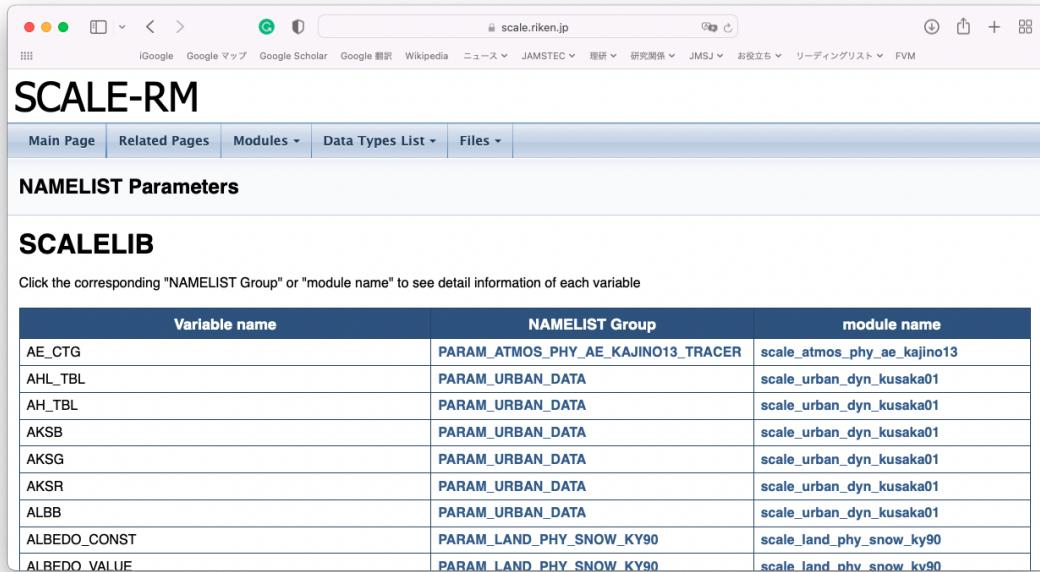
For example, the subroutine of `ATMOS_ADIABAT_cape` is contained
in the module of `scale_atmos_adiabat` in the file of `scale_atmos_adiabat.F90`.

```
scale_atmos_adiabat.F90

module scale_atmos_adiabat
  ... ...
 contains
 !----------------------------------------
         subroutine atmos_adiabat_cape( &
                 Kstr, &
                 DENS, &
                 ...
```

5.3.2.2 NAMELIST Parameters

You can see the NAMELIST parameter list throught the link at the top page of the reference manual, or the directly at `https://scale.riken.jp/archives/5.5.1/d5/d8a/namelist.html`. The list has the parameter name, NAMELIST group name, and module name where the variable is defined. The parameters are sorted by the variable name. The detail information of parameters can be see by clicking the NAMELIST group name or module name.

Figure 5.3.1: The image of the NAMELIST parameters list on the SCALE web page

5.3.2.3 History (output) Variables

You can see the history (output) variable list through the link at the top page of the reference manual, or the directly at https://scale.riken.jp/archives/5.5.1/dc/dd1/history.html. The list has the variable name, brief description, and module name where the variable is registered for the history data. The variables are sorted by the module name. The detail information of variables can be see by clicking the module name.


Figure 5.3.2: The image of the history variables list on the SCALE web page

Chapter 5.4

How to Conduct Bulk Job

5.4.1 What's Bulk Job

SCALE-RM has a functionality for bulk jobs that allows simultaneous multiple execution of independent runs. This function is useful for the parameter-sweep experiment, the ensemble experiment in different initial conditions, the time-slice climate simulation, and so on.

The bulk job function can be used not only for model simulation (`scale-rm`), but also for the generation of topographical data, land-use data, and initial/boundary data. Note that the generation of topographical/land-use data by this function is limited to the case without a topography copy function (See 4.2.8.1).

