

SCALE-DG USERS GUIDE

Version 1.2.0

Yuta Kawai, Team SCALE

September 30, 2026



Contents

1	Overview	3
1.1	Introduction	4
1.1.1	What is FE-Project?	4
1.1.2	Structure of SCALE-DG	6
2	Install	8
2.1	Preparation	9
2.1.1	System environment	9
2.2	Compile of SCALE-DG	11
2.2.1	Download source codes and Setting computational environment	11
2.2.2	Compile	12
2.3	Compile of post-processing tool	13
2.3.1	Compilation of programs with post-processing	13
3	Tutorial for SCALE-DG	14
3.1	Operation check and basic usage	15
3.1.1	Introduction	15
3.1.2	How to execute model	15
3.1.3	Post-processing and visualization	17
3.1.4	Guideline for further studies	18
3.1.5	Available test cases	18
4	Various settings	20
4.1	Pre-processing	21
4.1.1	Setting the topography	21
4.1.2	How to prepare initial and boundary data	21
4.2	Framework for SCALE-DG	22
4.2.1	Configure a target domain	22
4.2.1.1	Set domain size and the number of horizontal and vertical elements	22
4.2.2	Setting integration period and time step	23
4.2.3	Setting history files and output variables	23

4.3	Setting dynamical core	26
4.3.1	Temporal scheme	26
4.3.2	Modal filtering	27
4.3.3	Setting for boundary condition	28
4.3.4	Setting for Coriolis force	28
4.3.4.1	Regional model	28
4.3.4.2	Global model	29
4.3.5	Setting for shallow-atmosphere approximation	29
4.4	Setting the physical process	30
4.4.1	Cloud Micro-Physics	30
4.4.1.1	Kessler scheme	31
4.4.1.2	Tomita08 scheme	31
4.4.1.3	Large-scale condensation scheme	31
4.4.2	Cumulus parameterization	32
4.4.2.1	Moist convective adjustment scheme	32
4.4.3	Turbulence	33
4.4.3.1	Smagorinsky–Lilly type scheme	33
4.4.3.2	Smagorinsky–Lilly type scheme for global model	34
4.4.3.3	Constant viscosity and diffusivity	34
4.4.4	Planetary Boundary Layer Scheme	35
4.4.4.1	MYNN level 2 scheme	35
4.4.5	Radiation scheme	36
4.4.5.1	Configuration of incoming solar radiation	36
4.4.5.2	Simplified radiation scheme	37
4.4.6	Surface flux	39
4.4.6.1	Constant surface-flux scheme	39
4.4.6.2	Simple bulk surface-flux scheme	39
4.4.7	Ocean model	41
4.4.7.1	Setting grids for ocean model	41
4.4.7.2	Ocean-model configurations	42
4.4.7.3	Surface fluxes and atmosphere–ocean exchange	42
4.5	Common settings	44
4.6	Post-processing	45
4.6.1	Regrid tool	45
4.6.1.1	Basic usage	45
4.6.1.2	Regrid data from model-level coordinate to other vertical coordinate	47
4.6.2	Spectral analysis tool	48
5	Detailed explanation for advanced use	50
5.1	How to use user defined program	51
6	Appendix	54

Part 1

Overview

Chapter 1.1

Introduction

This user's manual is intended for users of a regional and global atmospheric model SCALE-DG version 1.2.0. The structure of this document is as follows: Part 1 provides an overview of our model, Part 2 describes how to install SCALE-DG Chapters 3.1 and ?? explain the basic use of SCALE-DG using examples of an idealized experiment and an realistic atmospheric experiment, respectively. Since these chapters are constructed as a series of tutorials, it is recommended that beginning users of SCALE-DG read these chapters meticulously. Parts 4 and 5 describe how to change the model configuration and explain data format and available functions and tools. Since each section is closed itself basically, these chapter can be utilized as a dictionary.

1.1.1 What is FE-Project?

FE-project provides a library and sample programs for the discontinuous Galerkin methods (DGM). Furthermore, we provide an atmospheric model with a regional/global dynamical core based on DGM, SCALE-DG. In FE-Project, we use scalable Computing for Advanced Library and Environment (SCALE) (<https://scale.riken.jp/>), which is a basic software library of weather and climate models of the earth and planets intended for widespread use.

Our project has the following advantages:

- FE-Project provides an open-source software under the “MIT license”. It is free for use, modification, and redistribution, regardless of whether the user is a business or other enterprise.
- FE-Project contains a regional and global atmospheric model called the SCALE-DG (SCALE regional/global model with discontinuous Galerkin method).
- In SCALE-DG, several physics processes are coupled to DG dynamical core using a framework for physical processes provided by SCALE library.

For the details of the license, the interested reader can refer to the file **FE-Project-1.2.0/LICENSE** under the main directory. The general references of SCALE-DG are [Kawai and Tomita \(2023, 2025\)](#). If SCALE-DG is used in your studies, please cite two our papers in addition to the reference papers of SCALE library ([Nishizawa et al., 2015](#); [Sato et al., 2015](#)).

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

Relations between FE-Project (SCALE-DG) and SCALE

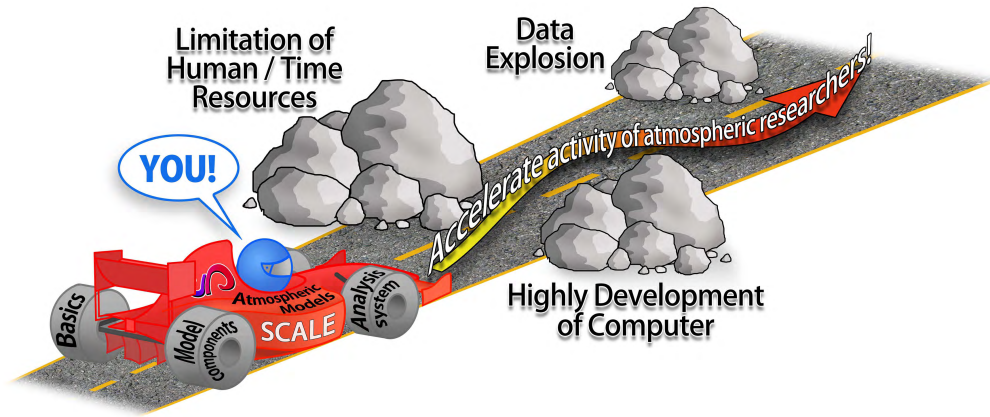


Figure 1.1.1: Aims of SCALE (cite: Fig. 1.1.1 of SCALE users guide)

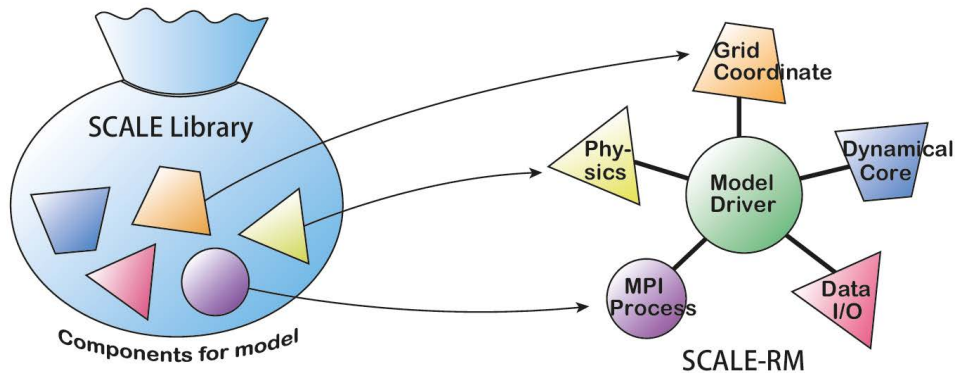


Figure 1.1.2: Relationship between the library SCALE and the model SCALE-RM (cite: Fig. 1.1.2 of SCALE users guide)

SCALE has been developed in RIKEN R-CCS with several outside contributors, and its improvement and extension continue. Figure 1.1.1 shows the schematic concept of SCALE. As shown in this figure, SCALE aims to resolve various problems. The development of SCALE is considered in the context of its wide use by devices ranging from small PC clusters to next-generation supercomputers. For this purpose, scientists in meteorology/climate science and computer science are cooperating. As shown in Fig. 1.1.2, SCALE manages the parallel processes, file I/O, and inner-communication. SCALE also provides the solver for atmospheric flow (dynamical core based on FVM with Arakawa C-grid) and physical processes such as micro-physics and radiation processes. SCALE-RM is a regional atmospheric model that fully uses SCALE.

