

SCALE USERS GUIDE

Version 5.1.5

Team SCALE
UGC working group

July 20, 2017



Contents

1	Overview	3
1.1	Introduction	3
1.1.1	What is SCALE ?	3
1.1.2	Structure of SCALE-RM	4
1.2	Notations used in this document	6
2	Install	8
2.1	Necessary system environment	8
2.2	Installation of external libraries and drawing tools	9
2.3	Compilation of SCALE	10
2.4	Compilation of post-processing tool (net2g)	12
3	Operation check and basic usage	13
3.1	Introduction	13
3.2	How to execute model	13
3.3	Post-processing and drawing	15
3.4	Guideline for further study	16
4	Conduction of real atomsphere experiment	19
4.1	Overview	19
4.2	Preparations for input data (boundary data)	20
4.3	Preparation for experiment set	22
4.4	Creating topographical data: pp	23
4.5	Creating the initial and boundary data: init	24
4.6	Execution of simulation: run	27
4.7	Quick look at simulation result: net2g	30
5	Various settings	33
5.1	Setting the target domain	33
5.1.1	Setting the number of MPI processes	34
5.1.2	Setting the number of horizontal and vertical grids	34
5.1.3	Setting grid intervals along the horizontal and vertical directions	35
5.1.4	Setting the buffer region and nudging method	36
5.2	Setting the map projection	38
5.3	Setting the topography	39
5.4	How to create initial and boundary data	41
5.5	Setting integration period and time step	45
5.6	Setting the history file and output variable	46
5.7	Setting the dynamical core	47
5.8	Setting the physical process	48
5.8.1	Cloud micro-physics	48
5.8.2	Turbulence scheme	50
5.8.3	Radiation scheme	50
5.8.4	Surface flux at the bottom of atmospheric boundary	52

5.8.5	Ocean model	54
5.8.6	Land model	55
5.8.7	Urban model	56
5.9	Post-processing	57
5.10	How to restart run	60
5.11	How to conduct nesting experiment	62
5.11.1	Treatment of topography in child domain	63
5.11.2	Offline nesting experiment	65
5.11.3	Online nesting experiment	69
5.12	How to run bulk job	72
5.13	Supporting tool for preparation of configuration files	75
A	Namelist of configuration file and variable name list for history output	79
B	Frequently asked questions	80

Chapter 1

Overview

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 5.1.5. The current version of SCALE contains a regional model SCALE-RM and a global model SCALE-GM . Only a dynamical core is provided for the latter. 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: Chapter 1 provides an overview of SCALE , Chapter 2 describes how to install SCALE-RM along with the system requirements. Using simple examples, Chapters 3 and 4 use examples and explain the basic use of SCALE-RM in an ideal experiment and the method to conduct an 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. Chapter 5 describes how to change the model configuration, and explains various functions and tools.

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-5.1.5/LICENSE` under the main directory. This explanation of its use is also provided on the SCALE webpage (<http://scale.aics.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

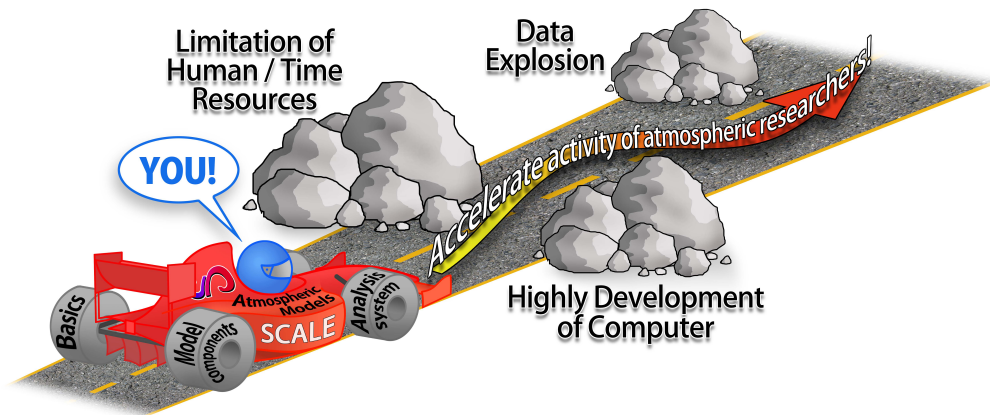


Figure 1.1: Aims of SCALE

SCALE was developed in RIKEN/AICS with several outside contributors, and its improvement and extension continue. Figure 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. This has led to high computational performance of SCALE not only in supercomputers, such as the K Computer and the Fujitsu FX10, but also for general-purpose commercial computers, such as Intel processor-based machines.

SCALE-RM is a numerical model that fully uses SCALE. This model is contained in the SCALE package, as shown in Fig. 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.

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).

^{*1)}Refer to [Team SCALE \(2015\)](#), [Sato et al. \(2015\)](#), and [Nishizawa et al. \(2015\)](#) for the details of the model structure and the discretization method.

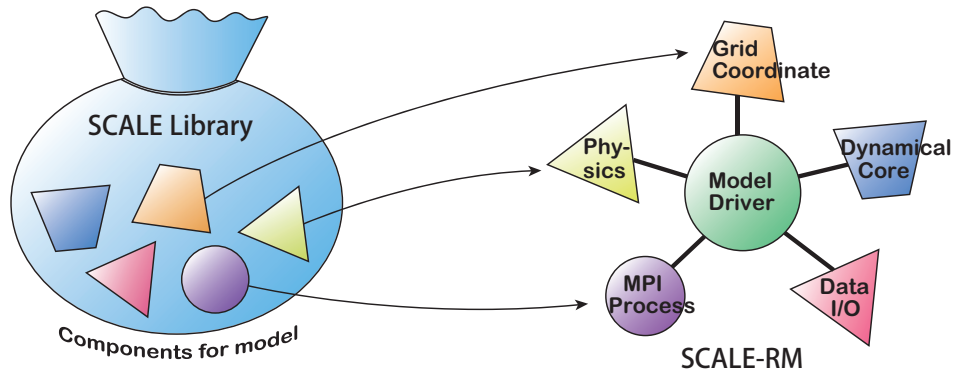


Figure 1.2: Relationship between the library SCALE and the model SCALE-RM

- 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
 - Supporting inputs from the WRF-ARW^{*3)} and GrADS^{*4)} formats.

Dynamical core

- Governing equations: 3D fully compressible non-hydrostatic equations
- Spatial discretization: finite volume method
 - 2nd-, 4th-, and 6th-order central differences
 - 3rd- and 5th-order upwind differences
- Time integration: selection from the “fully explicit method” or the “horizontally explicit and vertically implicit methods”
 - Heun-type 3rd-order Runge–Kutta scheme
 - Wicker and Skamarock (2002)’s 3-step Runge–Kutta scheme
 - 4th-order Runge–Kutta scheme
- Guarantee of non-negative value:
 - Flux corrected transport method (Zalesak, 1979)

^{*2)}<http://cfconventions.org/>

^{*3)}<http://www.wrf-model.org/>

^{*4)}<http://cola.gmu.edu/grads/>

- [Koren \(1993\)](#)'s filter: available only with the use of the 3rd-order upwind scheme
- Numerical filter: 4th-order hyper-viscosity and diffusion
- Topography: expressed using terrain-following coordinates

Physical processes

- Turbulence process: selectable from among the following
 - [Smagorinsky \(1963\)](#) & [Lilly \(1962\)](#)-type sub-grid model corrected by [Brown et al. \(1994\)](#) and [Scotti et al. \(1993\)](#)
 - [Deardorff \(1980\)](#) sub-grid model
 - MYNN level 2.5 boundary scheme ([Mellor and Yamada \(1982\)](#); [Nakanishi and Niino \(2004\)](#))
- 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](#))
 - * Louis-type bulk method ([Uno et al., 1995](#))

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 polygon with rounded corners means that the description is in an editable file.

```
---- description in a file  
----- description in a file  
----- description in a file
```

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

Chapter 2

Install

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

2.1 Necessary system environment

Recommended system environment

- **Hardware system**

Although the necessary hardware depends on the experiment to be performed, the following are the specifications for conducting a tutorial as in Chapters 3 and 4.

- **CPU** : The system need to have more than two and more than four physical cores for an ideal experiment and a real atmospheric experiment, respectively.
- **Memory** : The system needs 512 MB and 2 GB memory for the ideal experiment and a 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 a real atmospheric experiment.

- **System software**

- **OS** : Linux OS, Mac OS X
Refer to Table 2.1 for other OSs currently supported.
- **Compiler** : C, Fortran
The FORTRAN compiler should support FORTRAN 2003 syntax. Refer to Table 2.2 for compilers already confirmed as supported.

Table 2.1: OSs already checked (the ones below are based on x86-64.)

Name of OS	Version	Note
CentOS	6.5、 6.6、 6.8、 7.0、 7.1、 7.2	
openSUSE	13.2	
SUSE Enterprise Linux	11.1、 11.3	GNU compiler is not available in version 11.1.
fedora	20	
Vine Linux	6.2、 6.3	
Mac OS X	10.10(Yosemite)	

Table 2.2: Compilers and MPI libraries already checked

Name of compiler	Version	Note
GNU (gcc/gfortran)	4.4.x, 4.9.x	The series with version 4.4.x sometimes issues a warning
Intel (icc/fort)	13.0.1, 14.0.2, 15.0.0	Version of 2013 or later are recommended
Name of MPI library		
openMPI	1.8.5, 1.10.3	
Intel MPI	4.0 Update 3, 4.1.0, 5.0	The version released in 2013 or later is recommended.
SGI MPT	2.05, 2.09	A combined use with the Intel compiler is available.

- **MPI library** : The MPI library should support the MPI 1.0/2.0 protocol. Refer to Table 2.2 for MPI libraries already confirmed as supported.
- **File I/O library** : The development libraries of gzip, HDF5, and netCDF 4 are required.
In stead of the HDF5/netCDF 4, netCDF 3 is sufficient, but the size of the output file increases in this case.

Useful tools (not always necessary)

- **Data conversion tool** : wgrib, wgrib2, NCL
In the tutorial for the real atmospheric experiment, wgrib is used.
- **Viewer tool** : GrADS ^{*1)}, GPhys/Ruby-DCL ^{*2)}, ncview ^{*3)}, and so on.
- **Evaluation tool of computational performance** : The PAPI library ^{*4)} is available.

Since the source code of SCALE is written in FORTRAN 2003 standard syntax, the compiler must support it. For example, GNU gfortran version 4.3 or previous versions cannot be used for SCALE compilation because they do not follow the FORTRAN 2003 standard.

In addition to the above environment, it has been confirmed that SCALE can be compiled and run on machines with a SPARC-based instruction set, such as the K Computer and the Fujitsu PRIME-HPC FX10.

2.2 Installation of external libraries and drawing tools

Required libraries

The required external libraries are described below; each library is commonly used.

- HDF5 Library (<https://www.hdfgroup.org/HDF5/>)
- netCDF Library (<http://www.unidata.ucar.edu/software/netcdf/>)
- MPI Library (openMPI version, <http://www.open-mpi.org/>)
- LAPACK (<http://www.netlib.org/lapack/>) (only required for SCALE-GM)

*1) <http://cola.gmu.edu/grads/>

*2) <https://www.gfd-dennou.org/arch/ruby/products/gphys/>

*3) http://meteora.ucsd.edu/~piece/ncview_home_page.html

*4) <http://icl.utk.edu/papi/>

The interested reader can refer to the above URLs for the installation information of each library. Since most series of Linux distributions provide them as packages, their installation is easy. These libraries have been already installed on the K Computer as AICS software. Refer to the description “AICS software” in the K portal site or the URL

<http://www-sys-aics.riken.jp/releasedsoftware/ksoftware/pnetcdf.html>

for details. The library paths are described there in “Makedef.K.” The tutorials in Chapters 3 and 4 assume that the above library environment has been prepared.

Drawing tools

The following are typical drawing tools that can draw the initial conditions, boundary data, and the simulation results on SCALE . The GPhys and GrADS are used for a quick view and the drawing model output in the tutorial in Chapters 3 and 4, respectively. Other tools are of course available, if they can be read in netCDF format, which is the output format of SCALE .

- 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
<https://www.gfd-dennou.org/library/ruby/tutorial/install/index-j.html>
- 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 `netcdf2grads_h` is needed to combine the output of SCALE in one file that is readable from GrADS . Refer to `netcdf2grads_h` for installation instructions in Section , and to Chapters 3 and 4 and Section for details pertaining to its use.
 - How to install : Refer to <http://cola.gmu.edu/grads/downloads>.
- Ncview: a 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

2.3 Compilation of SCALE

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 (refer to Appendix ??)

Obtaining the source code

The source code for the latest release can be downloaded from <http://scale.aics.riken.jp/ja/download/index.html>.

The directory `scale-5.1.5/` can be seen when the tarball file of the source code is extracted.

```
$ tar -zxvf scale-5.1.5.tar.gz
$ ls ./
  scale-5.1.5/
```

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-5.1.5/sysdep/`. Choose a Makedef file according to your environment. Table 2.3 shows environments that support Makedef files that have already been checked. If there is no file for your environment, create a Makedef file by modifying any existing one.

Table 2.3: Examples of environments and their corresponding Makedef files.

OS/Computer	Compiler	MPI	Makedef file
Linux OS x86-64	gcc/gfortran	openMPI	Makedef.Linux64-gnu-ompi
	icc/fort	intelMPI	Makedef.Linux64-intel-impi
	icc/fort	SGI-MPT	Makedef.Linux64-intel-mpt
Mac OS X	gcc/gfortran	openMPI	Makedef.MacOSX-gnu-ompi
K Computer	fccpx/frtpx	mpiccpx/mpifrtpx	Makedef.K
Fujitsu PRIME-HPC FX10	fccpx/frtpx	mpiccpx/mpifrtpx	Makedef.FX10

When Linux OS is used, a GNU compiler and openMPI in "Makedef.Linux64-gnu-ompi" are available. If installation is conducted in another environment, change the file name according to Table 2.3. 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`.

The setting of following path is necessary. You must set the environmental variables for HDF5 and netCDF 4 as follows:

```
$ export HDF5="/opt/hdf5"
$ export NETCDF4="/opt/netcdf"
$ export NETCDF_INCLUDE="-I/opt/netcdf/include"
$ export NETCDF_LIBS="-L/opt/hdf5/lib64 -L/opt/netcdf/lib64 \
  -lnetcdf -lhdf5_hl -lhdf5 -lm -lz"
```

The above example shows that HDF5 is installed in the directory `/opt/hdf5` and netCDF 4 in the directory `/opt/netcdf` on an Intel compiler.

Compile

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

```
$ cd scale-{\version}/scale-rm/src
$ make -j 4
```

The option "-j 4" shows the parallel number; it is four-parallel in this example. Specify this number according to your environment. When a compilation is successful, the following three executable files are generated under the `scale-5.1.5/bin` directory.

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

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

```
$ make clean
```

Note that the library already compiled is not deleted in this command. If you want to remove all files, use the following command:

```
$ make allclean
```

Moreover, if the files are recompiled by changing the compilation environment and options, execute “make allclean.”

Points to note

- In SCALE , a compilation and an archive are conducted in directory `scale-5.1.5/scalelib/` just under the TOP directory `scale`. 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 DEBUG=T"`.
- If you want to change the compilation options in detail, edit `Makedef.***`.

2.4 Compilation of post-processing tool (net2g)

The output file of SCALE-RM is divided and stored in every computational node. SCALE provides a post-processing tool “net2g” to combine these output files (`history.*****.nc`) and convert them into a data format direct readable in GrADS . Since it is used in the tutorial Chapters 3 and 4, the method for the compilation of “net2g” is explained here.

Specify the environmental variable for the Makedef file according to your environment, such as in the compilation of the main body of SCALE . Then, move to directory “net2g” and execute a command. The parallel executable file using the MPI library is generated by the following command:

```
$ cd scale-{\version}/scale-rm/util/netcdf2grads_h  
$ make -j 2
```

If there is no MPI library, give a compile command to generate the serial executable binary.

```
$ make -j 2 NOMPI=T
```

If executable file “net2g” is generated, the compilation is successful. When the executable binary is deleted, execute the following command:

```
$ make clean
```

Chapter 3

Operation check and basic usage

3.1 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 recommend that the user perform this tutorial because it includes a check for whether the compilation of SCALE in Chapter 2 has been completed. This chapter assumes that the following file has been already generated:

```
scale-5.1.5/bin/scale-rm
scale-5.1.5/bin/scale-rm_init
scale-5.1.5/scale-rm/util/netcdf2grads_h/net2g
```

Furthermore, GrADS is used as a drawing tool. “gpview” can also be used for the confirmation of the result. Refer to Section 2.2 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.