In the following explanation, an independent execution in the bulk job is called a "sub-job". Three two-domain nesting experiments are taken up as an example. This set of experiments consists of three sub-jobs with different integration periods and calculation domain. (NUM_DOMAIN, PRC_DOMAINS, CONF_FILES) in [PARAM_LAUNCHER] in the file `launch.conf` (refer to Section 4.2.8.3) must be the same in all configurations. The other settings, such as integration time, the scheme used, and the number of grids per MPI process, do not need to be the same among the sub-jobs.

5.4.2 Settings for Bulk Job

Since the bulk job function is an extension of the division and redistribution of MPI processes used in online nesting, file `launch.conf` is required to launch the job. Even in case online nesting and bulk job functions are used together, only one file `launch.conf` is required. Below is an example of such a case.

```
&PARAM_LAUNCHER
  NUM_BULKJOB = 3,                              number of sub-jobs
  NUM_ITERATION_BULK = 1,                       number of iterations to execute sub-jobs
  BULKJOB_START_DIRNUM = 0,                     start number of the job directory
  ADD_BULKJOB_PATH = .false.,                   add job directory to file names
  NUM_DOMAIN = 2,                               number of nesting domains
  PRC_DOMAINS = 9, 36,                          number of total process in each domain
  CONF_FILES = run.d01.conf, run.d02.conf,      name of configulation files
  LOG_SPLIT = .false.,                          log-output for mpi splitting?
  COLOR_REORDER = .true.,                       coloring reorder for mpi splitting?
  FAILURE_PRC_MANAGE = .false.,                 use failure process management?
  NUM_FAIL_TOLERANCE = 1,                       tolerance number of failure processes
  FREQ_FAIL_CHECK = 5,                          FPM polling frequency per DT
/
```

The number of sub-jobs is specified by (NUM_BULKJOB). For single-domain experiments (no nesting), specify (NUM_DOMAIN) = 1.
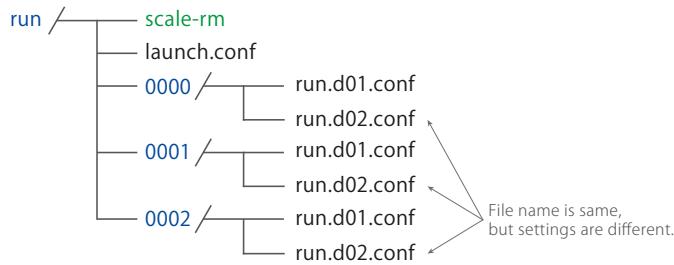
Figure 5.4.1: Structure of directories at the execution of SCALE-RM with bulk job. Numbers such as "0000" are the directory names and called the job directory. All necessary configuration files to execute sub jobs must be prepared in each job directory.

The job directory name is a four-degit number, which by default corresponds to a job ID starting with 0. You can change the starting number by specifing (BULKJOB_START_DIRNUM).

By default, the file names in the sub-job configuration files, such as init.conf and run.conf, must contain the job directory (e.g., 0000/history.nc). If (ADD_BULJJOB_PATH) is specified as .true., the job directory name will be added to all file names in the configuration files except the absolute path.

If your computer does not have enough resources to run all the sub-jobs at once, you can divide the sub-jobs into multiple groups. The number of sub-jobs to be executed at one time will be (NUM_BULKJOB) / (NUM_ITERATION_BULK).

The other configurations are described in Section 4.2.8.3.

5.4.3   Failure Process Management

In the SCALE-RM bulk job system, a failure process management (FPM) tool is available. FPM watches the job groups with a polling interval. If some jobs are unfortunately aborted, all other jobs are not aborted until the number of failure jobs reaches to the limit number. (FAILURE_PRC_MANAGE) is a switch for use FPM tool. (NUM_FAIL_TOLERANCE) and (FREQ_FAIL_CHECK) are setting parameters for limit number of failure jobs and interval of checking failure job, respectively. Note, this version of FPM tool is available only for single domain calculation, not available of full system for online-nesting calculation. If you want to use FPM tool even in online-nesting calculation, (NUM_FAIL_TOLERANCE) must be equal to the number of all jobs.