Figure 1.1.3 represents the relationship between the library SCALE and FE-Project. FELib provides Fortran modules for DGM which manage finite element, data communication, various temporal schemes, and many simple examples. In FELib, SCALE library is used for file IO, error handling, and physical processes, etc. SCALE-DG is constructed by combining functions provided

by FELib. SCALE-DG itself reads the input data of atmospheric status as prognostic variable, and conducts time-integration. Users can select a scheme in every component according to simulations they want.

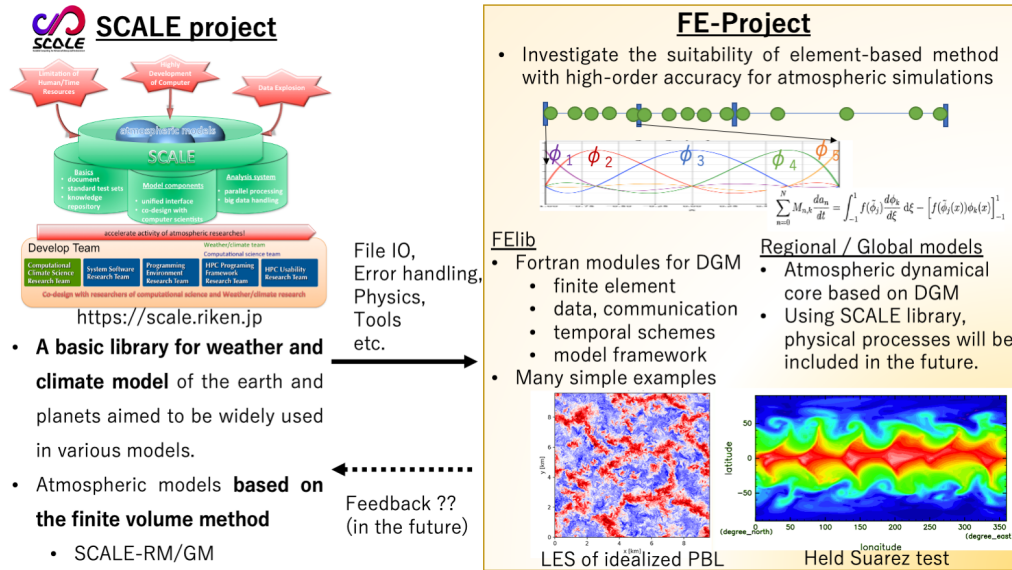


Figure 1.1.3: Relationship between the library SCALE and FE-Project including SCALE-DG

1.1.2 Structure of SCALE-DG

Several physics in SCALE library are available in SCALE-DG. The components are categorized into three parts: framework, dynamical core, and physical processes.

Framework

- The three-dimensional (3D) Cartesian and cubed-sphere grid systems for regional and global models
- 2D domain decomposition by Message Passing Interface (MPI) communication
- Collective execution system of multiple cases, i.e., bulk job system
- netCDF file I/O based on CF (Climate and Forecast) convention^{*1}
- Generation of initial data for idealized experiments

Dynamical core

- Governing equations: 3D fully compressible non-hydrostatic equations
- Spatial discretization: nodal discontinuous Galerkin method (Hesthaven and Warburton, 2007)
 - Polynomial order associated with numerical accuracy can be arbitrarily chosen
 - Numerical flux: Rusanov flux (Rusanov, 1961)

^{*1}<http://cfconventions.org/>

- Stabilization mechanisms: Inherent numerical dissipation with numerical fluxes and explicit modal filtering
- Temporal discretization: various type of Runge–Kutta (RK) schemes
 - Horizontally and vertically explicit (HEVE) strategy with full explicit RK schemes
 - Horizontally explicit and vertically implicit (HEVI) strategy with IMEX RK schemes
- Guarantee of non-negative value for tracer advection:
 - Preserving the non-negativity is ensured by a limiter ([Light and Durran, 2016](#))
- Topography: Terrain-following coordinates

Physical processes

- Turbulence process: selectable from among the following
 - Smagorinsky (1963) & Lilly (1962)-type sub-grid scale turbulent model with the corrections by [Brown et al. \(1994\)](#) and [Scotti et al. \(1993\)](#)
 - MYNN level 2.0 boundary layer scheme [Nakanishi and Niino \(2006\)](#)
- Cloud microphysics: selectable from among the following
 - 3-class 1 moment bulk scheme ([Kessler, 1969](#))
 - 6-class 1 moment bulk scheme ([Tomita, 2008](#))
 - Large-scale condensation scheme
- Cumulus process
 - Moist convective adjustment ([Manabe et al., 1965](#))
- Radiation process: selectable from among the following
 - Gray radiation scheme ([Byrne and O’Gorman, 2013](#); [Frierson et al., 2006](#))
 - Two-band infrared and one-band solar radiation scheme ([Geen et al., 2016](#); [Vallis et al., 2018](#))
- Surface processes
 - Ocean model: selectable from among the following
 - * Fixed to initial condition
 - * Slab model
 - Simple surface fluxes with a constant heat flux or constant bulk coefficients

Part 2

Install

Chapter 2.1

Preparation

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

2.1.1 System environment

Required Softwares

- **OS** : Linux OS, Mac OS
- **Compiler** : C, Fortran

Since Fortran source codes of SCALE library and FE-Project are written in FORTRAN 2008 standard syntax, the compiler must support it. Refer to Table 2.1.1 for compilers already confirmed as supported.

When NVIDIA HPC SDK is used, OpenACC/GPU execution is available for selected sample programs and the HEVE dynamical-core calculation. This GPU support is still under development. Therefore, some SCALE-DG components, especially those related to physical processes, may still be executed on CPUs or may not yet support GPU execution.

Required libraries

The required external libraries are described below:

- netCDF Library (<http://www.unidata.ucar.edu/software/netcdf/>)
- MPI Library (e.g., openMPI <http://www.open-mpi.org/>)
- LAPACK (<http://www.netlib.org/lapack/>)
- SCALE library (<https://scale.riken.jp>)

To install several libraries, you can use distributed binary packages for Linux OS and Mac OS. For the MPI library, the MPI 1.0/2.0 protocol should be supported. Refer to Table 2.1.2 for MPI libraries already confirmed as supported.

Table 2.1.1: Compiler already checked

Name of compiler	
GNU (gcc/gfortran)	Version 11.4.0 or later is supported.
Intel HPC Toolkit (icc/ifx)	Version 2025.0 or later is supported.
Fujitsu on Fugaku (fccpx/frtpx)	tcsds-1.2.41 or later is supported.
NVIDIA HPC SDK (nvcc/nvfortran)	Version 26.1 or later is supported.

Table 2.1.2: MPI libraries already checked

Name of MPI library	
Open MPI	Version 4.1.2 or later is supported.
Intel MPI	Version 2021.14 or later is supported.

Visualization tools

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

The Python scripts with Xarray and Matplotlib are used for a quick view and the drawing model output in the tutorial in Chapter 3.1. Other tools are also available, if they can be read in netCDF file format, which is the output of SCALE-DG .

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

Chapter 2.2

Compile of SCALE-DG

2.2.1 Download source codes and Setting computational environment

Get source codes

The source codes for the latest release can be downloaded from https://ywkawai.github.io/FE-Project_web/download/.