3.2 How to execute model

Experimental setup

For this tutorial for an ideal experiment, a case of cumulus cloud convection is used. In this experiment, a cumulus appears and develops in a quasi-2D model by providing a typical atmospheric vertical profile and an initial disturbance in the lower troposphere. Table 3.1 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-5.1.5/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. Reading the

Table 3.1: Experimental settings of ideal experiment

	Configuration	Note
Number of MPI processes	2: eastward direction, 1: northward direction	2 MPI parallel
Horizontal grid spacing	500m: eastward direction, 1000m :northward direction	The quasi 2 dimensional experiment in the eastward and vertical directions.
Number of horizontal grids	40: eastward direction, 2: northward direction ^a	
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 s	10 s in microphysics scheme
Integration time	3,600 s	The time step is 720.
Time interval of data output	300 s	
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.

^aIn the current version, the pure 2D experimental framework is not prepared. In this case, the initial conditions with the same values are set northward. Thus, the setting corresponds to a 2D experiment. Further, the number of grids northward should be same as the number of halos. Refer to Section ?? for the required numbers of halos.

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]. In the case that the configuration file `init_R20kmDX500m.conf` is used to conduct `scale-rm_init` using two-MPI parallel, give the command to execute `scale-rm_init` as follows:

```
$ 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
*** a single communicator
*** a single communicator
*** 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:

```
+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized
```

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 100 KB. The file whose name ends with “.nc” is formatted by netCDF . It can be directly read by GPhys/Ruby-DCL and nview.

Execution of simulation

The number of parallel process is required to be the same as that when creating the initial conditions. The configuration file for the run is `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:

```
+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized
```

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 each file is approximately 1.4 MB.

3.3 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 simple binary form (GrADS form) that can be directly accessed. The binary form makes it easy for users to analyze the result. Link to the post-processing tool `net2g` compiled in Section 2.4:

```
$ ln -s ../../../../util/netcdf2grads_h/net2g ./
```

The method of execution of `net2g` is the same as that of SCALE-RM , i.e.,

```
$ mpirun -n [the number of the processes] ./net2g [the configuration file]
```

The configuration file `net2g.conf` is intended for special uses of `net2g`. Give this configuration file to `net2g` and execute it as follows:


```
$ mpirun -n 2 ./net2g net2g.conf
```

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

```
+++ MPI COMM: Corrective Finalize
```

The execution of net2g should be handled, so that the number of MPI processes is identical to, or a divisor of, that used for the run of SCALE-RM . The following six files are generated under the same directory by this execution:

```
QHYD_d01z-3d.ct1
QHYD_d01z-3d.grd
U_d01z-3d.ct1
U_d01z-3d.grd
W_d01z-3d.ct1
W_d01z-3d.grd
```

The “grd” files are the converted files in the simple binary form of direct access (GrADS form) obtained by merging the divided files, whereas the “ct1” files are used to render them readable by GrADS .

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 can be found in the simulation, and post-processing is successfully concluded.

To convert the result of the output into binary data for other variables, add them to (VNAME) in [VARI] in the configuration file `net2g.conf`:

```
&VARI
VNAME = "U", "W", "QHYD"
/
```

To check the output variable in the history file, use `ncdump` of `netCDF` . Refer to Section 5.9 for the detailed use of `net2g`.

3.4 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 Chapter 5.

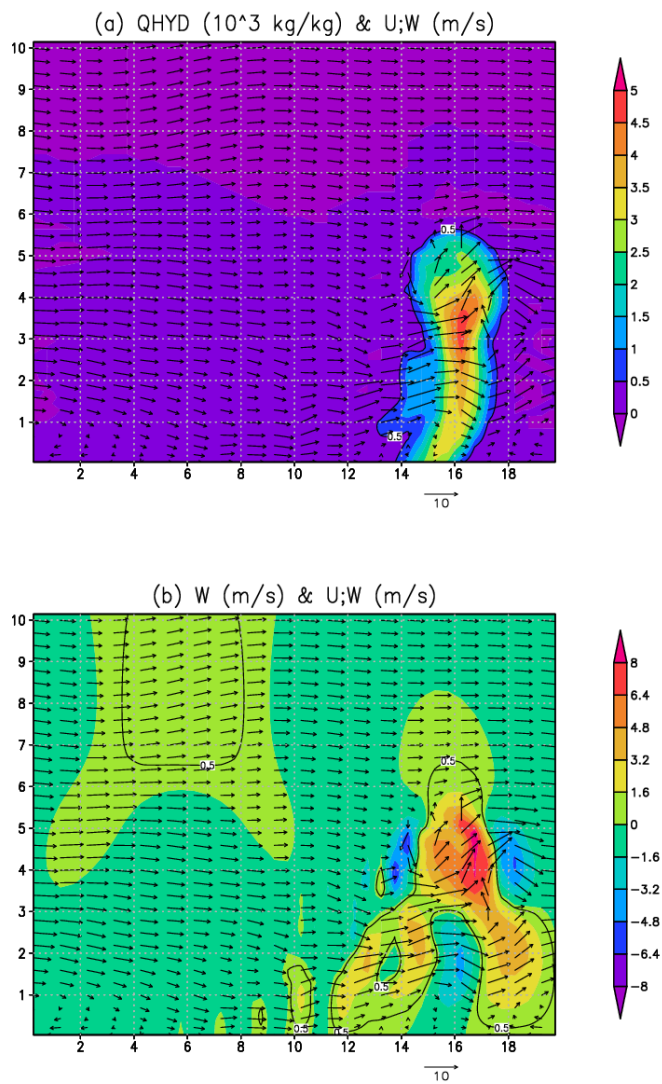


Figure 3.1: The horizontal-vertical cross-section at $Y=750\text{m}$ after $t=1200$ s (20 minutes later); The color indicates (a) the mass concentration of the hydrometeor and (b) vertical velocity. The vector indicates flow in both of figures.

Chapter 4

Conduction of real atmosphere experiment

4.1 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. 4.1.

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. `net2g`: Converting netCDF output data to GrADS format (optional)

Hereinafter, the absolute path `scale-5.1.5/scale-rm/test/tutorial/` is denoted by `#{Tutorial_DIR}`.

The settings for the calculation domain used in this tutorial are given in Table 4.1. Figure 4.2 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; e.g., there is no cumulus parameterization in this simulation, even though the horizontal resolution is 20 km.

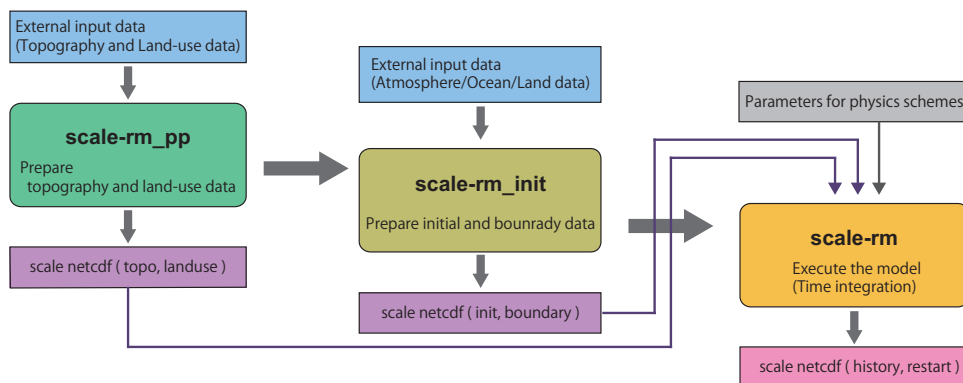


Figure 4.1: SCALE-RM procedure of model execution

Table 4.1: Overview of experimental settings

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

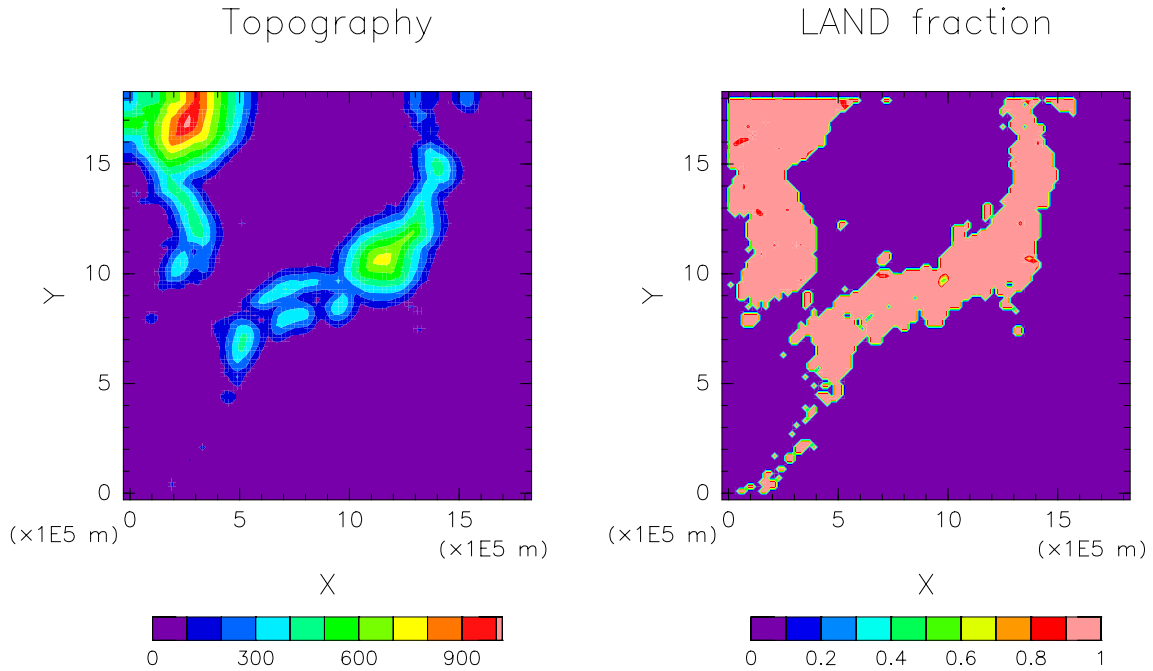


Figure 4.2: Topographical and land-ocean distribution in the domain

4.2 Preparations for input data (boundary data)

When a realistic atmospheric experiment is conducted, boundary data need to be provided to SCALE-RM . Table 4.2 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.

1. Downloading data

Obtain the data for altitude and land-use classification formatted for SCALE-RM from http://scale.aics.riken.jp/download/scale_database.tar.gz, and extract them to any directory:

Table 4.2: 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, where the above `tar` file is extracted, 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 database directory.

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 one-day 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/`. The following is the data list, formatted by `grib1`:

```
fnl_20070714_18_00.grib1
fnl_20070715_00_00.grib1
fnl_20070715_06_00.grib1
fnl_20070715_12_00.grib1
```

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

2. Conversion of data from grib to binary format

Execute `convert_grib2grads_FNLgrib1.sh` in the directory `${Tutorial_DIR}/real/tools/`:

```
$ cd ${Tutorial_DIR}/real/tools/
$ sh convert_grib2grads_FNLgrib1.sh
```

The following files are found if it is successful:

```
$ ls FNL_output/**
FNL_output/200707/FNLatm_2007071418.grd
FNL_output/200707/FNLatm_2007071500.grd
FNL_output/200707/FNLatm_2007071506.grd
FNL_output/200707/FNLatm_2007071512.grd
FNL_output/200707/FNLland_2007071418.grd
FNL_output/200707/FNLland_2007071500.grd
FNL_output/200707/FNLland_2007071506.grd
FNL_output/200707/FNLland_2007071512.grd
FNL_output/200707/FNLsfc_2007071418.grd
FNL_output/200707/FNLsfc_2007071500.grd
FNL_output/200707/FNLsfc_2007071506.grd
FNL_output/200707/FNLsfc_2007071512.grd
```

4.3 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 USERS.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/
  net2g/
  pp/
  run/
```

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 5.13 for a detailed explanation of "the making tool for the complete settings of the experiment."

4.4 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 4.1.

In Namelists in pp.d01.conf, the parameters related to the domain are configured in [PARAM_PRC], [PARAM_INDEX], and [PARAM_GRID]. The domain is decomposed along each of the X and Y directions into two domains. Thus, four MPI processes are used. The number of grids per MPI process is (IMAX = 45) and (JMAX = 45). Thus, the total number of grids is 90 (= 2 × 45) along the X and Y directions. The grid spacings in each direction (DX, DY) in [PARAM_GRID] 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
  PRC_NUM_X = 2,
  PRC_NUM_Y = 2,
  PRC_PERIODIC_X = .false.,
  PRC_PERIODIC_Y = .false.,
/

&PARAM_INDEX
  KMAX = 36,
  IMAX = 45,
  JMAX = 45,
/

&PARAM_GRID
  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_CNVTPOPO_GTOPO30`] and (`GLCCv2_IN_DIR`) in [`PARAM_CNVLANDUSE_GLCCv2`] specify the locations of altitude data and land-use classification data, respectively.

```
&PARAM_CNVTPOPO_GTOPO30
GTOPO30_IN_CATALOGUE = "GTOPO30_catalogue.txt",
GTOPO30_IN_DIR = "./topo/GTOPO30/Products",
/

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

After preparation of the configure 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 4.1. When the job is finished normally, the following message is output at the end of the log file: `pp_LOG_d01.pe000000`.

```
+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized
```

Furthermore, the files `topo_d01.pe#####.nc` (file size of approximately 180 KB) and `landuse_d01.pe#####.nc` (file size of approximately 220 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
$ gpview landuse_d01.pe00000*@FRAC_LAND --aspect=1 --nocont
```

The same figure as Fig. 4.2 is generated if the results are correct.

4.5 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 4.1. 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_TOPO
  TOPO_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 5.2
  BASENAME_ORG = "namelist.grads_boundary.FNL.grib1",
  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 5.4 for the details of the input file.

Link the input data (FNL) that are converted into binary form in Section 4.2 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.

```

FNLatm_00000.grd -> ../tools/FNL_output/200707/FNLatm_2007071418.grd
FNLatm_00001.grd -> ../tools/FNL_output/200707/FNLatm_2007071500.grd
FNLatm_00002.grd -> ../tools/FNL_output/200707/FNLatm_2007071506.grd
FNLatm_00003.grd -> ../tools/FNL_output/200707/FNLatm_2007071512.grd
FNLLand_00000.grd -> ../tools/FNL_output/200707/FNLLand_2007071418.grd
FNLLand_00001.grd -> ../tools/FNL_output/200707/FNLLand_2007071500.grd
FNLLand_00002.grd -> ../tools/FNL_output/200707/FNLLand_2007071506.grd
FNLLand_00003.grd -> ../tools/FNL_output/200707/FNLLand_2007071512.grd
FNLSfc_00000.grd -> ../tools/FNL_output/200707/FNLSfc_2007071418.grd
FNLSfc_00001.grd -> ../tools/FNL_output/200707/FNLSfc_2007071500.grd
FNLSfc_00002.grd -> ../tools/FNL_output/200707/FNLSfc_2007071506.grd
FNLSfc_00003.grd -> ../tools/FNL_output/200707/FNLSfc_2007071512.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.grib1 ./
```

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`:

```

+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized

```

The file sizes of the boundary and initial data, `boundary_d01.pe#####.nc` and `init_d01_20070714-180000.000.pe#####.nc`, are approximately 5.8 MB and 3.5 MB, respectively, where `#####` represents the MPI process number.

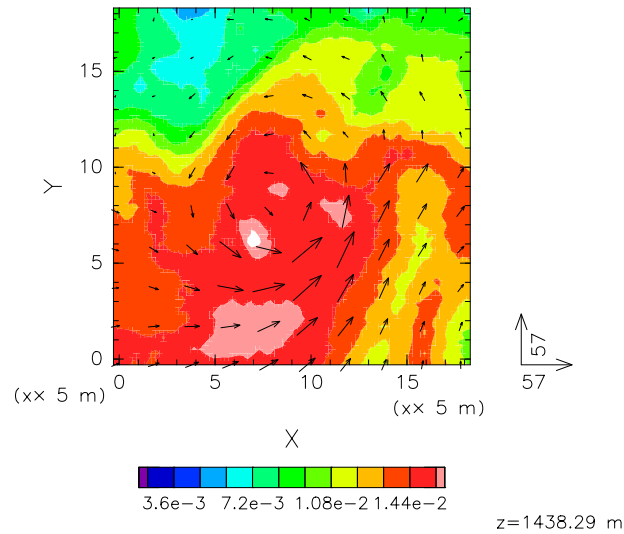
OPTION

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

```
$ gpvectl --scalar --slice z=1500 --nocont --aspect=1 --range=0.002:0.016 \
--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
```

If the same figure as Fig. 4.3 is found, it is successfully concluded.