5.4.4   Preparation for Bulk Job

Prior to the execution of bulk jobs, prepare as many directories as the number of sub-jobs, which are called "sub-directories." As in Fig. 5.4.1, these correspond to 0000/ 0001/ 0002/. The directory name is assigned as a four-digit number starting from zero. In each directory, all the necessary files (configuration files, input files, and output directories) must be prepared. Note that the path to directories or files specified in the configuration files must be correctly set up as explained below. An excerpt of run.d01.conf for job 0000 are as follows:

```
&PARAM_IO
 IO_LOG_BASENAME = "0000/LOG_d01",
/

&PARAM_RESTART
 RESTART_OUTPUT = .true.,
 RESTART_OUT_BASENAME = "0000/restart_d01",
 RESTART_IN_BASENAME = "../init/0000/init_d01_00013046400.000",
/

&PARAM_GRAPHY
 TOPOGRAPHY_IN_BASENAME = "../pp/0000/topo_d01",
/

&PARAM_LANDUSE
 LANDUSE_IN_BASENAME = "../pp/0000/landuse_d01",
/

&PARAM_ATMOS_BOUNDARY
 ~ ... ~
 ATMOS_BOUNDARY_IN_BASENAME = "../init/0000/boundary_d01",
 ~ ... ~
/

&PARAM_FILE_HISTORY
 FILE_HISTORY_DEFAULT_BASENAME = "0000/history_d01",
 ~ ... ~
/
```

As shown in Fig. 5.4.1, job directories exist in the same hierarchy as the directory of the executable binary. That is, a configuration file exists under each job directory, whereas the input files and the output directories must be described as relative paths from the location of the executable binary. The output directory in the 0000 experiment is `0000/` and the name of output files are as `0000/***`. Note that when the file name is common to all experiments without adding the name of job directory, the output is written to the same file, and the data disappear as a result.

5.4.5   Execution of Bulk Job

At the execution of bulk jobs, the total number of MPI processes is specified as

```
 $ mpirun  -n  135  ./scale-rm  launch.conf
```

In this example, the number of processes per sub-job is 45 ($= 9 + 36$) and the total number of processes used for three sub-jobs is 135. The message providing information regarding the division of MPI processes is written to the LOG files after the SCALE logo. The following example is the log output for process 0 of Domain 1:

```
+++++ start making Cartesian topology
*** UNIVERSAL_COMM_WORLD : 0           ; different by execution environment
*** total process [UNIVERSAL] : 135
*** my process ID [UNIVERSAL] : 36
*** master rank? [UNIVERSAL] : F
*** GLOBAL_COMM_WORLD : 3              ; different by execution environment
*** total process [GLOBAL] : 45
*** my process ID [GLOBAL] : 36
*** master rank? [GLOBAL] : F
*** LOCAL_COMM_WORLD : 4              ; different by execution environment
*** total process [LOCAL] : 9
*** my process ID [LOCAL] : 0
*** master rank? [LOCAL] : T
*** ABORT_COMM_WORLD : 0
*** master rank ID [each world] : 0
```

The items belonging to [LOCAL], [GLOBAL], and [UNIVERSAL] are the descriptions, respectively, of the process group in the domain, the nesting group, and the job group. The UNIVERSAL group includes the GLOBAL group and the GLOBAL group includes the LOCAL group. total process represents the total number of processes in each group and my process ID is the ID of the process in the group.

It can be confirmed in this case that 1) since total process [UNIVERSAL] is 135, all these processes are completed executed; 2) since total process [GLOBAL] is 45, these 45 processes are used in a sub-job; 3) since the log file for Domain 1 is shown in this example, it is correct to describe total process [LOCAL] as 9. If you see the log message for Domain 2, it is 39. my process ID [UNIVERSAL] is the process number corresponding to the LOG and history files. Since this notation is the same as in the abnormal completion of execution, you can immediately recognize the processes in which errors occur during executing a considerable number of sub-jobs.