The directory `FE-Project-1.2.0/` can be seen when the tarball file of the source code is extracted.

```
$ tar -zxvf FE-Project-1.2.0.tar.gz
$ ls ./FE-Project-1.2.0/
```

Set Makedef file and environmental variables

SCALE-DG is compiled using a Makedef file specified in the environment variable “`SCALE_FE_SYS`.” Several variations of the Makedef file are prepared in the directory `FE-Project-1.2.0/sysdep/`. According to your computational environment, choose a Makedef file in Table 2.2.1 and set the environment variable as

```
$ export SCALE_FE_SYS="Linux64-gnu-mpi" (for example)
```

If there is no Makedef file for your environment, create a Makedif file by modifying any existing one. Because FE-Project uses SCALE library, you need to set a directory including SCALE library as

```
$ export SCALE="~/workspace/scale-5.X.X/" (for example)
```

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

OS/Computer	Compiler	MPI	Makedef file
Linux OS x86-64	GNU	Open MPI	Makedef.Linux64-gnu-mpi
	Intel OneAPI	Intel MPI	Makedef.Linux64-llvm-intel-mpi
	NVIDIA HPC SDK	CUDA-aware Open MPI	Makedef.Linux64-nvidia
Mac OS	GNU	Open MPI	Makedef.MacOSX-gnu-mpi
Fugaku	Fujitsu	Fujitsu MPI	Makedef.FUGAKU

All the environment variables applied at compile time are listed in Table 2.2.2. For example, if you would like to enable a thread parallelization with OpenMP, set a variable as

```
$ export SCALE_ENABLE_OPENMP=T
```

Table 2.2.2: List of environment variables applied at compile time

Environment variable	Description
SCALE	Path of SCALE library (required)
SCALE_FE_SYS	Selection of system (required)
SCALE_ENABLE_OPENMP	Enable to use OpenMP
SCALE_DEBUG	Use compile option for debug
SCALE_QUICKDEBUG	Use compile option for quick debug (detect floating point error while maintaining speed-up option)
SCALE_ENABLE_OPENACC NVIDIA_GPU_COMPATIBILITY	Enable to use OpenACC Compute capability of the target NVIDIA GPU used by NVIDIA HPC SDK.
SCALE_NETCDF_INCLUDE	Include path of NetCDF library
SCALE_NETCDF_LIBS	Directory path of NetCDF library and specifying libraries
SCALE_MATHLIB_LIBS	Directory path of numerical library and specifying libraries

For file I/O, SCALE-DG requires netCDF . In the most cases, the pathes of netCDF may be automatically found by using “nc-config” command. If the paths cannot be found automatically, you should set the environmental variables for netCDF as follows:

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

2.2.2 Compile

Compile of SCALE-DG

To build SCALE-DG and conduct a test case such as idealized mountain wave tests, move to a directory of the test case and execute a command as

```
$ cd FE-Project-1.2.0/model/atm_nonhydro3d/test/case/mountain_wave/linear_hydrostatic_case/
$ make -j 4
```

The number of -j option shows a number of parallel compile processes and specify this number according to your environment. When a compilation is successful, the following three executable files are generated in the current directory.

```
scale-dg  scale-dg_init  scale-dg_pp
```

Note that we prepare various numerical experiments are prepared. The configuration files can be found in the directory of FE-Project-1.2.0/model/atm_nonhydro3d/test/case/*. It is useful for you to create a new experimental setting.

Points to note

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

```
$ make clean
```

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

```
$ make allclean
```

Chapter 2.3

Compile of post-processing tool

2.3.1 Compilation of programs with post-processing

regrid_tool is a post-processing tool for the SCALE-netCDF file generated by the SCALE library. The output file of SCALE-DG is divided and stored in every computational node. regrid_tool reads the output files (history file, `history.*****.nc`) and interpolate it to DG meshes with arbitrary the number of elements and polynomial order. The interpolated data is output to new files. The details of how to use is explained in Section [4.6.1](#).

regrid_tool uses `libScaleFECORE.a.a`, generated at the time of compilation of SCALE-DG . This library is located under the `FE-Project-1.2.0/lib` directory. To compile regrid_tool , execute the following command after the compilation of SCALE-DG :

```
$ cd FE-Project-1.2.0/model/atm_nonhydro3d/util/regrid_tool/  
$ make
```

If the compilation is succeeded, the executable binary file is generated under the `FE-Project-1.2.0/bin` directory.

The execution of regrid_tool is as follows:

```
$ mpirun -n 2 ./regrid_tool regrid.conf
```

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

Part 3

Tutorial for SCALE-DG

Chapter 3.1

Operation check and basic usage

3.1.1 Introduction

In this chapter, the basic operations of SCALE-DG are explained by conducting a test case of mountain wave. It is recommended that the users perform this tutorial because it includes a check for whether the compilation of SCALE in Part 2 has been completed. This chapter assumes that the following file has been already generated:

```
FE-Project-1.2.0/model/atm_nonhydro3d/test/case/mountain_wave/scale-rm
FE-Project-1.2.0/model/atm_nonhydro3d/test/case/mountain_wave/scale-rm_init
FE-Project-1.2.0/model/atm_nonhydro3d/test/case/mountain_wave/scale-rm_pp
FE-Project-1.2.0/bin/regrid_tool
```

Furthermore, python scripts are used as a visualization tool.

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

3.1.2 How to execute model

Experimental setup

This tutorial focuses on a test case of quasi two-dimensional mountain wave. Table 3.1.1 shows the experimental setting.

Table 3.1.1: Experimental setting of idealized experiment

	Configuration	Note
Number of MPI processes	8: x -direction, 1: y -direction	1 MPI parallelization
Horizontal DOF	40 elements $\times$ 8 DOF ($p = 7$): x -direction, 1 element $\times$ 8 DOF ($p = 7$): y -direction	
Vertical DOF	8 elements $\times$ 8 DOF ($p = 7$)	
Horizontal element size	2.5 km: x -direction, 2.5 km: y -direction	quasi 2D experiment in the x - and vertical directions.
Lateral boundary condition	Periodic condition	Both eastward and northward
Time step	12 sec	
Integration time	10 hour	The total number of time step is 3000.
Time interval of data output	1 hour	

Preparations

As described in Section 2.2.2, move to a directory of target test case and compile SCALE-DG .

```
$ cd FE-Project-{\version}/model/atm_nonhydro3d/test/case/mountain_wave/linear_hydrostatic_case
$ make -j4
```

Then, the following three executable files are generated in the current directory.

```
scale-dg  scale-dg_init  scale-dg_pp
```

Creating initial conditions

To create the initial conditions, the configuration file for `scale-dg_init` is required. The configuration file `init_R20kmDX500m.conf` has been prepared according to Table 3.1.1. Reading the configuration file, `scale-dg_init` calculates the stratified atmospheric structure and the initial disturbance.

The general form of the executable command in SCALE-DG 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-dg`, `scale-dg_init`, and so on. The configuration file, where the experimental settings are described, is given to [the configuration file]. If `init.conf` is used as the configuration file and `scale-dg_init` using eight-MPI parallel is executed, give the command as follows:

```
$ mpirun -n 8 ./scale-dg_init ./init.conf
```

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

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

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

```
init_LOG.pe000000
init_00000101-000000.000.pe000000[0-7].nc
```

where 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 eight MPI processes are used, only the log file of the 0th MPI rank (master rank) is output as default. If the execution is successfully finished, the statements below are output at the end of LOG file:

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

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

Execution of simulation

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

```
$ mpirun -n 8 ./scale_dg run.conf
```

The following files are then generated under the given directory:

```
LOG.pe000000
history.pe000000[0-7].nc
```

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

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

The history files `history.pe000000*.nc` contain the results of the calculation.

3.1.3 Post-processing and visualization

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

```
$ ln -s FE-Project-{version}/bin/regrid_tool ./
```

The method of execution of `regrid.tool` is the same as that of SCALE-DG, i.e.,

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

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

```
$ mpirun -n 8 ./regrid_tool ./regrid_topo.conf
$ mpirun -n 8 ./regrid_tool ./regrid.conf
```

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

```
*** End regrid_tool
```

To confirm whether the flow calculation succeeded, create a figure by the following command

```
$ bash visualize/visualize.sh
```

This shell script call a Python script for visualizing simulation results. If it is successfully completed, the following files are generated in the directory of analysis/:

```
U_t{0,7200,18000,36000}s.png
W_t{0,7200,18000,36000}s.png
```

Figure 3.1.1 shows the results of horizontal wind U and vertical wind W at $t = 10$ hours. If you obtain the same figures as Fig. 3.1.1, post-processing is successfully finished.