Vapor mass to total mass (Specific humidity)(momentum x,



/home1/gleo/share/SCALE/bin/gpvect 2016-09-01

Figure 4.3: Initial field at $z=1500\text{m}$ for the tutorial experiment. The color indicates relative humidity and the vector horizontal momentum flux.

4.6 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 4.1. 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_TOPO
  TOPO_IN_BASENAME = "../pp/topo_d01",
/

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

&PARAM_RESTART
  RESTART_RUN = .false.,
  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_START_DATE = 2007, 7, 14, 18, 0, 0,
  ATMOS_BOUNDARY_UPDATE_DT = 21600.0,
  ATMOS_BOUNDARY_USE_DENS = .true.,
  ATMOS_BOUNDARY_USE_VELZ = .true.,
  ATMOS_BOUNDARY_USE_QHYD = .false.,
  ATMOS_BOUNDARY_VALUE_VELZ = 0.0,
  ATMOS_BOUNDARY_ALPHAFACDENS = 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_HISTORY)`.

```

&PARAM_HISTORY
HISTORY_DEFAULT_BASENAME = "history_d01", : output file name
HISTORY_DEFAULT_TINTERVAL = 3600.D0, : time interval for output
HISTORY_DEFAULT_TUNIT = "SEC", : unit of output time interval
HISTORY_DEFAULT_TAVERAGE = .false.,
HISTORY_DEFAULT_DATATYPE = "REAL4",
HISTORY_DEFAULT_ZCOORD = "model", : no vertical interpolation
HISTORY_OUTPUT_STEPO = .true., : output at initial time(t=0) or not
/

```

According to the above setting, variables listed in the following (**HISTITEM**) are output. It is possible to change output intervals at every variable in (**HISTITEM**) by adding options if needed. The mean value instead of the snapshot value can be output. Refer to Section 5.6 for the details.

```

&HISTITEM item="MSLP" / : sea-level pressure
&HISTITEM item="PREC" / : precipitation intensity (2D)
&HISTITEM item="OLR" / : outgoing longwave radiation (2D)
&HISTITEM item="U10" / : horizontal wind speed along X direction at 10 m(2D)
&HISTITEM item="V10" / : horizontal wind speed along Y direction at 10 m(2D)
&HISTITEM item="T2" / : temperature at 2m (2D)
&HISTITEM item="Q2" / : specific humidity at 2m (2D)
&HISTITEM item="SFC_PRES" / : surface pressure (2D)
&HISTITEM item="SFC_TEMP" / : bulk surface temperature(2D)
&HISTITEM item="DENS" / : density (3D)
&HISTITEM item="QV" / : specific humidity (3D)
&HISTITEM item="QHYD" / : mass concentration of total hydrometeor (3D)
&HISTITEM item="PRES" / : pressure (3D)
&HISTITEM item="U" / : horizontal wind speed along X direction (3D)
&HISTITEM item="V" / : horizontal wind speed along Y direction (3D)
&HISTITEM item="T" / : temperature (3D)
&HISTITEM item="W" / : vertical wind speed (3D)
&HISTITEM item="Uabs" / : absolute value of wind velocity(3D)
&HISTITEM item="PT" / : potential temperature (3D)
&HISTITEM item="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 5.7 and 5.8 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 \scalerm
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:

```
+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized
```

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 23 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 (`HISTITEM`) are output. The files are formatted by NetCDF, corresponding to climate and forecast (CF) metadata convention.

4.7 Quick look at simulation result: net2g

In this section, we explain how to use `netcdf2grads`. The program `netcdf2grads` (for short, `net2g`) merges the netCDF files (`history.**.nc`)*²⁾, which are divided process by process, into a binary file in GrADS format. The simulation results are also validated using the converted GrADS binary data.

Conversion to GrADS binary

Use `net2g` to convert GrADS binary from the history file of netCDF having divided every process. Only a minimum of the procedures is explained here. Refer to Section 5.9 for their detailed use.

First, move to the directory `net2g`:

```
$ cd ${Tutorial_DIR}/real/experiment/net2g
$ ls
  net2g -> ../../../../util/netcdf2grads_h/net2g
  net2g.2D.d01.conf
  net2g.3D.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 2.4. As an example, a procedure to convert the 2D variables MSLP and PREC to GrADS format is explained here. We also explain how to extract the 3D variables U and V at 850 hPa, 500 hPa, and 200 hPa, and convert them into GrADS format. The configuration files for 2D and 3D variables are prepared as files `net2g.3D.d01.conf` and `net2g.2D.d01.conf`, respectively.

When `net2g` is executed, the number of processes needs to be a divisor of the number of the processes used for the simulation. The number of processes is four here. Since `net2g` cannot simultaneously convert 2D and 3D variables, they must be converted separately as follows:

```
$ mpirun -n 4 ./net2g net2g.2D.d01.conf
$ mpirun -n 4 ./net2g net2g.3D.d01.conf
```

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

```
+++ MPI COMM: Corrective Finalize
```

The following files are also found. `**ctl` represents the “ctl” file for the XY grid system of SCALE-RM, `**lccr.ctl` a ctl file for the drawing results on the latitude-longitude coordinates:

*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.

```
MSLP_d01z-2d.ct1  
MSLP_d01z-2d.grd  
MSLP_d01z-2d_lccr.ct1  
PREC_d01z-2d.ct1  
PREC_d01z-2d.grd  
PREC_d01z-2d_lccr.ct1  
PRES_d01z-3d.ct1  
PRES_d01z-3d.grd  
PRES_d01z-3d_lccr.ct1  
U_d01z-3d.ct1  
U_d01z-3d.grd  
U_d01z-3d_lccr.ct1  
V_d01z-3d.ct1  
V_d01z-3d.grd  
V_d01z-3d_lccr.ct1
```

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 are successful, the same figures as Figs. [4.4](#), [4.5](#), and [4.6](#) are obtained.

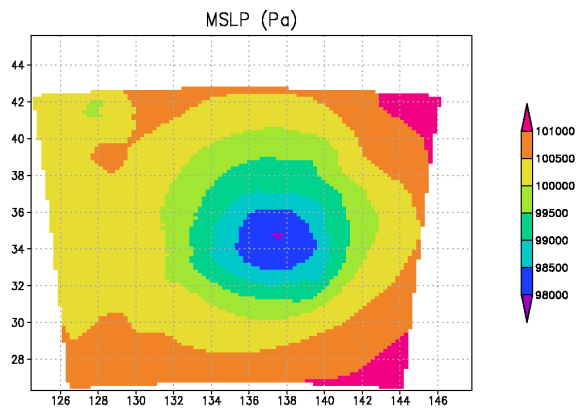


Figure 4.4: Sea-level pressure after 6 hours

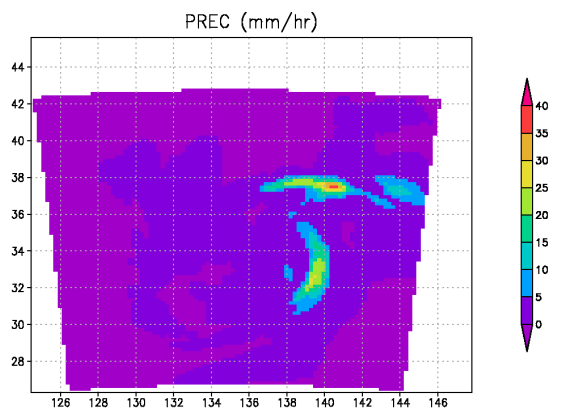


Figure 4.5: Precipitation flux after 6 hours

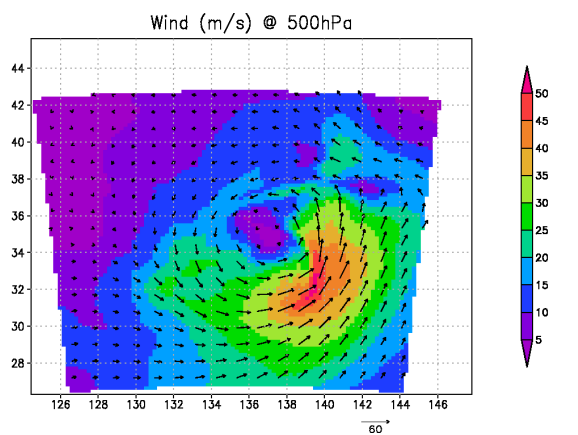


Figure 4.6: Wind velocity after 6 hours

Chapter 5

Various settings

This chapter describes various settings in more detail than in the previous tutorials. Since each section is closed itself basically, this chapter can be utilized as a dictionary.

5.1 Setting the target domain

This section explains the number of grids, and the target domain and its relationship with the MPI process. The calculation domain is determined by the horizontal grid spacing, the number of grid points, and the number of MPI processes. Figure 5.1 gives an example of this relationship. The parallelization is implemented by a 2D domain decomposition in the horizontal direction.

The numbers of grids and of MPI processes are specified in $(IMAX, JMAX)$ in $[PARAM_INDEX]$, and in (PRC_NUM_X, PRC_NUM_Y) in $[PARAM_PRC]$, respectively. As shown in Fig. 5.1, the entire domain is divided into (PRC_NUM_X) along the X direction and (PRC_NUM_Y) in the Y direction. Each sub-domain is managed by an MPI process, each of which takes charge of a grid block of $IMAX \times JMAX \times KMAX$. Care is taken to ensure that this number of grids is taken charge of by one process: its value is not identical to the total number of grids in the domain. In other words, the entire domain depends on the number of MPI processes, the horizontal grid spacing, and the number of grids.

Thus, the number of grids along each horizontal direction and that in the entire domain are derived as follows:

$$\text{Number of grids in the X direction} = IMAX \times PRC_NUM_X \quad (5.1)$$

$$\text{Number of grids in the Y direction} = JMAX \times PRC_NUM_Y \quad (5.2)$$

$$\begin{aligned} \text{Total number of grids in the domain} \\ = (IMAX \times PRC_NUM_X) \times (JMAX \times PRC_NUM_Y) \times (KMAX), \end{aligned} \quad (5.3)$$

where $(KMAX)$ is the number of grids along the vertical direction, which is specified in $[PARAM_INDEX]$.

By using Eqs. (5.1 and 5.2), the size of the entire domain is determined as follows:

$$\text{Domain length in the X direction} = \text{number of grids in the X direction} \times DX \quad (5.4)$$

$$\text{Domain length in the Y direction} = \text{number of grids in the Y direction} \times DY \quad (5.5)$$

where (DX, DY) is grid spacings specified in $[PRAM_GRID]$ as described in subsection 5.1.3. If the horizontal resolution and domain size are set and the available number of MPI processes is given, the number of grids that are controlled by one MPI process can be determined as in the above relationship.

In the next subsections, the configuration of the MPI processes, the number of grids, and the grid interval are described in more detail. [Note that it is necessary that these settings be identical among the configuration files `pp.conf`, `init.conf`, and `run.conf`.](#)

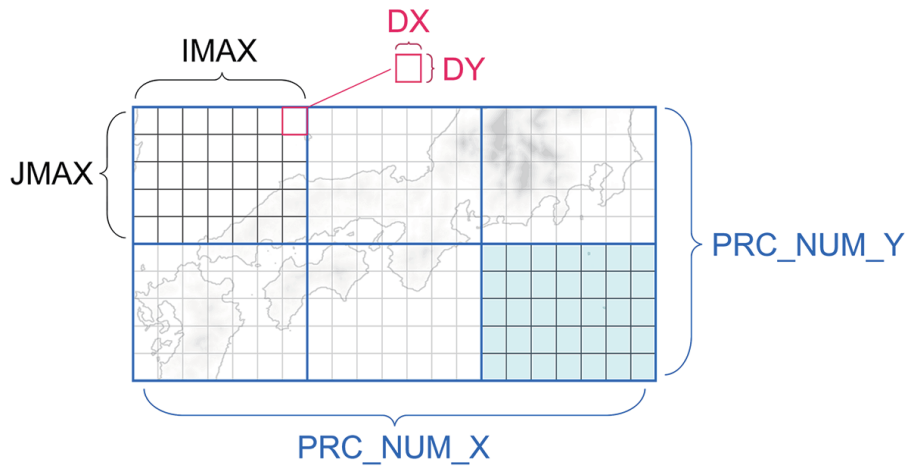


Figure 5.1: Relation between horizontal grid interval (DX , DY), number of grids ($IMAX$, $JMAX$) per MPI process, and the number of MPI processes (PRC_NUM_X , PRC_NUM_Y) in the entire domain. The light-blue part corresponds to a region managed by an MPI process.

5.1.1 Setting the number of MPI processes

The number of MPI processes is specified in `[PARAM_PRC]` in the configuration file. 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 condition file made by a two-MPI parallel combination cannot be used for model execution by a four-MPI parallel. It is necessary to edit `[PARAM_PRC]` in `pp.conf`, `init.conf`, and `run.conf` if the number of MPI processes is changed, and to conduct once again the processes of `pp` and `init`.

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

The total number of MPI processes is given by $PRC_NUM_X \times PRC_NUM_Y$. 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!
```

5.1.2 Setting the number of horizontal and vertical grids

The number of grids is specified in `[PARAM_INDEX]` in the configuration files `***.conf`. It should be noted that the specified horizontal grids are values controlled by one MPI process.

```

&PARAM_INDEX
  KMAX = 97,      ; number of vertical layers
  IMAX = 20,      ; number of grids along the X direction by an MPI process
  JMAX = 25,      ; number of grids along the Y direction by an MPI process
/

```

5.1.3 Setting grid intervals along the horizontal and vertical directions

Excluding the buffer region explained in Section 5.1.4, 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 zonal, meridional, and vertical grid intervals at (DX, DY, DZ) in [PARAM_GRID], respectively. The unit is [m].

```

&PARAM_GRID
  DX = 500.DO,    ; grid interval along the zonal (X) direction
  DY = 500.DO,    ; grid interval along the meridional (Y) direction
  DZ = 500.DO,    ; grid interval along the vertical (Z) direction
/

```

The box below shows the method to set how to specify a non-uniform grid system. Since the model employs the Lorenz 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. When the vertical grid locations are directly specified, provide them in (FZ(:)) in [PARAM_GRID] as an array.^{*1)} Refer to Figure 5.2 for the details. Note that the number of elements specified in (FZ(:)) should correspond to the number of vertical layers ((KMAX) in [PARAM_INDEX]). The following file for the ideal experiment is shown as an example:

```

&PARAM_GRID
  DX = 500.DO,      grid interval along the X direction (equidistant) [m]
  DY = 500.DO,      grid interval along the Y direction (equidistant) [m]
  FZ(:) =           location at face point along the Z direction [m]
    80.00000000000000 ,
    168.00000190734863 ,
    264.80000610351567 ,
    .....
    14910.428862936289 ,
    15517.262523292475 ,
    16215.121232702089 ,
    17017.658748523147 ,
    17940.576891717363 ,
    19001.932756390710 ,
    20222.492000765058 ,
  BUFFER_DZ = 5000.DO,  Refer to Section 5.1.4
  BUFFFACT = 1.0D0,     Refer to Section 5.1.4
/

```

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 freely. 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

^{*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.

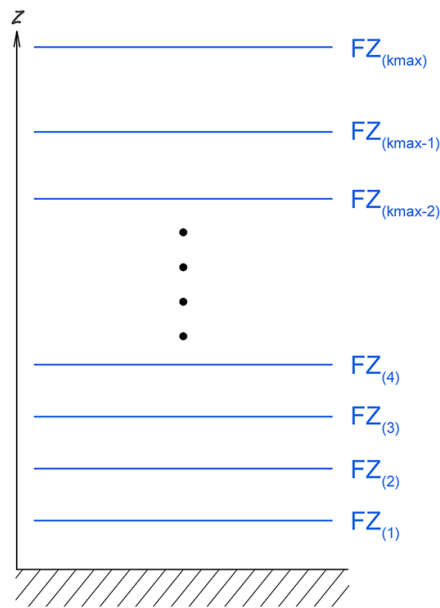


Figure 5.2: The point of definition of the face point in SCALE-RM . If (FZ) is given in [PARAM_GRID], the top height at the first layer is given for the value at $k = 1$. Note that $k = 1$ is not the ground surface height.

scale-5.1.5/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.

5.1.4 Setting the buffer region and nudging method

In general, nonphysical reflections of sound and gravity waves often occur at the top of the model boundary. Moreover, 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. 5.3, 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. Note that nudging at the upper layers is often called Rayleigh damping. The width of the buffer region is specified in [PARAM_GRID] in the configuration file. An example is as below. Note again that the configuration in all procedures must be identical.