Bibliography

A. C. M. Beljaars. The parameterization of surface fluxes in large-scale models under free convection. *Quarterly Journal of the Royal Meteorological Society*, 121:255–270, 1994. doi: 10.1002/qj.49712152203.

A. C. M. Beljaars and A. A. M. Holtslag. Flux parameterization over land surfaces for atmospheric models. *J. Appl. Meteor.*, 30:327–341, 1991.

A. R. Brown, S. H. Derbyshire, and P. J. Mason. Large-eddy simulation of stable atmospheric boundary layers with a revised stochastic subgrid model. *Quarterly Journal of the Royal Meteorological Society*, 120:1485–1512, 1994.

Committee on Space Research; NASA National Space Science Data Center. COSPAR International Reference Atmosphere (cira-86): Global climatology of atmospheric parameters, 2006.

J. W. Deardorff. Stratocumulus-capped mixed layers derived from a three-dimensional model. *Boundary-Layer Meteorology*, 18:495–527, 1980.

A. M. Golovin. The solution of the coagulation equation for cloud droplets in a rising air current. *Izv. Geophys. Ser*, 5:482–487, 1963.

J. S. Kain. The kain-fritsch convective parameterization: An update. *J. Appl. Meteor.*, 43:170–181, 2004.

J. S. Kain and J. M. Fritsch. A one-dimensional entraining/detraining plume model and its application in convective parameterization. *Journal of the Atmospheric Sciences*, 47:2784–2802, 1990.

E. Kessler. On the distribution and continuity of water substance in atmospheric circulation. *Meteorological Monograph*, 10:1–84, 1969.

Barry Koren. *A robust upwind discretization method for advection, diffusion and source terms.* Centrum voor Wiskunde en Informatica Amsterdam, 1993.

H. Kusaka, H. Kondo, Y. Kikegawa, and F. Kimura. A simple single-layer urban canopy model for atmospheric models: comparison with multi-layer and slab models. *Boundary-Layer Meteorol.*, 101:329–358, 2001.

D. K. Lilly. On the numerical simulation of buoyant convection. *Tellus*, 14:148—-171, 1962.

A. Long. Solutions to the droplet collection equation for polynomial kernels. *Journal of the Atmospheric Sciences*, 31:1041–1052, 1974.

J.-F. Louis. A parametric model of vertical eddy fluxes in the atmosphere. *Boundary-Layer Meteorol.*, 17:187–202, 1979. doi: 10.1007/BF00117978.

G. L. Mellor and T. Yamada. Development of a turbulence closure model for geophysical fluid problems. *Rev. Geophys. Space Phys.*, 20:851—875, 1982.

M. J. Miller, A. C. M. Beljaars, and T. N. Palmer. The sensitivity of the ecmwf model to the parameterization of evaporation from the tropical oceans. *Journal of Climate*, 5(5):418–434, 1992.

I.-J. Moon, I. Ginis, T. Hara, and B. Thomas. A physics-based parameterization of air-sea momentum flux at high wind speeds and its impact on hurricane intensity predictions. *Mon. Weather Rev.*, 135:2869–2878, 2007. doi: 10.1175/MWR3432.1.

T. Nakajima, M. Tsukamoto, Y. Tsushima, A. Numaguti, and T. Kimura. Modeling of the radiative process in an atmospheric general circulation model. *Applied Optics*, 39:4869–4878, 2000. doi: 10.1364/AO.39.004869.

M. Nakanishi and H. Niino. An improved mellor-yamada level-3 model with condensation physics: Its design and verification. *Bound-Lay. Meteorol.*, 112:1–31, 2004.

M. Nakanishi and H. Niino. Development of an imprived turbulence closure model for the atmospheric boundary layer. *Journal of the Meteorological Society of Japan*, 87:895–912, 2009. doi: 10.2151/jmsj.87.895.

M. Narita and S. Ohmori. Improving precipitation forecasts by the operational nonhydrostatic mesoscale model with the kain-fritsch convective parameterization and cloud microphysics. *12th Conference on Mesoscale Processes*, 6-9 August 2007, Waterville Valley conference & event center:available from https://ams.confex.com/ams/12meso/techprogram/paper_126017.htm, 2007.