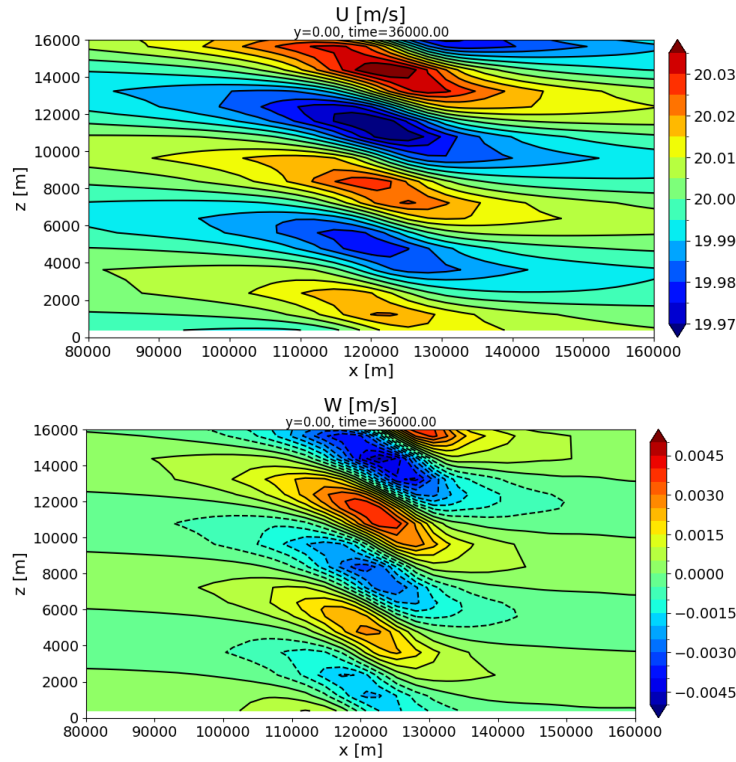


Figure 3.1.1: The vertical section of horizontal and vertical winds after $t = 10$ hours

3.1.4 Guideline for further studies

In this chapter, the strategy for the execution of SCALE-DG was explained by using a simple idealized test. We recommend studying procedures of changing the model resolution, the calculation domain, and the number of MPI processes for further study. With regard to the idealized experiment, several files of other configurations, e.g., to increase resolution, the number of domains, and the physical schemes, 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, as explained in Section 3.1.5, various idealized experimental settings have been prepared. 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 strategies for the configuration of physical processes, such as cloud microphysics, radiation, and turbulence schemes. They are described in Part 4.

3.1.5 Available test cases

Several test cases and sample experiments are prepared under `model/atm_nonhydro3d/test/case/`. These cases are useful not only for checking the installation and basic operation of SCALE-DG, but also as templates for user-defined experiments. Users are encouraged to copy one of these directories and modify the configuration files, initial conditions, physical processes, grid settings, and output settings according to their own purposes.

The prepared test cases cover regional and global configurations, dynamical-core test problems, tracer advection tests, turbulence experiments, and idealized planetary-atmosphere experiments. Examples are listed in Table 3.1.2.

A typical procedure is to move to a target directory and execute

Table 3.1.2: Examples of test cases prepared under `model/atm_nonhydro3d/test/case/`.

Directory	Description
<code>density_current</code>	Density current test for the regional dynamical core
<code>rising_thermal_bubble</code>	Rising thermal bubble test
<code>sound_wave</code>	Sound wave propagation test for the regional model
<code>inertia_gravity_wave</code>	Inertia-gravity wave test for the regional model
<code>mountain_wave</code>	Mountain wave test for the regional model
<code>baroclinic_wave</code>	Baroclinic wave test for the regional model
<code>tracer_advection</code>	Tracer advection test for the regional model
<code>isentropic_vortex</code>	Isentropic vortex test
<code>Kelvin_Helmholtz_wave</code>	Kelvin–Helmholtz instability test
<code>RB_convection</code>	Rayleigh–Bénard convection test
<code>pbl_turbulence</code>	Planetary boundary layer turbulence experiment
<code>sound_wave_global</code>	Sound wave test for the global model
<code>inertia_gravity_wave_global</code>	Inertia-gravity wave test for the global model
<code>mountain_wave_global</code>	Mountain wave test for the global model
<code>baroclinic_wave_global</code>	Baroclinic wave test for the global model
<code>tracer_advection_global</code>	Tracer advection test for the global model
<code>equatorial_wave_global</code>	Equatorial wave test for the global model
<code>Earth_like_atm</code>	Idealized Earth-like atmosphere experiments
<code>tidally_locked_planet_atm</code>	Idealized tidally locked planetary atmosphere experiments

```
$ make
$ make run
```

Some test cases also provide visualization scripts or additional sample configuration files. For example, if a visualization target is prepared in the Makefile, figures can be generated by

```
$ make vis
```

The settings in these test cases are not intended to cover all possible scientific applications. They should be regarded as starting points for constructing new experiments.

Part 4

Various settings

Chapter 4.1

Pre-processing

4.1.1 Setting the topography

SCALE-DG employs the terrain-following coordinates to represent topography. In these coordinates, the bottom face of the lowest grid is given such that it can follow the surface altitude, i.e., topography. Topography data used in the program `scale-dg_init` and `scale-dg` should be prepared in SCALE format beforehand, except for some idealized experiments.

4.1.2 How to prepare initial and boundary data

Settings common to NetCDF format

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

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

To generate initial file, `(OUTPUT)` is set to `.true..` The base name of initial file is specified by `(OUT_BASENAME)`. These settings are also used when outputting restart files in executing SCALE-DG (see Sec. ?? for details). The structure of generated restart files is same as that of initial files.

Chapter 4.2

Framework for SCALE-DG

4.2.1 Configure a target domain

4.2.1.1 Set domain size and the number of horizontal and vertical elements

For the regional model using the Cartesian coordinates, information with domain size and DG mesh are specified in `[PARAM_ATMOS]` and `[PARAM_ATMOS_MESH]` in the configuration files.

```
&PARAM_ATMOS
  ATMOS_MESH_TYPE = "REGIONAL", ; Mesh type
/
```

```
&PARAM_ATMOS_MESH
  NprcX = 2, ; Number of MPI process in the x-direction
  NeX = 2, ; Number of finite element in the x-direction in each MPI process
  NprcY = 2, ; Number of MPI process in the y-direction
  NeY = 2, ; Number of finite element in the y-direction in each MPI process
  NeZ = 4, ; Total number of finite element in the z-direction
  dom_xmin = 0D0, ; Minimum x-coordinate value [m]
  dom_xmax = 100D3, ; Maximum x-coordinate value [m]
  dom_ymin = 0D0, ; Minimum y-coordinate value [m]
  dom_ymax = 100D3, ; Maximum y-coordinate value [m]
  dom_zmin = 0D0, ; Minimum vertical coordinate value
  dom_zmax = 30D3, ; Maximum vertical coordinate value
  PolyOrder_h = 7, ; Polynomial order for the horizontal direction
  PolyOrder_v = 7, ; Polynomial order for the z-direction
  LumpedMassMatFlag = .false., ; Flag whether a mass lumping is applied
/
```

An arbitrary non-uniform element size can be specified in the vertical direction.

```
&PARAM_ATMOS_MESH
  FZ = 0.0D0, 2760D0, 10340.0D0, 19528.0D0, 30000D0, ; Location at element face in the vertical direction [m]
/
```

For global model using cubed-sphere coordinates, information with domain size and DG mesh are specified in `[PARAM_ATMOS]` and `[PARAM_ATMOS_MESH]` in the configuration files.

```
&PARAM_ATMOS
  ATMOS_MESH_TYPE = "GLOBAL", ; Mesh type
/
```

```
&PARAM_ATMOS_MESH
  Nprc = 6, ; Total number of MPI processes
  NeGX = 6, ; Total number of finite element in the x-direction
  NeGY = 6, ; Total number of finite element in the y-direction
  NeZ = 4, ; Total number of finite element in the z-direction
  dom_zmin = 0D0, ; Minimum vertical coordinate value
  dom_zmax = 30D3, ; Maximum vertical coordinate value
  PolyOrder_h = 7, ; Polynomial order for the horizontal direction
  PolyOrder_v = 7, ; Polynomial order for the z-direction
  LumpedMassMatFlag = .false., ; Flag whether a mass lumping is applied
/
```

We can set an arbitrary non-uniform element size in the vertical direction as mentioned above. The planetary radius is set in `[PARAM_CONST]` as

```
&PARAM_ATMOS_MESH
  CONST_RADIUS = 6.37122D6, ;
```

where the default value is the radius of Earth.