```

&PARAM_GRID
BUFFER_DZ = 5000.DO, ; The width of the buffer region along the Z direction from the top of
                    ; the model (a reference) [m]
BUFFER_DX = 300000.DO, ; The width of the buffer region along the X (zonal) direction ( a reference
                    ; ) [m]
BUFFER_DY = 300000.DO, ; The width of the buffer region along the Y (meridional) direction ( a
                    ; reference ) [m]
BUFFFACT = 1.DO, ; Stretched factor for grid intervals in the buffer region( default : 1.0 )
/

```

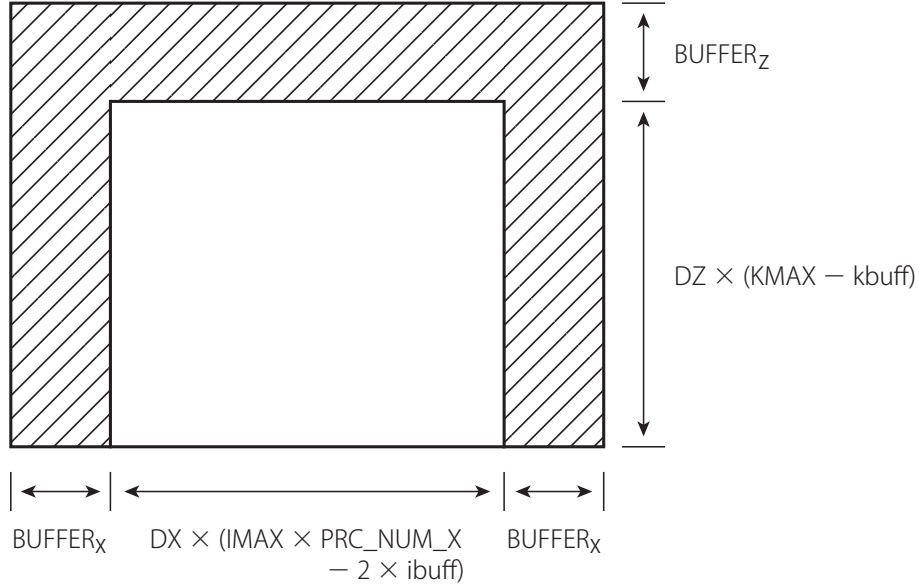


Figure 5.3: Location of the buffer region in the entire calculation domain: the shaded area indicates the buffer region. This figure shows the XZ cross-section. It is the same as the YZ cross-section.

The 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, and nothing is affected in the lower region. Note that the actual target region (the region excluding the buffer regions) narrows due to this setting because the buffer region is placed in the calculation domain.

The number of grids $ibuff$ in the buffer region is automatically calculated as the minimum integer satisfying the following condition:

$$\sum_{n=1}^{ibuff} BDX(n) \geq BUFFER_DX$$

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

$$DX \times (IMAX \times PRC_NUM_X - 2 \times ibuff). \quad (5.6)$$

The above explanation is for the X direction. 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

$$DZ \times (KMAX - kbuff), \quad (5.7)$$

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

In general, there is no clear criterion for setting the buffer region width and locating grids in the region. This depends on a problem to be solved. In SCALE-RM, the following 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, tune relaxation time, and so on. The relaxation time is defined as the time where the difference between the simulated value and the target value becomes $1/e$. This relaxation time is given in $(ATMOS_BOUNDARY_taux, ATMOS_BOUNDARY_tauy, ATMOS_BOUNDARY_tauz)$ in $[PARAM_ATMOS_BOUNDARY]$ in units of seconds.

Table 5.1: Map projections selectable in SCALE-RM

MPRJ_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

Method for stretching grid intervals in buffer region

The grid intervals in the buffer region are the same as (DX, DY, DZ) in [PARAM_GRID] by default. But, it is possible for them to be stretched by configuring (BUFFFACT). 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 5.1.3), 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, 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:

$$BUFFER_x = DX \times \frac{BUFFFACT^{ibuff} - 1}{BUFFFACT - 1} \quad (5.8)$$

Even if the same width of buffer region (BUFFER_DX) is specified, the number of grids in the buffer region decreases with increasing (BUFFFACT).

5.2 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_MAPPROJ). This configuration must be shared among the configuration files pp.conf, init.conf, and run.conf. An example is as follows:

```

&PARAM_MAPPROJ
  MPRJ_basepoint_lon = 138.727778D0,
  MPRJ_basepoint_lat = 35.360556D0,
  MPRJ_type = 'MER',           ; Choose from Table 5.1.
/

```

(MPRJ_basepoint_lat, MPRJ_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^{*2)}. In above setting, the center of the domain is configured at 35.360556°N and 138.727778°E,

^{*2)}It is possible to alternatively express either east longitude or west longitude as at an angle greater than 180°.

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.

(MPRJ_type) provides the kind of map projection and MER the Mercator projection. Table 5.1 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 (MPRJ_M_lat) in units of degrees. In general, the standard latitude is set to the Equator. In SCALE-RM , (MPRJ_basepoint_lat) is used as standard latitude, unless (MPRJ_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_MAPPROJ
MPRJ_basepoint_lon = 135.220404,
MPRJ_basepoint_lat = 34.653396,
MPRJ_type = 'LC',
MPRJ_LC_lat1 = 30.0,
MPRJ_LC_lat2 = 40.0,
/
```

In SCALE-RM , two standard parallels are used for this projection. The north and south of them are specified in (MPRJ_LC_lat1, MPRJ_LC_lat2) in units of degrees. In the region between the latitudes, the ratio of the latitudinal to longitudinal length is adjusted to be close to the Earth 's ellipsoid face.

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

```
&PARAM_MAPPROJ
MPRJ_basepoint_lon = 135.220404,
MPRJ_basepoint_lat = 34.653396,
MPRJ_basepoint_x = 100.0,
MPRJ_basepoint_y = 100.0,
MPRJ_type = 'LC',
MPRJ_LC_lat1 = 30.0,
MPRJ_LC_lat2 = 40.0,
/
```

The location of the reference point is specified as the distance from the south-west corner (bottom-left corner).; (MPRJ_basepoint_x) and (MPRJ_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 5.4 shows both cases.

5.3 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. The allowable maximum angle of inclination, θ_{\max} [radian] is calculated as follows:

$$\theta_{\max} = \arctan(\text{RATIO} \times \text{DZ}/\text{DX}),$$

where DZ and DX are the horizontal and vertical grid intervals, respectively. As shown in the above equation, θ_{\max} depends on spatial resolution. In SCALE-RM , the default value of RATIO

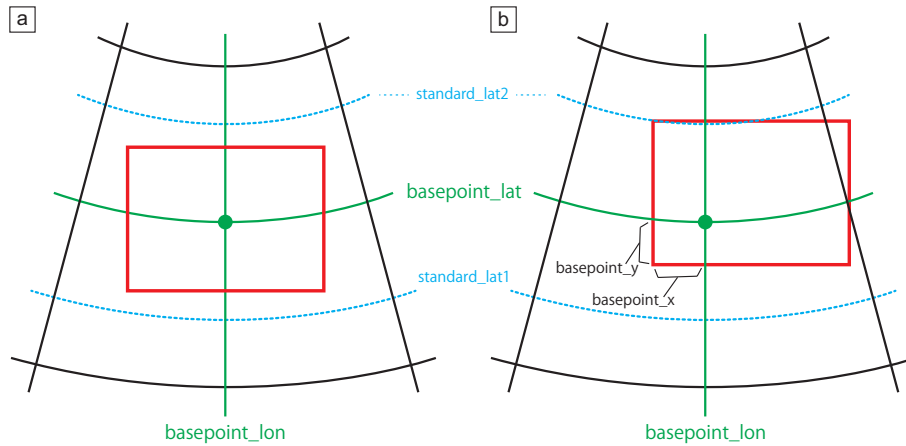


Figure 5.4: The relationship between projection center and calculation domain.; (a) Default setting and (b) the case where the center of projection is displaced at the center of the domain. The red line indicates the boundary of the calculation domain.

is 1.0. If RATIO is greater than unity, the fine topography is expressed, and vice versa. Note that at $RATIO > 1$, the risk of numerical instability increases.

The program `scale-rm_pp` converts external topography data into SCALE format. The detailed configurations are specified in `[PARAM_CNVTPOPO]` in configuration file `pp.conf`. An example is as follows:

```

&PARAM_CNVTPOPO
CNVTPOPO_name = "GTOPO30",           ; Data name used
CNVTPOPO_smooth_maxslope_ratio = 1.0, ; Allowable maximum ratio of inclination to DZ/DX
CNVTPOPO_smooth_local = .true.,      ; Whether smoothed or not, only grids whose angles
                                       of inclination exceed the maximum value
CNVTPOPO_copyparent = .false.,       ; Whether the topography in the buffer region is copied
/

```

SCALE-RM supports GTOPO30 and DEM50M provided by the Geospatial Information Authority of Japan as external input topography data.

The topographical data is formulated as the area-weighting mean of grid spacing. This conversion calculates the difference in altitude between them using their neighboring grids and inclinations. In case θ_{\max} exceeds the allowable maximum angle of inclination, i.e., `(CNVTPOPO_smooth_maxslope_ratio)`, the topography is smoothed by a Laplacian filter with several iterations until the angle is below the allowable maximum angle. Smoothing is selectively applied either only to grids whose angles of inclination exceed the maximum value, or to the entire domain. Since the former saves the sharp structure of the topography within the allowable maximum angle, it is selected if the representation of fine structures is desired. The Gaussian filter is also selected as a smoothing filter. In this case, the topography is smoother than when the Laplacian filter is used.

`(CNVTPOPO_copyparent)` is the item used for the nesting computation. In general, the topography in the child domain is finer than in the parent domain due to higher spatial resolution. At this time, problems often occurs due to an inconsistency between the atmospheric data in the buffer region of the child domain and that in the parent domain. To avoid this problem, the topography of the parent domain can be copied to the buffer region of the child domain by specifying `(CNVTPOPO_copyparent) = .true.` If there is no parent domain, `(CNVTPOPO_copyparent)` must be `.false.` Section 5.11.1 provides a more detailed explanation of the case that involves the use of `(CNVTPOPO_copyparent)`.

5.4 How to create initial and boundary data

Table 5.2: External input data supported in SCALE

Data type	FILETYPE_ORG	Note
SCALE format	SCALE-RM	Only history files are supported. The latitude-longitude catalog is needed.
Binary format	GrADS	Another namelist for data input is required.
WRF format	WRF-ARW	Both “wrfout” and “wrfst” are supported.

SCALE-RM can generate initial and boundary data by entering various types of external data, as shown in Table 5.2. The program `scale-rm_init` converts external data into boundary and initial data by configuring the file `init.conf`. The input data format is specified at (`FILETYPE_ORG`) in [`PARAM_MKINIT_REAL_***`]. The SCALE format is mainly used for offline nesting. Refer to Section 5.11.2 for details. The WRF data format is available; WRF model output data can be used directly. Note that the file should contain all data required for the generation of the boundary data of SCALE-RM. The “binary format” in this documentation is defined as binary data with single-precision floating points that FORTRAN can directly access. GRIB/GRIB2 data are available by converting them to binary format; this procedure is explained in Section 4.2. Other arbitrary data can also be used if it is converted into binary format.

Input from binary format data

The input data format is specified in [`PARAM_MKINIT_REAL_***`] in the configuration file `init.conf` as follows:

```
&PARAM_RESTART
  RESTART_OUTPUT = .true.,
  RESTART_OUT_BASENAME = "init_d01",
/

&PARAM_MKINIT_REAL_ATMOS
  NUMBER_OF_FILES = 2,
  FILETYPE_ORG = "GrADS",
  BASENAME_ORG = "namelist.grads_boundary.FNL.grib1",
  BASENAME_BOUNDARY = "boundary_d01",
  BOUNDARY_UPDATE_DT = 21600.0,
  PARENT_MP_TYPE = 3,
  USE_FILE_DENSITY = .false.,
/

&PARAM_MKINIT_REAL_OCEAN
  NUMBER_OF_FILES = 2,
  FILETYPE_ORG = "GrADS",
  BASENAME_ORG = "namelist.grads_boundary.FNL.grib1",
  INTRP_OCEAN_SFC_TEMP = "mask",
  INTRP_OCEAN_TEMP = "mask",
/

&PARAM_MKINIT_REAL_LAND
  NUMBER_OF_FILES = 2,
  FILETYPE_ORG = "GrADS",
  BASENAME_ORG = "namelist.grads_boundary.FNL.grib1",
  USE_FILE_LANDWATER = .true.,
  INTRP_LAND_TEMP = "mask",
  INTRP_LAND_WATER = "fill",
  INTRP_LAND_SFC_TEMP = "fill",
/
```

If binary data is entered, "GrADS" is given to (FILETYPE_ORG). In SCALE-RM , the namelist file `namelist.grads_boundary**`, which contains the file name and the structure of binary data, is prepared instead of the "ctl" file. Give its path at (BASENAME_ORG).

(NUMBER_OF_FILES) is the number of input files. In case of a single input file, prepare only file "filename.grd". In the case of multiple input files, prepare the files numbered as "filename.XXXX.grd" in the forward direction. The program `scale-rm_init` reads these files enumerated from 00000 to the given number (NUMBER_OF_FILES)-1. The header name of input files, i.e., "filename", is specified in the namelist file and explained later.

(BOUNDARY_UPDATE_DT) is the time step of input data. (RESTART_OUT_BASENAME) in [PARAM_RESTART] is the header name of the initial file converted. (BASENAME_BOUNDARY) is the header name of the boundary files converted.

The above configurations are the common among [PARAM_MKINIT_REAL_ATMOS], [PARAM_MKINIT_REAL_OCEAN], and [PARAM_MKINIT_REAL_LAND]. Unless otherwise specified in [PARAM_MKINIT_REAL_OCEAN] and [PARAM_MKINIT_REAL_LAND], these information are inherited.

(USE_FILE_DENSITY) is an option in case of FILETYPE_ORG="SCALE-RM". If binary data is selected, provide `.false.` to (USE_FILE_DENSITY). (PARENT_MP_TYPE) is the category type of the water substance in the parent model. If binary data format is entered, give 3 to (PARENT_MP_TYPE).

There are two options in preparation of soil moisture; one is a method to provide the data from the parent model and the other a method to provide it as a constant value in the entire region. In the former case, 3D soil moisture data are required. In the latter, configure `USE_FILE_LANDWATER = .false.` in [PARAM_MKINIT_REAL_LAND] in `init.conf`. The soil water condition is specified in `INIT_LANDWATER_RATIO` as the ratio of occupation of water to the void in the soil per unit volume (degree of saturation). The default value is 0.5. The size of void in the soil per unit volume (void ratio) depends on land use.

```

&PARAM_MKINIT_REAL_LAND
USE_FILE_LANDWATER = .false.  whether or not soil moisture is given by file. The default is
                             .true.
INIT_LANDWATER_RATIO = 0.5   in the case of USE_FILE_LANDWATER=.false.
                             the ratio of occupation of water to void (degree of saturation)
.....
/

```

If binary data (GrADS format) is used as input file, prepare them yourself. Refer to GrADS Web page (<http://cola.gmu.edu/grads/gadoc/aboutgriddeddata.html#structure>) for the format.

The following example is the namelist file `namelist.grads_boundary**` to provide information pertaining to the data file name and data structure in SCALE-RM in stead of "ctl" file.

```

#
# Dimension
#
&nml_grads_grid
  outer_nx = 360, ; the number of grids of the atmosphere along the x direction
  outer_ny = 181, ; the number of grids of the atmosphere along the y direction
  outer_nz = 26, ; the number of layers for the atmosphere
  outer_nl = 4, ; the number of layers for soil data
/

#
# Variables
#
&grdvar item='lon', dtype='linear', swpoint=0.0d0, dd=1.0d0 /
&grdvar item='lat', dtype='linear', swpoint=90.0d0, dd=-1.0d0 /
&grdvar item='plev', dtype='levels', lnum=26,
  lvars=100000,97500,.....,2000,1000, /
&grdvar item='MSLP', dtype='map', fname='FNLSfc', startrec=1, totalrec=6 /
&grdvar item='PSFC', dtype='map', fname='FNLSfc', startrec=2, totalrec=6 /
&grdvar item='U10', dtype='map', fname='FNLSfc', startrec=3, totalrec=6 /
&grdvar item='V10', dtype='map', fname='FNLSfc', startrec=4, totalrec=6 /
&grdvar item='T2', dtype='map', fname='FNLSfc', startrec=5, totalrec=6 /
&grdvar item='RH2', dtype='map', fname='FNLSfc', startrec=6, totalrec=6 /
&grdvar item='HGT', dtype='map', fname='FNLatm', startrec=1, totalrec=125 /
&grdvar item='U', dtype='map', fname='FNLatm', startrec=27, totalrec=125 /
&grdvar item='V', dtype='map', fname='FNLatm', startrec=53, totalrec=125 /
&grdvar item='T', dtype='map', fname='FNLatm', startrec=79, totalrec=125 /
&grdvar item='RH', dtype='map', fname='FNLatm', startrec=105, totalrec=125, knum=21 /
&grdvar item='llev', dtype='levels', lnum=4, lvars=0.05,0.25,0.70,1.50, /
&grdvar item='lsmask', dtype='map', fname='FNLLand', startrec=1, totalrec=10 /
&grdvar item='SKINT', dtype='map', fname='FNLLand', startrec=2, totalrec=10 /
&grdvar item='STEMP', dtype='map', fname='FNLLand', startrec=3, totalrec=10,
  missval=9.999e+20 /
&grdvar item='SMOISVC', dtype='map', fname='FNLLand', startrec=7, totalrec=10,
  missval=9.999e+20 /

```

The number of grids in the atmosphere is specified as `outer_nx`, `outer_ny`, `outer_nz`, and the number of layers of soil data (`STEMP`, `SMOISVC`) is specified as `outer_nl`.