S. Nishizawa and Y. Kitamura. A surface flux scheme based on the Monin-Obukhov similarity for finite volume models. *J. Adv. Model. Earth Syst.*, 10:3159–3175, 2018. doi: 10.1029/2018MS001534.

S. Nishizawa, H. Yashiro, Y. Sato, Y. Miyamoto, and H. Tomita. Influence of grid aspect ratio on planetary boundary layer turbulence in large-eddy simulations. *Geosci. Model Dev.*, 8(10): 3393–3419, 2015. doi: 10.5194/gmd-8-3393-2015.

Y. Ogura and H.-R. Cho. Diagnostic determination of cumulus cloud populations from observed large-scale variables. *Journal of the atmospheric sciences*, 30(15):1276–1286, 1973.

Katsuyuki V Ooyama. A dynamic and thermodynamic foundation for modeling the moist atmosphere with parameterized microphysics. *Journal of the atmospheric sciences*, 58(15):2073–2102, 2001.

J-L Redelsperger, PRA Brown, F Guichard, C How, M Kawasima, S Lang, Th Montmerle, K Nakamura, K Saito, C Seman, et al. A gcss model intercomparison for a tropical squall line observed during toga-coare. i: Cloud-resolving models. *Quarterly Journal of the Royal Meteorological Society*, 126(564):823–863, 2000.

J. J. Remedios, R. J. Leigh, A. M. Waterfall, D. P. Moore, H. Sembhi, I. Parkes, J. Greenhough, M. P. Chipperfield, and D. Hauglustaine. Mipas reference atmospheres and comparisons to v4.61/v4.62 mipas level 2 geophysical data sets. *Atmos. Chem. Phys. Discuss.*, 7:9973–10017, 2007. doi: doi:10.5194/acpd-7-9973-2007.

Y. Sato, S. Nishizawa, H. Yashiro, Y. Miyamoto, Y. Kajikawa, and H. Tomita. Impacts of cloud microphysics on trade wind cumulus: which cloud microphysics processes contribute to the diversity in a large eddy simulation? *Progress in Earth and Planetary Science*, 2(1):1–16, 2015. doi: 10.1186/s40645-015-0053-6.

Team SCALE. *The detail formulation of SCALE-RM*. RIKEN/R-CCS, 2015.

A. Scotti, C. Meneveau, and D. K. Lilly. Generalized smagorinsky model for anisotropic grids. *Physics of Fluids A*, 5:2306–2308, 1993.

T. Seiki and T. Nakajima. Aerosol effects of the condensation process on a convective cloud simulation. *Journal of the Atmospheric Sciences*, 71:833–853, 2014.

M. Sekiguchi and T. Nakajima. A k-distribution-based radiation code and its computational optimization for an atmospheric general circulation model. *J. of Quantitative Spectroscopy and Radiative Transfer*, 109:2779—-2793, 2008.

J. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Weather Rev.*, 91:99—-164, 1963.

K. Suzuki, T. Nakajima, T. Y. Nakajima, and A. P. Khain. A study of microphysical mechanisms for correlation patterns between droplet radius and optical thickness of warm clouds with a spectral bin microphysics cloud model. *Journal of the Atmospheric Sciences*, 67(4):1126–1141, 2010.

H. Tomita. New microphysical schemes with five and six categories by diagnostic generation of cloud ice. *Journal of the Meteorological Society of Japan*, 86:121–142, 2008.

H. Tomita. Analysis of spurious surface temperature at the atmosphere-land interface and a new method to solve the surface energy balance equation. *J. Hydrometeorology*, 10:833–844, 2009. doi: 10.1175/2008JHM1080.1.

I Uno, X-M Cai, DG Steyn, and S Emori. A simple extension of the louis method for rough surface layer modelling. *Boundary-Layer Meteorology*, 76(4):395–409, 1995.