4.2.2 Setting integration period and time step

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

```
&PARAM_TIME
  TIME_STARTDATE = 2014, 8, 10, 0, 0, 0, ; Start date of integration: it is required for the cal-
                                          ; culation of the radiation process.
  TIME_STARTMS = 0.D0, ; Start date [mili sec]
  TIME_DURATION = 12.0D0, ; Integration time [init is defined by
                           ; TIME_DURATION_UNIT]
  TIME_DURATION_UNIT = "HOUR", ; Unit for TIME_DURATION
  TIME_DT = 60.0D0, ; Time step for time integration
  TIME_DT_UNIT = "SEC", ; Unit for TIME_DT
/
```

`(TIME_DT)` in `[PARAM_TIME]` is the time step for time integration, usually described as Δt . For the time step with each component such as atmosphere and ocean component, we use `(TIME_DT)` and `(TIME_DT_UNIT)` in namelist for each component.

4.2.3 Setting history files and output variables

To output history files, SCALE-DG uses SCALE library. Similar in SCALE-RM, a history file and output variables are configured at `[PARAM_FILE_HISTORY]` and `[HISTORY_ITEM]` in `run.conf`. The default format of the history file is configured at `[PARAM_FILE_HISTORY]`.

```

&PARAM_FILE_HISTORY
FILE_HISTORY_TITLE = "",           ; Brief description of data (See Sec.??)
FILE_HISTORY_SOURCE = "",         ; Name of the source software (See
                                   ; Sec.??)

FILE_HISTORY_INSTITUTION = "",    ; Data author (See Sec.??)
FILE_HISTORY_TIME_UNITS = "seconds", ; Unit of time axis in netCDF
FILE_HISTORY_DEFAULT_BASENAME = "history_d01", ; Base name of the output file
FILE_HISTORY_DEFAULT_POSTFIX_TIMELABEL = .false., ; Add the time label to the file name?
FILE_HISTORY_DEFAULT_TINTERVAL = 3600.DO, ; Time interval of history output
FILE_HISTORY_DEFAULT_TUNIT = "SEC", ; Unit of DEFAULT_TINTERVAL
FILE_HISTORY_DEFAULT_TSTATS_OP = "none", ; Statistics operation
FILE_HISTORY_DEFAULT_DATATYPE = "REAL4", ; Output data type: REAL4 or REAL8
FILE_HISTORY_OUTPUT_STEPO = .true., ; Output data at initial time (t=0)?
FILE_HISTORY_OUTPUT_WAIT = 0.DO, ; Time to suppress output
FILE_HISTORY_OUTPUT_WAIT_TUNIT = "SEC", ; Unit of OUTPUT_WAIT
FILE_HISTORY_OUTPUT_SWITCH_TINTERVAL = -1.DO, ; Time interval to switch the file
FILE_HISTORY_OUTPUT_SWITCH_TUNIT = "SEC", ; Unit of OUTPUT_SWITCH_TINTERVAL
FILE_HISTORY_ERROR_PUTMISS = .true., ; Check missing preparation of the
                                   ; data?
/

```

In default, each process output the history file. Although SCALE-RM supports to aggregate history files to one file, SCALE -dg has not supported it yet.

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

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

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

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

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

```

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

```

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

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

```

&HISTORY_ITEM NAME="T" /
&HISTORY_ITEM NAME="U" /
&HISTORY_ITEM NAME="V" /
&HISTORY_ITEM NAME="RAIN", TINTERVAL=600.DO, TSTATS_OP="mean" /

```

Chapter 4.3

Setting dynamical core

This section describes how to configure regional and global dynamical cores. For the parameters with mesh type and finite elements, we note that `[PARAM_ATMOS]` and `[PARAM_ATMOS_DYN]` are used, as described in Section 4.2.1.

4.3.1 Temporal scheme

The settings of temporal scheme are specified in `[PARAM_ATMOS_DYN]`.

```
&PARAM_ATMOS_DYN
EQS_TYPE = NONHYDRO3D_RHOT_HEVE, ; Choose from governing equations in Table 4.3.1
TINTEG_TYPE = ERK_SSP_10s4o_2N, ; Choose from temporal schemes in Table 4.3.3 for HEVE
; and Table 4.3.2 for HEVI
TINTEG_TYPE_TRACER = ERK_SSP3s3o, ; Choose from temporal schemes in Table 4.3.3
TIME_DT = 5D-2, ; Time step for calculation of dynamical process
TIME_DT_UNIT = sec, ; Unit for TIME_DT
MODALFILTER_FLAG = .false., ; Flag to set whether a modal filtering is used (see Sec-
tion 4.3.2)
/
```

`(TIME_DT)` is the time step for time integration with dynamical process. It is also used as the time step for tracer advection.

Table 4.3.1: List of `EQS_TYPE` available in dynamical process. THERM means which variable is used in the thermodynamics equation. $\rho\theta$ and ρe_t represent density-weighted potential temperature and total energy, respectively.

EQS_TYPE	THERM	Temporal strategy	Note
NONHYDRO3D_RHOT_HEVE	$\rho\theta$	HEVE	regional
NONHYDRO3D_RHOT_HEVI	$\rho\theta$	HEVI	regional
NONHYDRO3D_ETOT_HEVE	ρe_t	HEVE	regional, experimental
NONHYDRO3D_ETOT_HEVI	ρe_t	HEVI	regional, experimental
GLOBALNONHYDRO3D_RHOT_HEVE	$\rho\theta$	HEVE	global
GLOBALNONHYDRO3D_RHOT_HEVI	$\rho\theta$	HEVI	global
GLOBALNONHYDRO3D_ETOT_HEVE	ρe_t	HEVE	global, experimental
GLOBALNONHYDRO3D_ETOT_HEVI	ρe_t	HEVI	global, experimental

Table 4.3.2: Implicit and Explicit (IMEX) Runge–Kutta schemes available in HEVI temporal integration

Abbrev.	Order	Num. of stages (I,E)	Note	Reference
IMEX_ARK232	2	(2,3)		Giraldo et al. (2013)
IMEX_ARK324	3	(3,4)		Kennedy and Carpenter (2003)

Table 4.3.3: Explicit Runge–Kutta schemes available in HEVE temporal integration

Abbrev.	Order	Num. of stages	Note	Reference
ERK_Euler	1	1	for debug	
ERK_SSP_2s2o	2	2	SSP	Shu and Osher (1988)
ERK_SSP_3s3o	3	3	SSP	Shu and Osher (1988)
ERK_SSP_4s3o	3	4	SSP	
ERK_SSP_5s3o_2N2*	3	5	SSP	Higueras and Roldán (2019)
ERK_RK4	4	4	classical RK4	
ERK_SSP_10s4o_2N	4	10	SSP	Ketcheson (2008)

4.3.2 Modal filtering

For high-order DGM, numerical instability is likely to occur in advection-dominated flows because the numerical dissipations with the upwind numerical fluxes weaken. Furthermore, we adopted a collocation approach due to its computational efficiency. One drawback is that the aliasing errors with evaluations of the nonlinear terms can drive numerical instability.