The input data to `QV` and `RH` is not often provided in the upper layers. In such cases, the number of layers where the data exist is specified as `knum`. Two methods of giving values to the upper layers are prepared. As default, `upper_qv_type = "ZERO"` as

```

&PARAM_MKINIT_REAL_GrADS
  upper_qv_type = "ZERO"   "ZERO": QV=0
                          "COPY": copy the RH at the top layer where input humidity data exists
                          to the upper layers without the data
/

```

The configuration of `[grdvar]` is different by data, as shown in Table 5.3. The list of `[grdvar]` is shown in Table 5.4. In Table 5.4, soil moisture (fraction of volume) is the ratio of water volume (V_w) to soil volume (V), i.e., V_w/V . The saturation ratio is the ratio of water volume V_w to void volume in V , i.e., V_w/V_v . If `(USE_FILE_LANDWATER)=.true.` in `[PARAM_MKINIT_REAL_LAND]`, prepare data either for `SMOISVC` or of `SMOISDS`.

Table 5.3: Variables of [grdvar]

item of grdvar	Explanation	Note
item	Variable name	Select from Table 5.4
dtype	Data type	"linear" or "levels" or "map"
namelist at (dtype)="linear" (Specific use of "lon", "lat")		
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"		
fname	Header name of files	
startrec	Recorded number of variables (item)	time at t=1
totalrec	Recorded length of all variables per time	
knum	Number of layers of 3D data	(option) in the case of specifying value that differs from outer_nz available for RH and QV
missval	missing value	(option)

Table 5.4: Variable list of (item) in [grdvar]. * means "required."

Variable name	Explanation	Unit	(dtype)
* lon	longitude data	[deg.]	linear, map
* lat	latitude data	[deg.]	linear, map
* plev	pressure data	[Pa]	levels, map
* HGT	geopotential height data	[m]	map
* U	eastward wind speed	[m/s]	map
* V	northward wind speed	[m/s]	map
* T	temperature	[K]	map
* QV	specific humidity (optional if RH is given)	[kg/kg]	map
* RH	relative humidity (optional if QV is given)	[%]	map
MSLP	sea level pressure	[Pa]	map
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
Q2	2m specific humidity (optional if RH2 is given)	[kg/kg]	map
RH2	2m relative humidity (optional if Q2 is given)	[%]	map
TOPO	topography of GCM	[m]	map
lsmask	ocean-land distribution of GCM	0:ocean1:land	map
* SKINT	surface temperature	[K]	map
* llev	soil depth	[m]	levels
* STEMP	soil temperature	[K]	map
(*) SMOISVC	soil moisture (volume fraction)	[-]	map
(*) SMOISDS	soil moisture (saturation ratio)	[-]	map
* SST	sea surface temperature (optional if SKINT is given)	[K]	map

5.5 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 `run.conf`.

```

&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_STARTTMS = 0.DO, Start date [mili sec]
TIME_DURATION = 12.0DO, Integration time [init is defined by
                           TIME_DURATION_UNIT]
TIME_DURATION_UNIT = "HOUR", Unit for TIME_DURATION
TIME_DT = 60.0DO, Time step for time integration
TIME_DT_UNIT = "SEC", Unit for TIME_DT
TIME_DT_ATMOS_DYN = 30.0DO, Time step for calculation of dynamical process
TIME_DT_ATMOS_DYN_UNIT = "SEC", Unit for TIME_DT_ATMOS_DYN
TIME_DT_ATMOS_PHY_MP = 60.0DO, 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_TB = 60.0DO, 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_RD = 600.0DO, 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 = 60.0DO, 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_OCEAN = 300.0DO, Time step for calculation of ocean process
TIME_DT_OCEAN_UNIT = "SEC", Unit for TIME_DT_OCEAN
TIME_DT_LAND = 300.0DO, Time step for calculation of land process
TIME_DT_LAND_UNIT = "SEC", Unit for TIME_DT_LAND
TIME_DT_URBAN = 300.0DO, Time step for calculation of urban process
TIME_DT_URBAN_UNIT = "SEC", Unit for TIME_DT_URBAN
TIME_DT_ATMOS_RESTART = 21600.DO, 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.DO, 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.DO, 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.DO, Output interval of restart files for urban variables
TIME_DT_URBAN_RESTART_UNIT = "SEC", Unit for TIME_DT_URBAN_RESTART
/

```

`(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 shorter time step for dynamical process, i.e. `(TIME_DT_ATMOS_DYN)`, should be given than Δt because the time integration of dynamic variables is constrained not by advection velocity, but by the speed of the acoustic wave. Note that `(TIME_DT_ATMOS_DYN)` needs to be a divisor of `(TIME_DT)`. `(TIME_DT_ATMOS_DYN)` depends on the time integration scheme in relation to the stability of calculation. As 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.

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 are better as multiples of `(TIME_DT)`.

The surface fluxes are calculated by the surface process for the atmosphere. On the contrary, if ocean, land, and urban models are used, the fluxes are calculated by these models. A model grid contains several types of land use, such as ocean, urban, and land. 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 5.10.

5.6 Setting the history file and output variable

A history file and output variables are configured at `[PARAM_HISTORY]` and `[HISTITEM]` in `run.conf`. The default format of the history file is configured at `[PARAM_HISTORY]`.

```

&PARAM_HISTORY
  HISTORY_DEFAULT_BASENAME = "history_d01", ; Header name of files
  HISTORY_DEFAULT_TINTERVAL = 3600.0, ; Output interval of history data
  HISTORY_DEFAULT_TUNIT = "SEC", ; Unit of HISTORY_DEFAULT_TINTERVAL
  HISTORY_DEFAULT_TAVERAGE = .false., ; .false.: snapshot value, .true.: Average
  ; value
  HISTORY_DEFAULT_ZCOORD = "model", ; Types of vertical interpolation.
  ; native, AGL, MDL: model level (not interpo-
  ; lated)
  ; z, Z, HGT: Interpolated value at absolute height
  ; pz, P, PRES: Interpolated value at pressure level
  HISTORY_DEFAULT_DATATYPE = "REAL4", ; Output data type: REAL4, REAL8
  HISTORY_OUTPUT_STEPO = .true., ; Whether or not data are output at initial time
  ; (t=0).
  ; .true.: output, .false.: Not output.
  History_PRES_VALUE = -1, ; Number of levels
  ; (OPTION for interpolation at pressure level)
  History_PRES_NLAYER = 0.0*100, ; Pressure levels in order from lower to upper
  ; level. The unit is hPa.
  ; (OPTION for interpolation at pressure level)
/

```

The unit for `(HISTORY_DEFAULT_TUNIT)` can be selected from "MSEC", "msec", "SEC", "sec", "s", "MIN", "min", "HOUR", "hour", "h", "DAY", "day". When the average value output is selected as `HISTORY_DEFAULT_TAVERAGE = .true.`, the history data averaged over the last period, given as `HISTORY_DEFAULT_TINTERVAL`, is output.

The output variables are set by adding `[HISTITEM]`. The output format follows the default setting specified at `[PARAM_HISTORY]`. By adding namelists with "(OPTION)," the format for a specific variable can be changed from the default setting.

```

&HISTITEM
ITEM = "RAIN",           Variable name. List of variables is shown in Appendix A.
BASENAME = "rain_d01",  (OPTION) same as HISTORY_DEFAULT_BASENAME
TINTERVAL= 600.0,       (OPTION) same as HISTORY_DEFAULT_TINTERVAL
TUNIT = "SEC",          (OPTION) same as HISTORY_DEFAULT_TINTERVAL
TAVERAGE = .true.,     (OPTION) same as HISTORY_DEFAULT_TAVERAGE
ZCOORD = "model",      (OPTION) same as HISTORY_DEFAULT_ZCOORD
DATATYPE = "REAL4",    (OPTION) same as HISTORY_DEFAULT_DATATYPE
/

```

Namelists with “(OPTION)” are applied only to the variable (**ITEM**). If the default setting is used for the variable, the namelist with “(OPTION)” can be omitted. For example, let the below setting for [**HISTITEM**] be added, maintaining the above setting of [**PARAM_HISTORY**]. The snapshot values of T, U, 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.

```

&HISTITEM ITEM = "T" /
&HISTITEM ITEM = "U" /
&HISTITEM ITEM = "V" /
&HISTITEM ITEM = "RAIN", TINTERVAL = 600.0, TAVERAGE = .true. /

```

5.7 Setting the dynamical core

Setting the 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 = "HEVI", ; Choose from Table 5.5.
/

```

Table 5.5: 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

Setting the temporal and spatial difference schemes

The temporal integration and spatial difference schemes are configured in [**PARAM_ATMOS_DYN**]. According to the spatial integration scheme, the specification of the number of halos is required. The following are recommended along with the other options listed in Table 5.6. For numerical stability, the time step is changed according to the schemes used. Methods to determine the time step are described in Section 5.5.


```

&PARAM_ATMOS_DYN
ATMOS_DYN_TINTEG_SHORT_TYPE = RK4, ; Choose from temporal integration schemes
; in Table 5.6
ATMOS_DYN_TINTEG_TRACER_TYPE = RK3WS2002, ; Choose from temporal integration schemes
ATMOS_DYN_FVM_FLUX_TYPE = CD4, ; Choose from temporal spatial difference
; schemes
ATMOS_DYN_FVM_FLUX_TRACER_TYPE = UD3KOREN1993, ; Choose from temporal spatial difference
; schemes
ATMOS_DYN_FLAG_FCT_TRACER = .false., ; Use FCT scheme (.true.) or not (.false.)
ATMOS_DYN_NUMERICAL_DIFF_COEF = 1.D-2,
ATMOS_DYN_NUMERICAL_DIFF_COEF_TRACER = 0.DO,
ATMOS_DYN_enable_coriolis = .true.,
/

```

Table 5.6: Setting time integration and spatial difference schemes

Scheme name	Description of scheme	
Temporal integration		
RK3	Heun-type 3rd-order Runge–Kutta scheme	
RK3WS2002	Wicker and Skamarock (2002)’s 3-step Runge–Kutta scheme	
RK4	4th-order Runge–Kutta scheme	
Spatial difference		Minimum number of halos
CD2	2nd-order central difference	1
CD4	4th-order central difference	2
CD6	6th-order central difference	3
UD3	3rd-order upwind difference	2
UD5	5th-order upwind difference	3
UD3KOREN1993	3rd-order upwind scheme + Koren (1993)’s filter	2

For tracer advection, the use of a scheme for guaranteeing a non-negative value is recommended. The (UD3KOREN1993) scheme guarantees a non-negative value, whereas other schemes do not. When schemes other than (UD3KOREN1993) are used by being specified in (ATMOS_DYN_FVM_FLUX_TRACER_TYPE), the FCT filter is recommended as (ATMOS_DYN_FLAG_FCT_TRACER)=.true.

By default, the number of halos is 2. The configuration of the halo is required according to the number shown in Table 5.6, except for the spatial difference schemes of CD4 and UD3. For example, the configuration of the halo for the fifth-order upwind difference scheme is as follows:

```

&PARAM_INDEX
IHALO = 3,
JHALO = 3,
/

```

5.8 Setting the physical process

5.8.1 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 both `init.conf` and `run.conf`. The update interval for the cloud micro-physics scheme is specified in `[PARAM_TIME]`. Refer to Section 5.5 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 5.7.
/
```

Table 5.7: Choices of cloud micro-physics scheme

Scheme name	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:

- 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.
- 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.
- 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.
- 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 files of `init.conf` and `run.conf`:

```
&PARAM_BIN
  nbin = 33, & : The number of bins
  ICEFLG = 1, & : Option for consideration of ice cloud: 0(not considered), 1(considered)
/
```

In this case, `[PARAM_BIN]` in `init.conf` must also be same as in `run.conf`. 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:

```
xxx 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`.

5.8.2 Turbulence scheme

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 5.5 for the detailed configuration of the calling timing.

```
&PARAM_ATMOS
  ATMOS_PHY_TB_TYPE = "MYNN", ; Select the scheme shown in Table 5.8
/
```

Table 5.8: Choices of turbulent scheme

Value	Description of scheme	Reference
OFF	Do not calculate the turbulence process	
SMAGORINSKY	Smagorinsky Lily-type sub-grid model	Smagorinsky (1963); Lilly (1962); Brown et al. (1994); Scotti et al. (1993)
D1980	Deardorff-type sub-grid model	Deardorff (1980)
MYNN	MYNN Level 2.5 boundary scheme	Mellor and Yamada (1982); Nakanishi and Niino (2004)

5.8.3 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 5.5 for the detailed configuration of calling timing.

```
&PARAM_ATMOS
  ATMOS_PHY_RD_TYPE = "MSTRNX", ; Select the radiation scheme shown in Table 5.9
/
```

Table 5.9: Choices of radiation scheme

Value	Explanation of scheme	Reference
OFF or NONE	Do not calculate the radiation scheme	
MSTRNX	mstrnX (A k-distribution-based broadband radiation transfer model)	Sekiguchi and Nakajima (2008)

The solar radiation is calculated by using date and time, and latitude and longitude, in the model domain. 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_fixedlatlon = .false.        Whether values are fixed for latitude and longi-
                                              tude at radiation calculation
  ATMOS_SOLARINS_fixeddate = .false.         Whether values are fixed for date and time at ra-
                                              diation calculation
  ATMOS_SOLARINS_lon = 135.221               Longitude in case that
  ATMOS_SOLARINS_lat = 34.653               ATMOS_SOLARINS_fixedlatlon=.true. [deg.]
                                              Latitude in case that
  ATMOS_SOLARINS_date = -1,-1,-1,-1,-1,-1,  ATMOS_SOLARINS_fixedlatlon=.true. [deg.]
                                              Year, month, day, and time in the case that
                                              ATMOS_SOLARINS_fixeddate=.true. [Y,M,D,H,M,S]
/

```

If (MPRJ_basepoint_lon,MPRJ_basepoint_lat) in [PARAM_MAPPROJ] are given, (ATMOS_SOLARINS_lon, ATMOS_SOLARINS_lat) are set identical to the values of (MPRJ_basepoint_lon, MPRJ_basepoint_lat). Refer to Section 5.2 for the explanation of [PARAM_MAPPROJ].