Louis J Wicker and William C Skamarock. Time-splitting methods for elastic models using forward time schemes. *Monthly Weather Review*, 130(8):2088–2097, 2002.

D. K. Wilson. An alternative function for the wind and temperature gradients in unstable surface layers. *Bound-Lay. Meteorol.*, 99:151—-158, 2001.

S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31:335–362, 1979. doi: 10.1016/0021-9991(79)90051-2.

Part 6

Appendix

Appendix A

Frequently asked questions

In this chapter, frequently asked questions are shown as specific exercises, and answers to them are provided.

Questions

1. **How is the number of MPI processes changed while maintaining the computational domain?**
   Change the MPI process number from 4-MPI parallel to 6-MPI parallel for the tutorial case of the real atmospheric experiment in Chap. 3.2. (Reference: Section 4.2.3.3 and 4.2.3.1).

2. **How is the computational domain altered while maintaining the number of MPI processes?**
   Extend the computational domain to four-thirds of its original size along the x direction and shrink it to two-thirds along the y direction for the tutorial case of the real atmospheric experiment in Chap. 3.2 (Reference: Section 4.2.3.1).

3. **How is horizontal grid interval changed while maintaining the computational domain?**
   Change the horizontal grid interval from the default value to 5 km for the tutorial case of the real atmospheric experiment in Chap. 3.2 (Reference: Section 4.2.3.1, 4.2.3.2, 4.2.3.5, 4.2.5, 4.2.6).

4. **How is the location of computational domain altered?**
   Change the domain center from the default value to a longitude of $135°45.4'$ and a latitude of $35°41.3'$ while maintaining the domain size for the tutorial case of the real atmospheric experiment in Chap. 3.2 (Reference: Section 4.2.4).

5. **How is the integration time changed?**
   Change the integration period from 6 hours to 12 hours for the tutorial case of the real atmospheric experiment in Chap. 3.2 (Reference: Section 4.2.5).

6. **How are output variables added and their output interval changed?**
   Change the output interval from the default to 30 min, and add the downward and upward shortwave radiations at the surface as output variables for the tutorial case of the real atmospheric experiment in Chap. 3.2. (Reference: Section 4.2.6, Section 5.3.2).

7. **How does the model restart?**
   Using the tutorial of the real atmospheric experiment in Chap. 3.2, execute 3-hour integration. Then execute another 3 hours of integration using the restart files created in the first integration. (Reference: Section 4.2.7 and 4.1.2).

Answer

1. **How is the number of MPI processes changed while maintaining the computational domain?**
   (PRC_NUM_X, PRC_NUM_Y) in [PARAM_PRC] are changed. If the below conditions are satisfied, your answer is correct:

   $$\text{Number of MPIs parallel} = \text{PRC\_NUM\_X} \times \text{PRC\_NUM\_Y} = 6$$
   $$\text{IMAXG} \bmod \text{PRC\_NUM\_X} = 0$$
   $$\text{JMAXG} \bmod \text{PRC\_NUM\_Y} = 0$$

2. **How is the computational domain altered while maintaining the number of MPI processes?**
   If the number of grid points increases n times, the domain size also increases n times. Thus, only (IMAXG, JMAXG) in [PARAM_ATMOS_GRID_CARTESC_INDEX] are changed. The red parts indicate the answer below:

   ```
   &PARAM_ATMOS_GRID_CARTESC_INDEX
     KMAX = 36,
     IMAXG = 120,                    (IMAXG = 90 in the original settings)
     JMAXG = 60,                     (JMAXG = 90 in the original settings)
   /
   ```

3. **How is horizontal grid interval changed while maintaining the computational domain?**
   When the number of MPI processes is not changed, (DX, DY) in [PARAM_ATMOS_GRID_CARTESC] and (IMAXG, JMAXG) in [PARAM_ATMOS_GRID_CARTESC_INDEX] should be changed.