To suppress this numerical instability, a modal filter with an exponential function is used as an additional stabilization mechanism. The fundamental parameters are the order of p_m and the non-dimensional decay strength α_m . We applied the filter to the solution vector at the final stage of the RK scheme with a timestep Δt . Then, the decay timescale for the highest mode is approximately $\Delta t / \alpha_m$.

In SCALE-DG, modal filters can be configured separately for dynamical variables and tracer variables. The switch of modal filtering exists in `[PARAM_ATMOS_DYN]` and the parameters are specified in `[PARAM_ATMOS_DYN_MODALFILTER]` and `[PARAM_ATMOS_DYN_TRACER_MODALFILTER]`. The flag for modal filtering is set as

```
&PARAM_ATMOS_DYN
MODALFILTER_FLAG = .true.,           ; Flag to set whether a modal filtering is used
TRACERADV_MODALFILTER_FLAG = .true., ; Flag to set whether a modal filtering is used for tracer
                                      variables
/
```

For the modal filtering of dynamical variables, the parameters are set as

```
&PARAM_ATMOS_DYN_MODALFILTER
MF_ALPHA_h = 1D-1,                   ; Non-dimensional strength of horizontal modal filter
MF_ORDER_h = 16,                     ; Order of horizontal modal filter
MF_ALPHA_v = 1D-1,                   ; Non-dimensional strength of vertical modal filter
MF_ORDER_v = 16,                     ; Order of vertical modal filter
/
```

while for the modal filtering of tracer variables, the parameters are set as

```
&PARAM_ATMOS_DYN_TRACER_MODALFILTER
MF_ALPHA_h = 1D-1,           ; Non-dimensional strength of horizontal modal filter
MF_ORDER_h = 16,             ; Order of horizontal modal filter
MF_ALPHA_v = 1D-1,           ; Non-dimensional strength of vertical modal filter
MF_ORDER_v = 16,             ; Order of vertical modal filter
/
```

We set the order p_m , and decay coefficient α_m such that the filter is as weak as possible while maintaining numerical stability.

4.3.3 Setting for boundary condition

Boundary conditions with atmospheric dynamical core are specified in `[PARAM_ATMOS_DYN_BND]`. For the default, slip and adiabatic conditions are imposed at rigid walls for velocity and heat, respectively. The boundary condition top and bottom boundaries are specified as

```
&PARAM_ATMOS_DYN_BND
btm_vel_bc = SLIP,           ; Velocity BC at bottom boundary
top_vel_bc = SLIP,           ; Velocity BC at top boundary
btm_thermal_bc = ADIABATIC,   ; Thermal BC at bottom boundary
top_thermal_bc = ADIABATIC,   ; Thermal BC at top boundary
/
```

In the regional model, when lateral boundaries are not periodic, we should impose lateral boundary conditions as

```
&PARAM_ATMOS_DYN_BND
north_vel_bc = SLIP,         ; Velocity BC at northern boundary
south_vel_bc = SLIP,         ; Velocity BC at southern boundary
west_vel_bc = SLIP,          ; Velocity BC at west boundary
east_vel_bc = SLIP,          ; Velocity BC at east boundary
north_thermal_bc = ADIABATIC, ; Thermal BC at northern boundary
south_thermal_bc = ADIABATIC, ; Thermal BC at southern boundary
west_thermal_bc = ADIABATIC, ; Thermal BC at west boundary
east_thermal_bc = ADIABATIC, ; Thermal BC at east boundary
/
```

4.3.4 Setting for Coriolis force

4.3.4.1 Regional model

For regional model, the Coriolis parameter is zero as the default. Thus, you should set the parameters to introduce the Coriolis force in simulations. There are two types of setting for the Coriolis parameter: f -/ β -plane and sphere. The type can be specified by `(CORIOLIS_type)` in `[PARAM_ATMOS_DYN_CORIOLIS]`.

If `(CORIOLIS_type)` is set to “PLANE”, the Coriolis parameter f is $f = f_0 + \beta(y - y_0)$. The plane for $\beta = 0$ is called f -plane, otherwise it is called β -plane. The parameters of f_0, β and y_0 is set with the parameters of `[PARAM_ATMOS_DYN_CORIOLIS]` as

```
&PARAM_ATMOS_DYN_CORIOLIS
CORIOLIS_type = NONE,           ; Type of coriolis force: NONE, PLANE, SPHERE
CORIOLIS_f0 = 0D0,             ;  $f_0$ 
CORIOLIS_beta = 0D0,           ;  $\beta_0$ 
CORIOLIS_y0 = 0D0,             ;  $y_0$ 
/
```

On the other hand, CORIOLIS_type=SPHERE, the Coriolis parameter depends on the latitude as $f = 2\Omega \sin(\phi)$, where Ω and ϕ are angular velocity of the sphere and latitude, respectively. The angular velocity of the sphere is set by (CONST_OHM) parameter of [PARAM_CONST]. The latitude of the individual grids is determined depending on the map projection.

4.3.4.2 Global model

For the global model, the Coriolis parameter is nonzero for the default and depends on the latitude. The angular velocity of rotation is set by (CONST_OHM) parameter of [PARAM_CONST]. If the Coriolis force is not considered, you should set (CONST_OHM) to be zero.

4.3.5 Setting for shallow-atmosphere approximation

For the global model, the shallow-atmosphere approximation can be switched by (SHALLOW_ATM_APPROX_FLAG) in [PARAM_ATMOS_MESH].

```
&PARAM_ATMOS_MESH
SHALLOW_ATM_APPROX_FLAG = .true., ; Enable shallow-atmosphere approximation
/
```

By default, the shallow-atmosphere approximation is applied, and the metric terms and the Coriolis terms are simplified consistently with the governing equations under this approximation. When (SHALLOW_ATM_APPROX_FLAG) is set to .false., the deep-atmosphere formulation is used, and the radial variation of the geometry is retained in the global dynamical core.

Currently, the deep-atmosphere formulation is available only for GLOBALNONHYDRO3D_RHOT_HEVE and GLOBALNONHYDRO3D_RHOT_HEVE_ASIS. When the other global dynamical-core options are used, (SHALLOW_ATM_APPROX_FLAG) should be set to .true..

Chapter 4.4

Setting the physical process

4.4.1 Cloud Micro-Physics

Cloud microphysics is configured by two namelists. First, the cloud microphysics component is activated by `(ATMOS_PHY_MP_DO)` in `[PARAM_ATMOS]`. Then, the microphysics scheme and its calling interval are specified in `[PARAM_ATMOS_PHY_MP]`.

```
&PARAM_ATMOS
ATMOS_PHY_MP_DO = .true., ; Enable cloud microphysics
/
```

```
&PARAM_ATMOS_PHY_MP
MP_TYPE = "TOMITA08", ; Choose from cloud microphysics schemes in Table 4.4.1
TIME_DT = 60.D0, ; Time interval of cloud microphysics process
TIME_DT_UNIT = "SEC", ; Unit for TIME_DT
do_precipitation = .true., ; Enable sedimentation of precipitation particles
do_negative_fixer = .true., ; Enable negative-value fixer for water substances
ntmax_sedimentation = 1, ; Number of substeps for sedimentation
/
```

The available choices of `(MP_TYPE)` are listed in Table 4.4.1. Several options common to cloud microphysics schemes can be specified in `[PARAM_ATMOS_PHY_MP]`. `(TIME_DT)` and `(TIME_DT_UNIT)` specify the time interval of the cloud microphysics process. `(do_precipitation)` controls whether sedimentation of precipitation particles is calculated. When precipitation is calculated, `(do_negative_fixer)` controls the fixer for negative values of water substances and `(ntmax_sedimentation)` specifies the number of substeps with which sedimentation is calculated.

The Kessler and Tomita08 schemes use the cloud microphysics components provided by the SCALE library. Therefore, the scheme-dependent parameters are basically the same as those used in SCALE-RM. For the detailed description of these parameters, refer to the SCALE-RM User's Guide.

Table 4.4.1: Cloud microphysics schemes available in SCALE-DG .