Depending on experimental design, the top of the model is often too low, such as 10~20km, compared to the height of atmosphere. To remedy this situation, another top height used only for radiation calculation is set. The number of layers to be placed at levels higher than the top of the model can be configured. The top for radiation depends on the parameter file of the radiation scheme. For example, when MSTRNX is used, the default parameter table in MSTRNX assumes that it is 100 km. The additional layers added 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] 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 = 11            Number of aerosol species
                                              (depending on parameter file
                                              used)
  ATMOS_PHY_RD_MSTRN_nradius = 6            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
                                              parameter by gas
  ATMOS_PHY_RD_MSTRN_AEROPARA_IN_FILENAME = "PARAPC.29"  Input file for absorption
                                              and scattering parameter by
                                              aerosol
  ATMOS_PHY_RD_MSTRN_HYGROPARA_IN_FILENAME = "VARDATA.RM29"  Input file for aerosol particle
                                              parameter
  ATMOS_PHY_RD_MSTRN_ONLY_QCI = .false.     Whether rain, snow, and
                                              graupel are considered
/

```

MSTRNX requires a parameter table for radiation calculation. By default, the wavelength from

solar radiation to infrared radiation is divided into 29 bands/111 channels; 11 types of cloud and aerosol particles with six particle bins are prepared in the table. Three kinds of parameter files are prepared in the directory `scale-rm/test/data/rad/`. The file and directory are specified in `[PARAM_ATMOS_PHY_RD_MSTRN]`.

It is necessary to provide vertical profiles in additional layers for temperature, pressure, and gas concentration, such as CO₂ and ozone. 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 for CIRA86^{*3)} (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 according to date and time, and latitude and longitude are calculated in the model domain. These input files are also provided in the directory `scale-rm/test/data/rad/`. The file and directory names are specified in `[PARAM_ATMOS_PHY_RD_PROFILE]`.

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 the other gases are set to zero and temporal variation is not considered. An example of user-defined files is given for the directory `scale-rm/test/data/rad/`. Given as `(ATMOS_PHY_RD_PROFILE_use_climatology) = .false.` in `[PARAM_ATMOS_PHY_RD_PROFILE]`, the file and directory names are required to be specified.

<code>&PARAM_ATMOS_PHY_RD_PROFILE</code>	
<code>ATMOS_PHY_RD_PROFILE_use_climatology = .true.</code>	Whether climatologies of CIRA86 and MIPAS2001 are used
<code>ATMOS_PHY_RD_PROFILE_CIRA86_IN_FILENAME = "cira.nc"</code>	File name of CIRA86 (NetCDF format)
<code>ATMOS_PHY_RD_PROFILE_MIPAS2001_IN_BASENAME = "."</code>	Directory name of MIPAS2001 (ASCII format)
<code>ATMOS_PHY_RD_PROFILE_USER_IN_FILENAME = ""</code>	User-defined file in the case of not using climatologies (ASCII format)
<code>ATMOS_PHY_RD_PROFILE_USE_CO2 = .true.</code>	When false, CO ₂ concentration is zero.
<code>ATMOS_PHY_RD_PROFILE_USE_O3 = .true.</code>	When false, O ₃ concentration is zero.
<code>ATMOS_PHY_RD_PROFILE_USE_N2O = .true.</code>	When false, N ₂ O concentration is zero.
<code>ATMOS_PHY_RD_PROFILE_USE_CO = .true.</code>	When false, CO concentration is zero.
<code>ATMOS_PHY_RD_PROFILE_USE_CH4 = .true.</code>	When false, CH ₄ concentration is zero.
<code>ATMOS_PHY_RD_PROFILE_USE_O2 = .true.</code>	When false, O ₂ concentration is zero.
<code>ATMOS_PHY_RD_PROFILE_USE_CFC = .true.</code>	When false, CFC concentration is zero.
<code>/</code>	

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.

5.8.4 Surface flux at the bottom of atmospheric boundary

The flux scheme at the bottom atmospheric boundary is configured in `(ATMOS_PHY_SF_TYPE)` in `[PARAM_ATMOS]` as follows:

*3) <http://catalogue.ceda.ac.uk/uuid/4996e5b2f53ce0b1f2072adadaeda262>

```
&PARAM_ATMOS
```

```
ATMOS_PHY_SF_TYPE = "COUPLE", ; Select the surface flux scheme shown in Table 5.10
```

```
/
```

If ocean, land, and urban models are not used, the bottom atmospheric boundary is assumed to be a virtual surface used in an ideal experiment. The timing of the calling of the surface flux scheme is configured in [PARAM_TIME]. Refer to Section 5.5 for the detailed configuration of the calling timing. If ocean, land, and urban models are used, “COUPLE” is given to (ATMOS_PHY_SF_TYPE):

Table 5.10: Choices for the atmospheric bottom boundary

Value	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	Fix the constant value of surface flux
BULK	Calculate the surface flux in bulk mode
COUPLE	Receive surface flux from ocean, land, and urban models

Configuration of constant

If (ATMOS_PHY_SF_TYPE) = "CONST", the surface flux can be kept to a value specified in file run.conf as follows. The values below are the default ones.

```
&PARAM_ATMOS_PHY_SF_CONST
```

```
ATMOS_PHY_SF_FLG_MOM_FLUX = 0
```

0: Bulk coefficient is constant

1: Frictional 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

(Active at ATMOS_PHY_SF_FLG_MOM_FLUX = 0)

```
ATMOS_PHY_SF_CM_min = 1.0E-5
```

Lower limit of bulk coefficient for momentum

(Active at ATMOS_PHY_SF_FLG_MOM_FLUX = 1)

```
ATMOS_PHY_SF_Const_Ustar = 0.25E0
```

Constant frictional velocity [m/s]

(Active at ATMOS_PHY_SF_FLG_MOM_FLUX = 1)

```
ATMOS_PHY_SF_Const_SH = 15.E0
```

Constant sensible heat flux at the surface [W/m²]

```
ATMOS_PHY_SF_FLG_SH_DIURNAL = .false.
```

Whether diurnal variation is enabled 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/m²]

```
/
```

Bulk configuration

If (ATMOS_PHY_SF_TYPE) = "BULK", the surface flux is calculated by the bulk model using the surface temperature. The roughness is calculated by the scheme for ocean roughness as default. On the contrary, ocean roughness can be given as a constant value, and evaporation efficiency can be given arbitrarily. This flexibility enables the ideal experiment not only for ocean surfaces but also for land. The scheme for ocean roughness is specified in (ROUGHNESS_TYPE) in [PARAM_ROUGHNESS] in the file run.conf as follows:

```
&PARAM_ROUGHNESS
ROUGHNESS_TYPE = "MOON07", ; Select the radiation scheme shown in Table 5.11
/
```

Table 5.11: Choices of ocean roughness

Value	Description of scheme	Reference
MOON07	Based on empirical formula with time development as default	Moon et al. (2007)
MILLER92	Based on empirical formula without time development	Miller et al. (1992)
CONST	Constant value	

The scheme for the bulk coefficient is configured in `(BULKFLUX_TYPE)` in `[PARAM_BULKFLUX]` in the file `run.conf` as follows:

```
&PARAM_BULKFLUX
BULKFLUX_TYPE = "B91W01", ; Select the radiation scheme shown in Table 5.12
/
```

Table 5.12: Choices of bulk coefficient scheme

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

5.8.5 Ocean model

The ocean process consists of two main parts, i.e., update of sea surface state and flux calculation at the interface of atmosphere and ocean. The timing of calling these scheme is configured in `[PARAM_TIME]`. Refer to Section 5.5 for the detailed configuration of the calling timing.

Sea surface scheme

The manner of updating sea surface state, such as sea surface temperature, is configured in `(OCEAN_TYPE)` in `[PARAM_OCEAN]` in the files `init.conf` and `run.conf`:

```
&PARAM_OCEAN
OCEAN_TYPE = "CONST", ; Select the sea surface type shown in Table 5.13
/
```

If the ocean category is included in land-use setup by `[PARAM_LANDUSE]`, neither "NONE" nor "OFF" can be given to `(OCEAN_TYPE)`. If this condition is not satisfied, the program immediately stops without the computation, and outputs the following message to LOG file:

```
xxx Ocean fraction exists, but ocean components never called. STOP.
```

Table 5.13: Choices of sea surface schemes

Values	Description of scheme
NONE or OFF	Not using the sea surface model
CONST	Fixed as initial condition
FILE	Input from external data (temporal variation is available)
SLAB	Slab ocean model

If `(OCEAN_TYPE) = "FILE"`, it is necessary to specify an external input file name in files `init.conf` and `run.conf`. In this case, the sea surface temperature varies with time according to spatial distribution and the temporal history of the external file.

```
&EXTITEM
  basename = "../init/output/ocean_d01", ; input file
  varname = "OCEAN_TEMP",
  step_limit = 1800,
  step_fixed = -1,
  enable_periodic_year = .false.,
  enable_periodic_month = .false.,
  enable_periodic_day = .false.,
/
```

If `(OCEAN_TYPE) = "SLAB"`, the depth of the mixed slab layer can be specified in files `init.conf` and `run.conf`. In this case, the temperature in the mixed slab layer develops over time through heat flux exchange between the atmosphere and the ocean:

```
&PARAM_OCEAN_PHY_SLAB
  OCEAN_PHY_SLAB_DEPTH = 10.0D0, ; Default [m]
/
```

The albedo on the sea surface does not depend on the scheme selected, and is calculated by using only the solar zenith angle.

Flux in atmosphere and ocean

Once a sea surface scheme except for NONE or OFF is specified, flux is exchanged at the interface of atmosphere and ocean. The fluxes between the atmosphere and the ocean are calculated by some kind of bulk schemes contained in SCALE-RM. The scheme for the length of the roughness on the sea surface is also selected from several schemes prepared. The scheme of roughness is specified in `(ROUGHNESS_TYPE)` in `[PARAM_ROUGHNESS]` in file `run.conf`. The scheme for the bulk transfer coefficient is specified in `(BULKFLUX_TYPE)` in `[PARAM_BULKFLUX]` in file `run.conf`. Refer to Section 5.8.4 for more details.

5.8.6 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 flux at the interface of atmosphere and land. The timing of the calling of these scheme is configured in `[PARAM_TIME]`. Refer to Section 5.5 for the detailed configuration of calling timing.

Land surface scheme

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

```
&PARAM_LAND
LAND_TYPE = "SLAB", ; Select the land type shown in Table 5.14
/
```

Table 5.14: Choices of land surface scheme

Values	Description of scheme
NONE or OFF	Do not use land surface model
SLAB	Heat diffusion/bucket model
CONST	Do not update soil temperature, soil moisture, and land surface temperature in SLAB

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

```
xxx Land fraction exists, but land components never called. STOP.
```

If `(LAND_TYPE) = "SLAB"` or `(LAND_TYPE) = "CONST"`, it is necessary to prepare parameter tables for the length of roughness and the input to the land-use distribution. A parameter table is provided in the file `scale-rm/test/data/land/param.bucket.conf`.

Flux in atmosphere and on land

Once any land surface scheme except NONE or OFF is specified, flux is exchanged at the interface of the atmosphere and land. The flux between atmosphere and land is calculated by some kind of land scheme contained in SCALE-RM. If `(LAND_TYPE) = "SLAB"` or `(LAND_TYPE) = "CONST"`, the bulk scheme is the same as that for ocean or ideal land surface. This is specified in `(BULKFLUX_TYPE)` in `[PARAM_BULKFLUX]` in file `run.conf`. Refer to Section 5.8.4 for details.

5.8.7 Urban model

The urban process consists of two main parts, i.e., updating the urban state and flux calculation at the interface of the atmosphere and the urban environment. The timing of calling these scheme is configured in `[PARAM_TIME]`. Refer to Section 5.5 for the detailed configuration of calling timing.

Flux in atmosphere and urban area

The urban scheme that updates the urban surface state, e.g., urban surface temperature and moisture, and calculates flux at the interface of atmosphere and the urban environment is configured in `(URBAN_TYPE)` in `[PARAM_URBAN]` in `init.conf` and `run.conf`, as follows:

```
&PARAM_URBAN
URBAN_TYPE = "SLC", ; Select the urban type shown in Table 5.15
/
```

Table 5.15: Choices of urban scheme

Value	Description of scheme	reference
NONE or OFF	Do not use the urban scheme	
SLC	Single-layer canopy model	Kusaka et al. (2001)

If the type of urban area is included in the land-use setup by `[PARAM LANDUSE]`, neither NONE nor OFF can be given to `(URBAN_TYPE)`. If this condition is not satisfied, the program immediately stops without computation, outputting the following message to LOG file:

```
xxx Urban fraction exists, but urban components never called. STOP.
```

5.9 Post-processing

In this section, the post-processing tool `net2g` is described and its function explained. `net2g` generates the binary format that is readable in GrADS by combining history files `history.***.nc` with the netCDF format. The conversion of all files to one enable GrADS to draw the result. Thus, formatted data can be easily analyzed, even by a FORTRAN serial program. Since `net2g` can also be executed as an MPI parallel program, the elapsed time can be to a greater extent than in the case of serial processing. The following functions are also available

- Interpolate data from the surface of the model to arbitrary height coordinates or pressure coordinates.
- Output the mean, maximum, and minimum values of the vertical integration of 3D variables.
- Output multiple files for 3D variables, layer by layer.
- Output multiple files, time-step by time-step.

It is convenient to divide files layer by layer or time-step by time-step because such data can be easily handled, particularly large-scale computational data. Refer to Section 2.4 for a guide to installing `net2g`. Note that the current version of `net2g` has the following limitations:

- The number of MPI processes in `net2g` must be a divisor of the number of MPI processes at execution in SCALE-RM .
- The history output format at the execution of SCALE-RM must be output with `(HIST_BND) = .false.` in `[PARAM_HIST]`.
- Two- and three-dimensional data cannot be simultaneously converted.
- Only history data can be converted.

Beware that if too large a number of MPI processes is set, computational performance is affected. Since most instruction in `net2g` are for the input/output of data, it is possible that the number of requests for storage access becomes too large. In particular, this situation should be attended to if the conversion of large-scale computation results is carried out on small-scale machines.

If MPI parallel is used, `net2g` is executed as follows:

```
$ mpirun -n [number of process] ./net2g net2g.conf
```

The last argument `net2g.conf` is the configuration file for `net2g`. On the contrary, if `net2g` is compiled as a single process version,

```
$ ./net2g net2g.conf
```

If only the following message without error can be found, the execution is concluded normally:

```
+++ MPI COMM: Corrective Finalize
```

The following explains how to describe the configuration files in the case of 2D or 3D variables by using sample files `net2g.3d.conf` and `net2g.2d.conf` in the directory `scale-5.1.5/scale-rm/util/netcdf2grads_h/`. In this section, only the major issues pertaining to its use are treated. Refer to the sample files `net2g.all.conf` in `scale-5.1.5/scale-rm/util/netcdf2grads_h/` for the other options.

The sample configuration file: Conversion of 3D variables

```
&LOGOUT
LOG_BASENAME = "LOG_d01_3d",
LOG_ALL_OUTPUT = .false.,
/

&INFO
TIME_STARTDATE = 2000, 1, 1, 0, 0, 0,
START_TSTEP = 1,
END_TSTEP = 25,
DOMAIN_NUM = 1,
CONFFILE = "../run/run.d01.conf",
IDIR = "../run",
Z_LEV_TYPE = "plev",
MAPPROJ_ctl = .true.
/

&VARI
VNAME = "PT", "U", "V", "W", "QHYD",
TARGET_ZLEV = 850, 500, 200,
/
```

The above example shows a configuration in the case where 3D variables in a domain are converted into pressure and/or height coordinates. The setting items are as follows:

- **[LOGOUT]** (The following items are not required.)
 - **(LOG_BASENAME)** : If the default LOG file name LOG is changed, this item is specified.
 - **(LOG_ALL_OUTPUT)** : If processes other than the 0th process are output in the LOG files, ".true." is assigned to this item. The default value is ".false".
- **[INFO]**
 - **(TIME_STARTDATE)** : The start date and time of converted netCDF data are specified.
 - **(START_TSTEP)** : The start time step of converted netCDF data is specified. If several first steps are skipped, the appropriate values are assigned to this item. The default value is 1.
 - **(END_TSTEP)** : The end of the time step of converted netCDF data is specified. It is required in all cases.

- (DOMAIN_NUM) : The domain number is specified. The default value is 1.
 - (CONFFILE) : The path of run.***.conf at the execution of SCALE-RM is specified, including the file name.
 - (IDIR) : The path of history files is specified.
 - (Z_LEV_TYPE) : The type of vertical data conversion is specified. "original" represents the model surface, "plev" the interpolation to the pressure surface, and "zlev" that to the height surface. If "anal" is specified, the result with a simple analysis is output. The details for "anal" is explained subsequently. The default value is "plev".
 - (MAPPROJ_ctl) : It indicates whether the "ctl" file corresponding to the map projection using pdef is output.
- [VARI]
 - (VNAME) : The variables converted are specified. As default, "PT","PRES","U","V", "W","QHYD" are given.
 - (TARGET_ZLEV) : The heights corresponding to (Z_LEV_TYPE) are specified. In the case of "plev", the unit is [hPa], in the case of "zlev", the unit is [m], and in the case of "original", the grid numbers are specified. As default, 14 layers (1000 hPa, 975 hPa, 950 hPa, 925 hPa, 900 hPa, 850 hPa, 800 hPa, 700 hPa, 600 hPa, 500 hPa, 400 hPa, 300 hPa, 250 hPa, and 200 hPa) are given.

Example of configuration file: Vertical integration of 3D variable data

The description below presents an excerpt of a configuration file along with a simple analysis. The other item settings are the same as in the previous configuration.

```

&INFO
  ~ . . . ~
  Z_LEV_TYPE = "anal",
  ZCOUNT = 1,
/

&ANAL
  ANALYSIS = "sum",
/

&VARI
  VNAME = "QC", "QI", "QG",
/
```

If (Z_LEV_TYPE)="anal", simple analysis is applied to the 3D variable. This setting enables [ANAL]. The specification of (TARGET_ZLEV) in [VARI] is disabled, and (ZCOUNT) in [INFO] is necessarily given as "1" because of the output of 2D data.

- [ANAL]
 - (ANALYSIS) : The type of vertical simple analysis is specified. "max" and "min" represent the maximum and minimum value outputs in the vertical column, respectively, whereas "sum" and "ave" represent the vertical integration and the vertical average outputs, respectively. The default value is "ave".

Example of configuration file: Conversion of 2D variables

The example below shows a configuration in the case where 2D variables are converted. Because of the output of 2D data, (ZCOUNT) in [INFO] is necessarily given as "1."