   ```
   &PARAM_PRC_CARTESC
     PRC_NUM_X = 2,
     PRC_NUM_Y = 2,

   &PARAM_ATMOS_GRID_CARTESC_INDEX
     KMAX = 36,
     IMAXG = 360,                    (IMAXG = 90 in the original settings)
     JMAXG = 360,                    (JMAXG = 90 in the original settings)
   /

   &PARAM_ATMOS_GRID_CARTESC
     DX = 5000.D0,                   (DX = 20000.D0 in the original settings)
     DY = 5000.D0,                   (DY = 20000.D0 in the original settings)
   /
   ```

   In the case that the number of MPI processes is changed, if the below conditions are satisfied in the setting of [PARAM_ATMOS_GRID_CARTESC], your answer is correct:

   $$\text{IMAXG} \bmod \text{PRC\_NUM\_X} = 0$$
   $$\text{JMAXG} \bmod \text{PRC\_NUM\_Y} = 0$$

   Moreover, it is necessary to suitably change the time interval for the dynamics, i.e., (TIME_DT_ATMOS_DYN) and (TIME_DT) (refer to Section 4.2.5), and also the output intervals (refer to Section 4.2.6). The width of the buffer region should be set between 20 and 40 times of the grid spacing. Below is an example of the answer, indicating the case where the buffer region was set to 20 times the grid spacing.

```
&PARAM_PRC_CARTESC
  BUFFER_DX = 100000.D0,   (BUFFER_DX = 400000.D0, in the original settings)
  BUFFER_DY = 100000.D0,   (BUFFER_DY = 400000.D0, in the original settings)
/
```

4. **How is the location of the computational domain changed?**
   The coordinates of the center of the computational domain are changed as follows. Note that
   the unit is degrees. For example, 139 degrees 45.4 min = 139 + 45.4/60 degrees.

```
&PARAM_MAPPROJECTION
  MAPPROJECTION_basepoint_lon = 139.7567D0,   (135.220404D0, in the original setting)
  MPRPROJECTION_basepoint_lat = 35.6883D0,    (34.653396D0, in the original setting)
  MAPPROJECTION_type = 'LC',
  MAPPROJECTION_LC_lat1 = 30.00D0,
  MAPPROJECTION_LC_lat2 = 40.00D0,
/
```

5. **How is the integration time changed?**

```
&PARAM_TIME
  TIME_STARTDATE = 2007, 7, 14, 18, 0, 0,
  TIME_STARTMS = 0.D0,
  TIME_DURATION = 12.0D0,                    ( 6.0D0, in the original settings)
  TIME_DURATION_UNIT = "HOUR",
/
```

Furthermore, it is necessary to prepare a boundary condition of more than 12 hours by
`scale-rm_init`. Referring to Section 4.1.2, (NUMBER_OF_FILES) must be set greater than 3.

6. **How are output variables added and their output interval changed?**
   The output interval is set to (FILE_HISTORY_DEFAULT_TINTERVAL) in [PARAM_FILE_HISTORY]
   as below. The output variables are specified at (NAME) in [HISTORY_ITEM]. The list of the
   history variables is in the reference manual. See Section 5.3.2 for the reference manual.
   In order to horizontally/vertically remap a variable whose output interval is set at (TINTERVAL)
   in [HISTORY_ITEM] by SNO , the value of (TINTERVAL) must be equal to that of
   (FILE_HISTORY_DEFAULT_TINTERVAL).

```
&PARAM_FILE_HISTORY
  FILE_HISTORY_DEFAULT_BASENAME = "history_d01",
  FILE_HISTORY_DEFAULT_TINTERVAL = 1800.D0,        (3600.D0, in the original settings)
  FILE_HISTORY_DEFAULT_TUNIT = "SEC",
/

&HISTORY_ITEM NAME="SFLX_SW_up" /                  added
&HISTORY_ITEM NAME="SFLX_SW_dn" /                  added
 ....(omission)....
&HISTORY_ITEM NAME="DENS", TINTERVAL=1800.D0 /     when remapping this variable by SNO
 ....(omission)....
/
```