MP_TYPE	Description	Reference
KESSLER	Three-class one-moment bulk scheme	Kessler (1969)
TOMITA08	Six-class one-moment bulk scheme	Tomita (2008)
LSCOND	Large-scale condensation scheme	

Note that it is necessary to specify the same scheme for `(MP_TYPE)` in the configuration files for both init and run executions.

4.4.1.1 Kessler scheme

This is a three-class one-moment bulk microphysics scheme based on [Kessler \(1969\)](#). It treats warm-rain processes using water vapor, cloud water, and rain water. Since ice-phase water substances are not included, this scheme is mainly suitable for idealized warm-cloud experiments or for cases where a simple and computationally inexpensive cloud microphysics scheme is sufficient. This scheme uses the SCALE library, and its parameters are configured as in SCALE-RM.

4.4.1.2 Tomita08 scheme

This is a six-class one-moment bulk microphysics scheme based on [Tomita \(2008\)](#). It treats water vapor, cloud water, rain water, cloud ice, snow, and graupel. This scheme is appropriate for experiments where mixed-phase cloud and precipitation processes should be represented. This scheme uses the SCALE library, and its parameters are configured as in SCALE-RM.

4.4.1.3 Large-scale condensation scheme

The large-scale condensation scheme is a simplified moist-physics scheme. It represents condensation and associated latent heating without explicitly calculating detailed bulk microphysical conversion processes among several hydrometeor categories. This scheme is useful for idealized moist experiments where the feedback between moisture and atmospheric motion is required, but detailed cloud microphysics is not the main target. The precipitation is not explicitly calculated in this scheme, and the condensed water is removed from the atmosphere with a timescale of microphysics timestep.

4.4.2 Cumulus parameterization

Cumulus parameterization represents the effects of unresolved moist convection on the resolved atmospheric state. The cumulus parameterization component is activated by (ATMOS_PHY_CP_DO) in [PARAM_ATMOS]. The cumulus parameterization scheme and its calling interval are specified in [PARAM_ATMOS_PHY_CP].

```
&PARAM_ATMOS
ATMOS_PHY_CP_DO = .true., ; Enable cumulus parameterization
/
```

```
&PARAM_ATMOS_PHY_CP
CP_TYPE = "MOIST_CONV_ADJUSTMENT", ; Choose from cumulus parameterization schemes in Table
                                         4.4.2
TIME_DT = 60.DO, ; Time interval of cumulus parameterization process
TIME_DT_UNIT = "SEC", ; Unit for TIME_DT
/
```

The available choices of (CP_TYPE) are listed in Table 4.4.2. (TIME_DT) and (TIME_DT_UNIT) specify the time interval of the cumulus parameterization process.

Table 4.4.2: Cumulus parameterization schemes available in SCALE-DG.

CP_TYPE	Description	Reference
MOIST_CONV_ADJUSTMENT	Moist convective adjustment scheme	Manabe et al. (1965)

4.4.2.1 Moist convective adjustment scheme

The moist convective adjustment (MCA) scheme is a simple cumulus parameterization based on the concept of [Manabe et al. \(1965\)](#). The scheme diagnoses saturated convective instability in nearly saturated atmospheric columns and adjusts unstable layers toward a moist-neutral thermodynamic state.

Saturated convective instability is diagnosed using the vertical distribution of saturated moist static energy (MSE). When a nearly saturated layer has saturated MSE decreasing with height, a contiguous unstable region is identified and subject to the adjustment. The adjusted thermodynamic profile is determined so as to satisfy the available-water constraint and conservation of column-integrated MSE. If a consistent adjusted state cannot be obtained within the initially diagnosed unstable region, the adjustment region is extended vertically and the calculation is repeated.

Water vapor removed by the adjustment is diagnosed as precipitation and is immediately removed from the atmospheric column; no condensate is retained by the MCA scheme. After each adjustment, the atmospheric state is updated and saturated convective instability is diagnosed again over the entire column. This diagnosis-adjustment cycle is repeated until no saturated convective instability remains.

The MCA scheme is applied at the interval specified by (TIME_DT) and (TIME_DT_UNIT) in the cumulus parameterization process. The precipitation generated during the successive adjustments is accumulated within each call of the scheme and contributes to the surface precipitation flux.

4.4.3 Turbulence

The turbulence scheme represents the sub-grid scale turbulent transport of momentum, heat, and tracers. The turbulence component is activated by (`ATMOS_PHY_TB_DO`) in `[PARAM_ATMOS]`. The turbulence scheme and its calling interval are specified in `[PARAM_ATMOS_PHY_TB]`.

```
&PARAM_ATMOS
ATMOS_PHY_TB_DO = .true., ; Enable turbulence process
/
```

```
&PARAM_ATMOS_PHY_TB
TB_TYPE = "SMAGORINSKY", ; Choose from turbulence schemes in Table 4.4.3
TIME_DT = 60.DO, ; Time interval of turbulence process
TIME_DT_UNIT = "SEC", ; Unit for TIME_DT
/
```

The available choices of (`TB_TYPE`) are listed in Table 4.4.3. (`TIME_DT`) and (`TIME_DT_UNIT`) specify the time interval of the turbulence process.

Table 4.4.3: Turbulence schemes available in SCALE-DG .

TB_TYPE	Description	Note
SMAGORINSKY	Smagorinsky–Lilly type SGS turbulence scheme	for regional model
SMAGORINSKY_GLOBAL	Smagorinsky–Lilly type SGS turbulence scheme	for global model
DNS	Constant viscosity and diffusivity	mainly for idealized experiments

4.4.3.1 Smagorinsky–Lilly type scheme

The Smagorinsky–Lilly type scheme is a sub-grid scale turbulence model based on Smagorinsky (1963) and Lilly (1962). The eddy viscosity is diagnosed from the local strain rate and a characteristic filter length. The stratification effect is also taken into account following Brown et al. (1994). This scheme is suitable for large-eddy simulations and high-resolution experiments where sub-grid scale turbulent transport should be represented. For the detailed formulation, refer to the model description.

The scheme-dependent parameters are specified in `[PARAM_ATMOS_PHY_TB_DGM_SMG]`.

```
&PARAM_ATMOS_PHY_TB_DGM_SMG
Cs = 0.13D0, ; Smagorinsky constant
NU_MAX = 10000.DO, ; Maximum eddy viscosity
filter_fac = 2.0D0, ; Factor for the filter length
consistent_tke = .true., ; Use TKE consistently in the SGS stress
/
```

(`Cs`) specifies the Smagorinsky constant. (`NU_MAX`) specifies the upper limit of eddy viscosity. (`filter_fac`) controls the characteristic filter length used in the SGS model. (`consistent_tke`) controls whether the parameterized turbulent kinetic energy is included consistently in the SGS stress.

The parameterized eddy viscosity, eddy diffusivity, and turbulent kinetic energy can be output as history variables, for example, NU, KH, and TKE.

4.4.3.2 Smagorinsky–Lilly type scheme for global model

The scheme-dependent parameters are specified in [\[PARAM_ATMOS_PHY_TB_DGM_GLOBALSMG\]](#).

```
&PARAM_ATMOS_PHY_TB_DGM_GLOBALSMG
Cs = 0.13D0,                ; Smagorinsky constant
NU_MAX = 10000.D0,          ; Maximum eddy viscosity
filter_fac = 2.0D0,         ; Factor for the filter length
consistent_tke = .true.,    ; Use TKE consistently in the SGS stress
/
```

The parameters have the same meaning as those in [\[PARAM_ATMOS_PHY_TB_DGM_SMG\]](#). The global version should be used with the global dynamical core.