```

&LOGOUT
LOG_BASENAME = "LOG_d01_2d",
/

&INFO
TIME_STARTDATE = 2000, 1, 1, 0, 0, 0,
START_TSTEP = 1,
END_TSTEP = 25,
DOMAIN_NUM = 1,
CONFFILE = "../run/run.d01.conf",
IDIR = "../run",
ZCOUNT = 1,
MAPPROJ_ctl = .true.
/

&VARI
VNAME = "T2","MSLP","PREC"
/

```

Example of configuration file change: Conversion of irregular data in time

The description below is an example of configuration change. The other item settings are the same as in the case of 2D or 3D variables.

```

&EXTRA
EXTRA_TINTERVAL = 600.0,
EXTRA_TUNIT = "SEC",
/

&VARI
VNAME = "RAIN",
/

```

In the case where (**TINTERVAL**) in [**HISTITEM**] and the output of several variables does not obey (**HISTORY_DEFAULT_TINTERVAL**) upon the execution of SCALE-RM in the configuration of output item (refer to Section 5.6), **net2g** supports such situations by setting [**EXTRA**]. This example shows the case where only 2D data "RAIN" are output at a time interval of 600 s, according to the configuration described in the last paragraph of Section 5.6. The history file that contains data with multiple time intervals is handled. Since **net2g** does not support the simultaneous treatment of different output intervals, it is necessary for **net2g** to be run separately.

Note for execution on supercomputer

When a simulation is conducted on a supercomputer such as the K Computer, many output files are generated, where the size of each file is large. In such cases, the local disk space is often not large enough to store them, and post-processing may take a long time. In this case, both a simulation by SCALE-RM and its post-processing on the same supercomputer are recommended. On the K Computer, **net2g** can be compiled by the "make" command if the environmental variable is appropriately set as described in Section ??.

5.10 How to restart run

The restart function is beneficial in limiting the job execution time of a computer system and in cases of unexpected termination in a long simulation. A run can usually be executed by being divided into multiple sequential runs using this function. In the restart run, it is also possible to

output restart files for the next restart simulation. The file outputs for restart are configured in `[PARAM_RESTART]` and `[PARAM_TIME]` in `run.conf`. The example below indicates that the restart run starts using the initial file `restart1_***`, and generates restart files called `restart2_***` every six hours.

```

&PARAM_RESTART
RESTART_RUN = .true.,           .true.: Restart run
                                .false.: Normal run (Default)

RESTART_IN_BASENAME = "restart1_d01_20070715-000000.000",
                                File name of input initial (restart) file.

RESTART_OUTPUT = .true.,       Setting of restart file output
                                .true.: Create restart files, .false.: Not create
RESTART_OUT_BASENAME = "restart2_d01",
                                Head of output restart file name
                                After the head name, the output date is added.

/

&PARAM_TIME
TIME_STARTDATE = 2007, 7, 15, 00, 0, 0, Start date of restart run
TIME_STARTMS = 0.DO,             Start date [ms]
TIME_DURATION = 12.0D0,          Integration time [Unit is defined by
                                TIME_DURATION_UNIT]
TIME_DURATION_UNIT = "HOUR",     Unit for TIME_DURATION
..... *snip* .....
TIME_DT_ATMOS_RESTART = 21600.DO, 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.DO, 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.DO, 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.DO, Output interval of restart files for urban variables
TIME_DT_URBAN_RESTART_UNIT = "SEC", Unit for TIME_DT_URBAN_RESTART

/

```

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 generated at the end of the simulation. The normal run is performed by specifying `(RESTART_RUN) = .false.` and `(RESTART_IN_BASENAME) = init_***`, which are prepared by `scale-rm_init`. The restart run is performed by specifying `(RESTART_RUN) = .true.` and assigning the restart file name to `(RESTART_IN_BASENAME)`.

The restart run is independent of the calculation that generates a restart file for the run. Note that the output variables in the restart file must be sufficient for the next restart run because the prognostic variables depend on the scheme used, especially in case of physical schemes. The simplest way for preparing the sufficient variables in the restart file is to use the same dynamic and physics schemes, that would be used in the restart simulation, in the simulation creating the restart file. `(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",
  ATMOS_BOUNDARY_IN_BASENAME = "../init/output/boundary_d01",
  ATMOS_BOUNDARY_START_DATE = 2010, 7, 14, 18, 0, 0,
  ATMOS_BOUNDARY_UPDATE_DT = 21600.D0,
/

```

"REAL": Real case experiment
Head of file name of boundary data
Initial date of boundary file
Time interval of boundary data

The boundary data at appropriate date are read in a restart simulation by specifying the first date of the boundary data in `boundary_***.nc` at (`ATMOS_BOUNDARY_START_DATE`) in [`PARAM_ATMOS_BOUNDARY`]. When (`ATMOS_BOUNDARY_START_DATE`) is not given, SCALE-RM treats the first data in `boundary_***.nc` as the boundary condition at the initial date of the restart simulation.

5.11 How to conduct nesting experiment

“Nesting” is the method for settings multiple computational domains overlapping one another in a nested structure. Figure 5.5 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 v5.1.5.

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 (DX_{d01}/DX_{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.

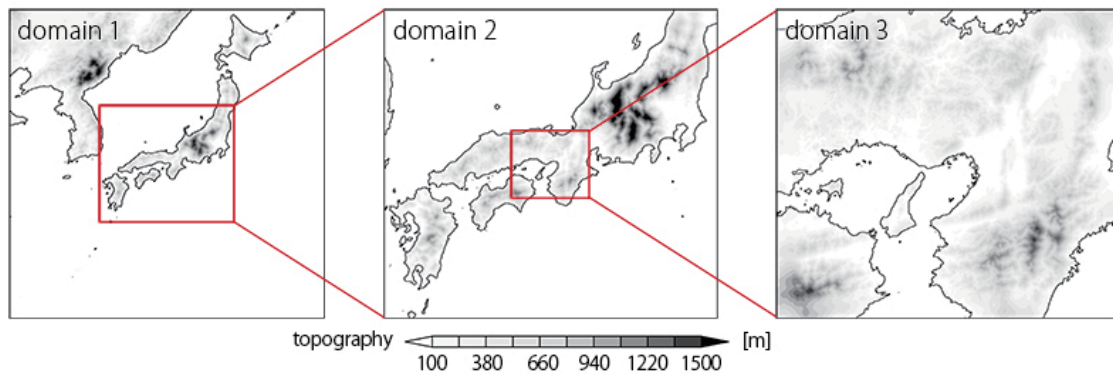


Figure 5.5: An example of domain nesting targeting the western area of Japan. Domain 1 is the outermost domain and domain 3 the innermost. The red rectangle and lines represents the domain 's location and its relations with other domains. The grid spacings are 7.5 km, 2.5 km, and 0.5 km for Domains 1, 2, and 3, respectively.

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

5.11.1 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 5.1.4), 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 topography-copy” function in SCALE-RM is recommended. This function copies the topography of the parent 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 5.6. To increase the resolution of topography 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 inner calculation 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 5.13), 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

Generate the topography of the parent domain (`scale-rm_pp`) first, and output a catalog file that gives the size of the parent domain to the child domain. The following configuration is needed in `pp.d01.conf` to output a catalog file:

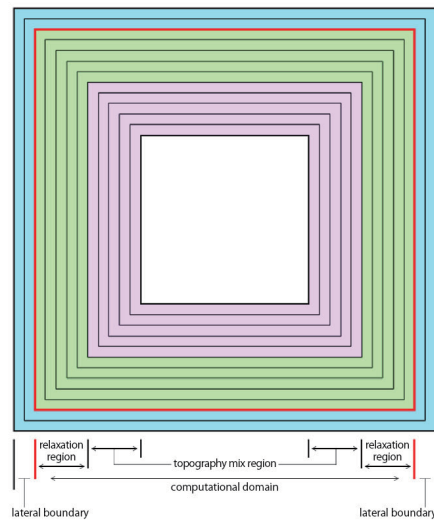


Figure 5.6: Horizontal distribution of topography when the topography-copy function is applied. The outermost grids shaded in light blue represent the HALO region, and the number of grids depends on the horizontal advection scheme. These grids are the lateral boundary. The area demarcated by the red 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_DOMAIN_CATALOGUE
  DOMAIN_CATALOGUE_FNAME = "latlon_domain_catalogue.d01.txt",  Name of catalog file
  DOMAIN_CATALOGUE_OUTPUT = .true.,                          Whether catalog file is out-
                                                                put
/

```

The other parameters are the same as usual.

To use the parent topography for the topography-topo 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`.

```

&PARAM_NEST
  USE_NESTING = .true.,
  OFFLINE = .true.,
  OFFLINE_PARENT_PRC_NUM_X = 2,
  OFFLINE_PARENT_PRC_NUM_Y = 2,
  OFFLINE_PARENT_KMAX = 36,
  OFFLINE_PARENT_IMAX = 45,
  OFFLINE_PARENT_JMAX = 45,
  OFFLINE_PARENT_LKMAX = 5,
  LATLON_CATALOGUE_FNAME = "latlon_domain_catalogue.d01.txt",
/

&PARAM_CNVTPOPO
  ~ . . . . ~
  CNVTPOPO_copy_parent = .true.,
/

&PARAM_COPYTOPO
  COPYTOPO_IN_BASENAME = "topo_d01",
  COPYTOPO_ENTIRE_REGION = .false.,
  COPYTOPO_LINEAR_H = .true.,
/

```

PRC_NUM_X of the parent domain
PRC_NUM_Y of the parent domain
KMAX of the parent domain
IMAX of the parent domain
JMAX of the parent domain
LKMAX of the parent domain
catalog file for the parent domain

whether the topography-copy function is applied

header name of parent topography data
whether the parent's topology is copied over the entire child domain

If (`CNVTPOPO_copy_parent`) in [`PARAM_CNVTPOPO`] 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 using which the topography transition method is applied. If (`COPYTOPO_LINEAR_H`) = `.true.`, the mixing ratio of the parent's topography to the child's topography is linearly changed. Otherwise, it is changed exponentially.

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

```

5.11.2 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 output of the parent domain.
3. Using the generated initial and boundary conditions, the temporal integration of the child domain is conducted.

An explanation is provided according to the above workflow.

Time integration of the parent domain

Below explanation assumes that the following procedure for the parent domain has already been completed: 1) Edit files `pp.***.conf`, `init.***.conf`, and `run.***.conf`. 2) Generate the topography and land-use data (`scale-rm_pp`). 3) Generate the initial and boundary data (`scale-rm_init`). The configuration files can be generated by renaming the sample files `/${Tutorial_dir}/real/sample/USER.offline-nesting-parent.sh` by `USER.sh` and using “the making tool for the complete settings of the experiment.”

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 of `run.***.conf`:

- All variables required for the computation of the child domain are have already been generated with the computation of the parent domain as history output.
- The interval of the history output is sufficiently short.
- The history in the parent domain is output at the model levels.
- The “catalog file,” which gives information pertaining to the parent ’s calculation domain to the child domain, is output.
- 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.

These configuration are found in the following in `run.d01.conf`. The part in blue corresponds to the above cautionary note:

```

&PARAM_DOMAIN_CATALOGUE
  DOMAIN_CATALOGUE_FNAME = "latlon_domain_catalogue_d01.txt",  Name of catalog file
  DOMAIN_CATALOGUE_OUTPUT = .true.,                          Output the catalog file
/

&PARAM_HISTORY
  HISTORY_DEFAULT_BASENAME = "history",
  HISTORY_DEFAULT_TINTERVAL = 900.DO,                        Time interval of history
                                                                data output
  HISTORY_DEFAULT_TUNIT = "SEC",                             Unit                of
                                                                HISTORY_DEFAULT_TINTERVAL

  HISTORY_DEFAULT_TAVERAGE = .false.,
  HISTORY_DEFAULT_DATATYPE = "REAL4",
  HISTORY_DEFAULT_ZCOORD = "model",                          Output the data at model
                                                                surface
  HISTORY_OUTPUT_STEPO = .true.,                              Include the output data at
                                                                t = 0
/

```

If the output option of the catalog file is `.true.`, the file `latlon_domain_catalogue_d01.txt` is output. In the case “the making tool for the complete settings of the experiment” is used, the file with the same name is generated in the directory `pp`. In this file, the latitudes and longitudes of the four corners of the regions where each MPI manages the parent domain calculation are described. (`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 5.6 for details of the items in [PARAM_HISTORY].

All necessary variables for the generation of initial and boundary data for the child domain must be described in [HISTITEM] in the file `run.d01.conf`. The necessary variables for offline nesting are shown below. Once this configuration is complete, conduct the time integration of the parent domain by `scale-rm`.

```
T2, Q2, MSLP, DENS, MOMZ, MOMX, MOMY, RHOT, QV  
LAND_SFC_TEMP, URBAN_SFC_TEMP, OCEAN_SFC_TEMP  
OCEAN_ALB_LW, OCEAN_ALB_SW, LAND_ALB_LW, LAND_ALB_SW  
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
```

Generation of initial and boundary data for child domain using output file of parent domain

Below explanation assumes that the following procedure for the child domain has already been completed: 1) Edit files `pp.***.conf`, `init.***.conf`, and `run.***.conf`. 2) Generate the topography and land-use data (`scale-rm_pp`). The configuration files for the child domain can be generated by renaming the sample files `${Tutorial_dir}/real/sample/USER.offline-nesting-child.sh` by `USER.sh` and using “the making tool for the complete settings of the experiment.”

We form the initial and boundary data using the history data generated by the calculation of the parent domain. The execution program typically used is `scale-rm_init`. `init.d02.conf` is configured as follows:

```

&PARAM_NEST
USE_NESTING = .true.,
OFFLINE = .true.,
OFFLINE_PARENT_PRC_NUM_X = 2,

OFFLINE_PARENT_PRC_NUM_Y = 2,

OFFLINE_PARENT_KMAX = 36,

OFFLINE_PARENT_IMAX = 45,

OFFLINE_PARENT_JMAX = 45,

OFFLINE_PARENT_LKMAX = 5,

LATLON_CATALOGUE_FNAME = "latlon_domain_catalogue_d01.txt",

/

&PARAM_MKINIT_REAL_ATMOS
NUMBER_OF_TSTEPS = 25,

FILETYPE_ORG = "SCALE-RM",
BASENAME_ORG = "history_d01",

BASENAME_BOUNDARY = "boundary_d01",
BOUNDARY_UPDATE_DT = 900.DO,

/

&PARAM_MKINIT_REAL_OCEAN
NUMBER_OF_TSTEPS = 25,

BASENAME_ORG = "history_d01",

FILETYPE_ORG = "SCALE-RM",
/

&PARAM_MKINIT_REAL_LAND
NUMBER_OF_TSTEPS = 25,

BASENAME_ORG = "history_d01",

FILETYPE_ORG = "SCALE-RM",
/

```

PRC_NUM_X in the file
 run.d01.conf
 PRC_NUM_Y in the file
 run.d01.conf
 KMAX in the file
 run.d01.conf
 IMAX in the file
 run.d01.conf
 JMAX in the file
 run.d01.conf
 LKMAX in the file
 run.d01.conf
 catalog file generated at the
 parent domain process

number of time steps in the
 history file
 HISTORY_DEFAULT_BASENAME
 in the file run.d01.conf
 time interval of history out-
 put (unit:"SEC")

number of time steps in the
 history file
 HISTORY_DEFAULT_BASENAME
 in the file run.d01.conf

number of time steps in the
 history file
 HISTORY_DEFAULT_BASENAME
 in the file run.d01.conf

In the case of generation of initial and boundary data from the output of SCALE-RM format, (**FILETYPE_ORG**) is specified as "SCALE-RM". (**BOUNDARY_UPDATE_DT**) is set to the same value of (**HISTORY_DEFAULT_TINTERVAL**) as in the configuration file in the parent domain (**run.d01.conf**). The items in [**PARAM_NEST**] are prepared for the nesting experiment. In offline nesting, (**USE_NESTING**) = **.true.**, (**OFFLINE**) = **.true.**. The six items beginning from characters (**OFFLINE_PARENT_*****) are intended for the configuration of the parent domain. Configure them correctly by referring to the configuration file for the parent domain (**run.d01.conf**). 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 following message, it means that the child domain is not completely contained in the parent domain:

```
xxx ERROR: REQUESTED DOMAIN IS TOO MUCH BROAD
xxx -- LONGITUDINAL direction over the limit
```

Time integration of child domain using initial and boundary data generated

Following the generation of the initial and boundary conditions, compute the child domain by `scale-rm`. This is same as in the usual atmospheric experiment. Before this, confirm that (`ATMOS_BOUNDARY_UPDATE_DT`) of [`PARAM_ATMOS_BOUNDARY`] in the file `run.d02.conf` is the same as the time interval of the output of history data for the parent domain. Note that the calculation continues without any message with regard to this inconsistency in the current version.

```
&PARAM_ATMOS_BOUNDARY
  ~      . . .      ~
  ATMOS_BOUNDARY_UPDATE_DT = 900.DO,
/
```

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.