7. **How does the model restart?**
   The first integration for 3 hours is configured in `run.conf` as below. Once the 3 hours for
   integration have elapsed, the restart file is created.

```
&PARAM_TIME
 TIME_STARTDATE = 2007, 7, 14, 18, 0, 0,
 TIME_STARTMS = 0.D0,
 TIME_DURATION = 3.0D0,          necessary for more than 3 hours
 TIME_DURATION_UNIT = "HOUR",
 ....(omission)....
 TIME_DT_ATMOS_RESTART = 10800.D0,
 TIME_DT_ATMOS_RESTART_UNIT = "SEC",
 TIME_DT_OCEAN_RESTART = 10800.D0,
 TIME_DT_OCEAN_RESTART_UNIT = "SEC",
 TIME_DT_LAND_RESTART = 10800.D0,
 TIME_DT_LAND_RESTART_UNIT = "SEC",
 TIME_DT_URBAN_RESTART = 10800.D0,
 TIME_DT_URBAN_RESTART_UNIT = "SEC",
/

&PARAM_RESTART
 RESTART_OUTPUT = .true.,
 RESTART_IN_BASENAME = "../init/init_d01_20070714-180000.000",
 RESTART_OUT_BASENAME = "restart_d01",
/

&PARAM_ATMOS_BOUNDARY
 ATMOS_BOUNDARY_TYPE = "REAL",
 ATMOS_BOUNDARY_IN_BASENAME = "../init/output/boundary_d01",
/

&PARAM_ATMOS_REFSTAT
 ATMOS_REFSTATE_TYPE = "INIT",
 ATMOS_REFSTART_UPDATE_DT = 10800.D0,
/
```

When `(TIME_DURATION)` is set to 3 hours and `(RESTART_OUTPUT)` is `.true.`, the restart file is created at the end of the integration. Therefore, `(TIME_DT_ATMOS_RESTART)`, `(TIME_DT_OCEAN_RESTART)`, `(TIME_DT_LAND_RESTART)`, and `(TIME_DT_URBAN_RESTART)` are not necessary.

When `(TIME_DURATION)` is set to more than 3 hours, it is necessary to specify `(TIME_DT_ATMOS_RESTART)`, `(TIME_DT_OCEAN_RESTART)`, `(TIME_DT_LAND_RESTART)`, and `(TIME_DT_URBAN_RESTART)`. The values for `(TIME_DT_ATMOS_RESTART)`, `(TIME_DT_OCEAN_RESTART)`, `(TIME_DT_LAND_RESTART)`, and `(TIME_DT_URBAN_RESTART)` must be a divisor of 3 hours (10800 s) and a multiple of `(TIME_DT)`. In order to make the reference state (See Section 4.3.1.7) in the restart calculation identical to that in the first calculation, the update interval of the reference state is specified to be a divisor of 3 hours.

The configuration for the restart calculation from 3 to 6 hours in integration time is as follows:

```
&PARAM_TIME
 TIME_STARTDATE = 2007, 7, 14, 21, 0, 0,
 TIME_STARTMS = 0.D0,
 TIME_DURATION = 3.0D0,                                          set for longer than 3 hours
 TIME_DURATION_UNIT = "HOUR",
/

&PARAM_RESTART
 RESTART_OUTPUT = .true.,                                        (not always necessary)
 RESTART_IN_BASENAME = "restart_d01_20070714-210000.000",        (necessary)
 RESTART_OUT_BASENAME = "restart2_d01",                          (not always necessary)
/

&PARAM_ATMOS_BOUNDARY
 ATMOS_BOUNDARY_TYPE = "REAL",
 ATMOS_BOUNDARY_IN_BASENAME = "../init/output/boundary_d01",
/

&PARAM_ATMOS_REFSTAT
 ATMOS_REFSTATE_TYPE = "INIT",
 ATMOS_REFSTART_UPDATE_DT = 10800.D0,
/
```

SCALE USERS GUIDE

Author & Editor

Team SCALE User's guide working group

> If you have any questions or concerns,
> please contact developer team (`scale@ml.riken.jp`).