5.11.3 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. 5.7, 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.

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 making tool for the complete settings of the experiment” (refer to Section 5.13). 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 5.11.1.

Editing the configuration file

In the configuration files `run.***.conf` for the parent and child domains, some nesting settings are added to [`PARAM_NEST`]:

— Configuration in `run.d01.conf` —

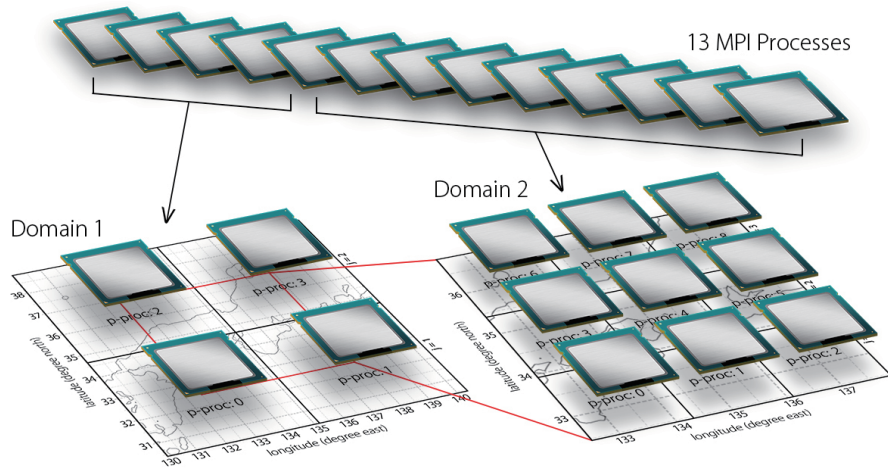


Figure 5.7: MPI process distribution in the online nesting experiment. In this example, 13 processes were launched at the beginning. These processes were distributed appropriately; 4-MPI parallel for 2×2 in Domain 1 and 9-MPI parallel for 3×3 in Domain 2 were executed. MPI communication flowed from Domain 1 to Domain 2.

```

&PARAM_NEST
USE_NESTING = .true.,           .true. at the online nesting
OFFLINE = .false.,            .false. at the online nesting
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 = .true.,
ONLINE_AGGRESSIVE_COMM = .true.,
/

```

— Configuration in run.d02.conf —

```

&PARAM_NEST
USE_NESTING = .true.,           .true. at online nesting
OFFLINE = .false.,            .false. at online nesting
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 = .true.,
ONLINE_AGGRESSIVE_COMM = .true.,
/

```

The experiment is identified as online nesting by `(USE_NESTING) = .true.` and `(OFFLINE) = .false.` The configuration items starting from `(ONLINE_**)` are particular to the online nesting experiment. `(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 N th domain is received from the parent with the domain number of $N-1$. Table 5.16 gives the configuration for an N -domain nesting experiment. 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 items are `.true.`

Table 5.16: 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 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 nesting experiment, water condensation calculated in the parent domain can be used for the boundary condition of the child domain because there is often no difference in physical schemes between the two domains. The delay in the generation of clouds and rain, as influence on the target domain, is expected to be suppressed by this remedy.

Configuration of launch file

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)
/

```

`(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`.

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 the error message indicates “SW search,” the western or southern area of the child domain is beyond the parent domain. If the error message indicates “NE search,” the eastern or northern area of the child domain is beyond 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:

```
xxx region of daughter domain is larger than that of parent: SW search
```


xxx region of daughter domain is larger than that of parent: NE search

Guideline for distribution of MPI processes

As shown in Fig. 5.7, 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 such the magnitude of time integrations for each process is as similar in value as is possible among processes^{*4)}. 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 number of grids in the x, y, and z directions in n-th domain are denoted by $IMAX_n$, $JMAX_n$, 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

$$OPR_n = IMAX_n \times JMAX_n \times KMAX_n \times TSTEP_n$$

The standard number of processes allocated to the n-th domain is estimated as

$$MPI_total \times \frac{OPR_n}{\sum_{m=1}^N OPR_m}, \quad (5.9)$$

where MPI_total is the total number of MPI processes.

The number of processes distributed along the x and y directions (**PRC_NUM_X**, **PRC_NUM_Y**) can be arbitrarily decided. It is recommended to configure them so that the difference between $IMAX$ and $JMAX$ 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^{*5)}.

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, the calculation in the innermost domain is largest in general. 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.

5.12 How to run bulk job

SCALE-RM has a function for bulk jobs that allows multiple independent jobs be handled simultaneously. 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 5.11.1).

^{*4)}More accurately, floating-point operations should be estimated.

^{*5)}Note that in case of hybrid parallelization used together with thread parallelism, e.g., in the K computer, it is necessary to take a larger number of grids along the y-axis than the x-axis to minimize computational imbalance between threads.

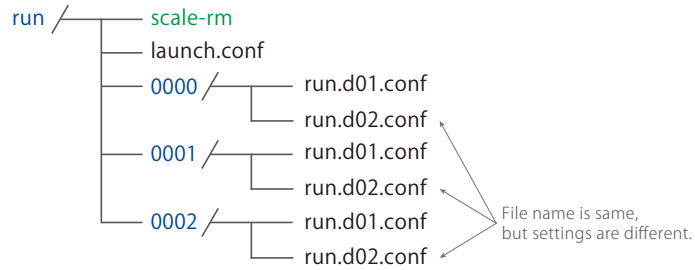


Figure 5.8: Structure of directories at the bulk job execution of SCALE-RM . Numbers such as “0000” are the directory names corresponding to job number, and called the job directory. All necessary configuration files must be prepared in each job directory.

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 is imaged as three sub-jobs with different integration periods and centers of calculation domain. (**NUM_DOMAIN**, **PRC_DOMAINS**, **CONF_FILES**) in [**PARAM_LAUNCHER**] in the file **launch.conf** (refer to Section 5.11.3) 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.

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 prepared. Below is an example of such a case.

```

&PARAM_LAUNCHER
  NUM_BULKJOB = 3,                number of sub-jobs
  NUM_DOMAIN = 2,                number of nesting domains
  PRC_DOMAINS = 9, 36,
  CONF_FILES = run.d01.conf, run.d02.conf,
/
  
```

It is sufficient to add item (**NUM_BULKJOB**) to the **launch.conf**. The other configurations are similar to those in Section 5.11.3. In case of a single-domain experiment (no nesting), specify (**NUM_DOMAIN**) = 1 and assign a file name to (**CONF_FILES**).

Prior to the execution of bulk jobs, prepare as many directories as the number of jobs, which are called “sub-directories.” As in Fig. 5.8, 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 than the path to directoies 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_TOPO
  TOPO_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_HISTORY
  HISTORY_DEFAULT_BASENAME = "0000/history_d01",
  ~ ... ~
/

```

As shown in Fig. 5.8, 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 job directory settings, the output is written to the same file, and the data disappear as a result.**

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 MPI
*** 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` 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 this example is the log file for Domain 1, `total process [LOCAL]` is correctly described 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 the execution of the sub-jobs. [Note that in the current version, all jobs are forcibly terminated if a job is abnormally finished.](#)

5.13 Supporting tool for preparation of configuration files

In the current version, the configuration files `***.conf` for the experiment are individually prepared for `pp`, `init`, `run`. Some items are required to be the same among these configure 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 script of USER.sh

```

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

NUMB_DOMAIN specifies the number of domains appearing in the script. Change the items below appropriately; as many items following "`# require parameters for each domain`" are described as the number of domains. Note that if the number of nesting domains according to NUM_DOMAIN and the number of variables 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`.

Bibliography

- A. C. M. Beljaars and A. A. M. Holtslag. Flux parameterization over land surfaces for atmospheric models. *J. Appl. Meteor.*, 30:327341, 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- Team SCALE. *The detail formulation of SCALE-RM*. RIKEN/AICS, 2015.
- H. Tomita. New microphysical schemes with five and six categories by diagnostic generation of cloud ice (special issue; the international workshop on high-resolution and cloud modeling, 2006). *Journal of the Meteorological Society of Japan. Ser. II*, 86:121–142, 2008.
- 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.

Appendix A

Namelist of configuration file and variable name list for history output

Namelist in configuration file

SCALE-RM Document page

NAMELIST Parameters

<http://scale.aics.riken.jp/doc/5.1.5/namelist.html>

List of variable names for history output

SCALE-RM Document page

History Variables

<http://scale.aics.riken.jp/doc/5.1.5/history.html>

Appendix B

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 MPIs parallel 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. 4. (Reference : Section 5.1.1 and 5.1.2).
- 2. How is the computational domain altered while maintaining the number of MPIs parallel?**
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. 4 (Reference : Section 5.1.2).
- 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. 4 (Reference : Section 5.1.2, 5.1.3, 5.1.4, and 5.5).
- 4. How is the location of computational domain altered?**
Change the domain center from the default value to a longitude of $135^{\circ}45.4'$ and a latitude of $35^{\circ}41.3'$ while maintaining the domain size for the tutorial case of the real atmospheric experiment in Chap. 4 (Reference : Section 5.2).
- 5. How is the integration time changed?**
Change the integration period from six hours to 12 hours for the tutorial case of the real atmospheric experiment in Chap. 4 (Reference : Section 5.5).
- 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. 4. (Reference : Section 5.6, Appendix A).
- 7. How does the model restart?**
Using the tutorial of the real atmospheric experiment in Chap. 4, execute three-hour integration. Then execute another 3 hours of integration using the restart files created in the first integration. (Reference : Section 5.10 and 5.4).

Answer

1. **How is the number of MPIs parallel changed while maintaining the computational domain?**

(*IMAX*, *JMAX*) in [*PARAM_INDEX*] and (*PRC_NUM_X*, *PRC_NUM_Y*) in [*PARAM_PRC*] are changed. If the below conditions are satisfied, your answer is correct:

$$\text{Number of MPIs parallel} = (\text{PRC_NUM_X}) \times (\text{PRC_NUM_Y}) = 6$$

$$\text{Number of grid cells along the X direction} = (\text{IMAX} \times \text{PRC_NUM_X}) = 90$$

$$\text{Number of grid cells along the Y direction} = (\text{JMAX} \times \text{PRC_NUM_Y}) = 90$$

2. **How is the computational domain altered while maintaining the number of MPIs parallel?**

If the number of grid cells per MPI process increases *n* times, the total number of grid cells also increases *n* times. Thus, only (*IMAX*, *JMAX*) in [*PARAM_INDEX*] are changed. The red parts indicate the answer below:

```
&PARAM_INDEX
KMAX = 36,
IMAX = 60,      (IMAX = 45 in the original settings)
JMAX = 30,      (JMAX = 45 in the original settings)
/
```

3. **How is horizontal grid interval changed while maintaining the computational domain?**

When the number of MPIs parallel is not changed, (*DX*, *DY*) in [*PARAM_GRID*] and (*IMAX*, *JMAX*) in [*PARAM_INDEX*] should be changed.

```
&PARAM_PRC
PRC_NUM_X = 2,
PRC_NUM_Y = 2,

&PARAM_INDEX
KMAX = 36,
IMAX = 180,     (IMAX = 45 in the original settings)
JMAX = 180,     (JMAX = 45 in the original settings)
/

&PARAM_GRID
DX = 5000.DO,   (DX = 20000.DO in the original settings)
DY = 5000.DO,   (DY = 20000.DO in the original settings)
/
```

In case the number of MPIs parallel are also changed, if the below conditions are satisfied in addition to &PARAM_GRID, your answer is correct:

$$\text{Number of grids along the X direction} = (\text{IMAX} \times \text{PRC_NUM_X}) = 360 \quad (\text{B.1})$$

$$\text{Number of grids along the Y direction} = (\text{JMAX} \times \text{PRC_NUM_Y}) = 360$$

Moreover, it is necessary to suitably change the time interval for the dynamics, i.e., (*TIME_DT_ATMOS_DYN*) and (*ATMOS_DYN_TINTEG_SHORT_TYPE*) (refer to Section 5.5). 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
BUFFER_DX = 100000.DO, (BUFFER_DX = 400000.DO, in the original settings)
BUFFER_DY = 100000.DO, (BUFFER_DY = 400000.DO, 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_MAPPROJ
MPRJ_basepoint_lon = 139.7567D0, (135.220404D0, in the original setting)
MPRJ_basepoint_lat = 35.6883D0, (34.653396D0, in the original setting)
MPRJ_type = 'LC',
MPRJ_LC_lat1 = 30.00D0,
MPRJ_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 5.4, `(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 `(HISTORY_DEFAULT_TINTERVAL)` in `[PARAM_HISTORY]` as below. The output variables are specified at `(ITEM)` in `[HISTITEM]`. Refer to Appendix A for variable names.

```
&PARAM_HISTORY
HISTORY_DEFAULT_BASENAME = "history_d01",
HISTORY_DEFAULT_TINTERVAL = 1800.D0, (3600.D0, in the original settings)
HISTORY_DEFAULT_TUNIT = "SEC",
/

&HISTITEM item="SFLX_SW_up" / added
&HISTITEM item="SFLX_SW_dn" / added
```

7. How does the model restart?

The first integration for three hours is configured in `run.conf` as below. Once the three hours for integration have elapsed, the restart file is created.

```

&PARAM_TIME
TIME_STARTDATE = 2007, 7, 14, 18, 0, 0,
TIME_STARTMS = 0.DO,
TIME_DURATION = 3.0DO,          necessary for more than 3 hours
TIME_DURATION_UNIT = "HOUR",
....(省略)....
TIME_DT_ATMOS_RESTART = 10800.DO,
TIME_DT_ATMOS_RESTART_UNIT = "SEC",
TIME_DT_OCEAN_RESTART = 10800.DO,
TIME_DT_OCEAN_RESTART_UNIT = "SEC",
TIME_DT_LAND_RESTART = 10800.DO,
TIME_DT_LAND_RESTART_UNIT = "SEC",
TIME_DT_URBAN_RESTART = 10800.DO,
TIME_DT_URBAN_RESTART_UNIT = "SEC",
/

&PARAM_RESTART
RESTART_RUN = .false.,          (added, but it is not necessary.)
RESTART_OUTPUT = .true.,        (.false., in the original setting)
RESTART_IN_BASENAME = "../init/init_d01_20070714-180000.000",
RESTART_OUT_BASENAME = "restart_d01",          (added)
/

&PARAM_ATMOS_BOUNDARY
ATMOS_BOUNDARY_TYPE = "REAL",
ATMOS_BOUNDARY_IN_BASENAME = "../init/output/boundary_d01",
ATMOS_BOUNDARY_START_DATE = 2010, 7, 14, 18, 0, 0,          (added, but is not necessary.)
ATMOS_BOUNDARY_UPDATE_DT = 21600.DO,
/

```

When `(TIME_DURATION)` is set to three 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 three 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 three hours (10800 s) and a multiple of `(TIME_DT)`.

The configuration for the restart calculation from three to six hours in integration time is as follows:

```

&PARAM_TIME
  TIME_STARTDATE = 2007, 7, 14, 21, 0, 0,
  TIME_STARTMS = 0.DO,
  TIME_DURATION = 3.0D0,
  TIME_DURATION_UNIT = "HOUR",
/

&PARAM_RESTART
  RESTART_RUN = .true.,
  RESTART_OUTPUT = .true.,
  RESTART_IN_BASENAME = "restart_d01_20070714-210000.000",
  RESTART_OUT_BASENAME = "restart2_d01",
/

&PARAM_ATMOS_BOUNDARY
  ATMOS_BOUNDARY_TYPE = "REAL",
  ATMOS_BOUNDARY_IN_BASENAME = "../init/output/boundary_d01",
  ATMOS_BOUNDARY_START_DATE = 2010, 7, 14, 18, 0, 0,
  ATMOS_BOUNDARY_UPDATE_DT = 21600.DO,
/

```

set for longer than 3 hours

(necessary)
(not always necessary)
(necessary)
(not always necessary)

(necessary)

SCALE-RM USERS GUIDE

Author & Editor

Team SCALE User's guide working group

If you have any questions or concerns,
please contact us (scale@ml.riken.jp) via e-mail.

Copyright ©Team SCALE, RIKEN AICS, 2016. All rights reserved.