4.4.3.3 Constant viscosity and diffusivity

For idealized experiments or direct numerical simulations (DNS), constant viscosity and diffusivity can be used by setting [\(TB_TYPE\)](#) to DNS.

```
&PARAM_ATMOS_PHY_TB
TB_TYPE = "DNS",            ; Constant viscosity and diffusivity
TIME_DT = 60.D0,           ; Time interval of turbulence process
TIME_DT_UNIT = "SEC",      ; Unit for TIME_DT
/
```

The scheme-dependent parameters are specified in [\[PARAM_ATMOS_PHY_TB_DGM_DNS\]](#).

```
&PARAM_ATMOS_PHY_TB_DGM_DNS
DNS_NU = 0.D0,              ; Constant viscosity
DNS_MU = 0.D0,              ; Constant diffusivity
/
```

4.4.4 Planetary Boundary Layer Scheme

The planetary boundary layer (PBL) scheme parameterizes vertical turbulent mixing in the atmospheric boundary layer. It is intended for Reynolds-averaged Navier–Stokes (RANS) simulations. The PBL process is configured by two namelists. First, the PBL component is enabled by (`ATMOS_PHY_BL_DO`) in [`PARAM_ATMOS`]. Then, the PBL scheme and its calling interval are specified in [`PARAM_ATMOS_PHY_BL`].

```
&PARAM_ATMOS
ATMOS_PHY_BL_DO = .true., ; Enable planetary boundary layer parameterization
/
```

```
&PARAM_ATMOS_PHY_BL
BL_TYPE = "MYNN_LEVEL2", ; Choose from PBL schemes in Table 4.4.4
TIME_DT = 600.DO, ; Time interval of PBL process
TIME_DT_UNIT = "SEC", ; Unit for TIME_DT
/
```

The available choices of (`BL_TYPE`) are listed in Table 4.4.4.

Table 4.4.4: Planetary boundary layer schemes available in SCALE-DG .

BL_TYPE	Description	Reference
MYNN_LEVEL2	MYNN level 2 scheme	Nakanishi and Niino (2006)

4.4.4.1 MYNN level 2 scheme

The current implementation provides the Mellor–Yamada–Nakanishi–Niino (MYNN) level 2 turbulence closure scheme [Nakanishi and Niino \(2006\)](#). The MYNN level 2 scheme diagnoses turbulent diffusion coefficients for momentum and scalars from the local stability of the atmosphere and applies vertical turbulent mixing within the planetary boundary layer.

4.4.5 Radiation scheme

The radiation process is configured by two namelists. First, the radiation component is enabled by (`ATMOS_PHY_RD_DO`) in [`PARAM_ATMOS`]. Then, the radiation scheme and its calling interval are specified in [`PARAM_ATMOS_PHY_RD`].

```
&PARAM_ATMOS
ATMOS_PHY_RD_DO = .true., ; Enable radiation
/
```

```
&PARAM_ATMOS_PHY_RD
RD_TYPE = "SIMPLE", ; Choose from radiation schemes in Table 4.4.5
RD_SOLARINS_TYPE = "SIMPLE", ; Type of incoming solar-insolation scheme
TIME_DT = 600.DO, ; Time interval of the radiation process
TIME_DT_UNIT = "SEC", ; Unit for TIME_DT
/
```

The available choices of (`RD_TYPE`) are listed in Table 4.4.5. The incoming solar-insolation scheme is selected by (`RD_SOLARINS_TYPE`). Currently, "SIMPLE" and "NONE" are available. When "SIMPLE" is selected, its detailed configuration is given in [`PARAM_ATMOS_SOLARINS_SIMPLE`] (Section 4.4.5.1).

Table 4.4.5: Radiation schemes available in SCALE-DG .

RD_TYPE	Description	Reference
SIMPLE	Simplified radiation scheme	Geen et al. (2016); Vallis et al. (2018)

4.4.5.1 Configuration of incoming solar radiation

The incoming solar radiation used by the simplified radiation scheme is prescribed by an idealized solar-insolation model. Two types of idealized forcing are currently available: a spatially uniform constant flux and an annual-mean insolation that varies with latitude. These settings are configured in [`PARAM_ATMOS_SOLARINS_SIMPLE`].

```
&PARAM_ATMOS_SOLARINS_SIMPLE
SOLARINS_TYPE = "ANNUAL_MEAN", ; Choose from solar-insolation types in Table ??
ORBIT_REFERENCE_YEAR = 2000, ; Reference year for orbital parameters
ANNUAL_MEAN_YEAR = 2000, ; Year used for the annual-mean calculation
ANNUAL_SAMPLES_PER_DAY = 1, ; Number of samples per day
/
```

The choice of (`SOLARINS_TYPE`) determines how the incoming solar flux is specified. The available options are summarized in Table 4.4.6.

When `SOLARINS_TYPE="CONST"` is selected, (`CONST_FLUX`) is applied uniformly over the model domain.

```
&PARAM_ATMOS_SOLARINS_SIMPLE
SOLARINS_TYPE = "CONST", ; Type of idealized solar insolation;
CONST_FLUX = 340.DO, ; Constant incoming solar flux [W m-2]
/
```

Table 4.4.6: Incoming solar radiation options available in SCALE-DG .

SOLARINS_TYPE	Description	Reference
CONST	Spatially uniform incoming solar radiation	
ANNUAL_MEAN	Annual-mean incoming solar radiation	
FRIERSON2006	Idealized latitudinal distribution of incoming solar radiation	Frierson et al. (2006)

When SOLARINS_TYPE="ANNUAL_MEAN" is selected, the daily-mean solar insolation is calculated from the orbital configuration and latitude and then averaged over (ANNUAL_MEAN_YEAR). The default is "ANNUAL_MEAN". (ORBIT_REFERENCE_YEAR) specifies the reference year used to initialize the orbital parameters in the SCALE solar-insolation module. (ANNUAL_SAMPLES_PER_DAY) specifies the number of equally spaced samples used within each day when forming the annual mean. The default value is one sample per day.

When FRIERSON2006 is selected, the incoming solar radiation is prescribed using the idealized latitudinal distribution proposed by [Frierson et al. \(2006\)](#):

$$S(\phi) = \frac{S_0}{4} \left[1 + \frac{\Delta_s}{4} (1 - 3 \sin^2 \phi) \right], \quad (4.4.1)$$

where S_0 is the solar constant, ϕ is latitude, and Δ_s controls the meridional contrast of the incoming solar radiation. The parameter Δ_s can be specified by (FRIERSON2006_DELTA_s) in [PARAM_ATMOS_SOLARINS_SIMPLE]. Its default value is 1.4, following [Frierson et al. \(2006\)](#). The incoming solar radiation is independent of longitude and time; therefore, diurnal and seasonal variations are not included.

4.4.5.2 Simplified radiation scheme

The simplified radiation schemes are intended for idealized atmospheric experiments. Compared with comprehensive radiative transfer models, they employ simplified spectral representations while retaining the dominant radiative heating processes.

Two formulations are available: a gray longwave radiation scheme and a two-band longwave plus one-band shortwave radiation scheme. The formulation and its optical-depth parameters are configured in [PARAM_ATMOS_PHY_RD_DGM_SIMPLE]. The default is the gray longwave radiation scheme.

Gray longwave radiation scheme

A basic configuration using the default parameters is, for example,

```
&PARAM_ATMOS_PHY_RD_DGM_SIMPLE
SIMPLE_RD_TYPE = "Gray",           ; Type of simplified radiation scheme
V2018_GRAY_OPTDEPTH_PARAMS_TYPE = "V2018", ; Parameter set for gray optical depth
DiffusivityFactor = 1.D0,         ; Diffusivity factor for two-stream approximation
pCO2 = 360.D0,                   ; CO2 concentration [ppmv]
/
```

For the gray scheme, the optical-depth parameters are controlled by (V2018_GRAY_OPTDEPTH_PARAMS_TYPE). The available choices are "B02013", "V2018", and "USER", with "V2018" as the default. When

"USER" is selected, the four coefficients (`V2018_GRAY_OPTDEPTH_LW_PARAMS`) must be specified in the order A, μ, B, C , corresponding to Eq. (5) of Vallis et al. (2018).

The preset coefficient values used internally are $(A, \mu, B, C) = (0.8678, 1.0, 1997.9, 0.0)$ for "B02013" and $(0.1627, 1.0, 1997.9, 0.17)$ for "V2018".

For the gray radiation scheme, the shortwave optical depth is set to zero. Therefore, the prescribed incoming solar flux reaches the surface without attenuation by atmospheric shortwave absorption.

Two-band longwave and one-band shortwave scheme

```
&PARAM_ATMOS_PHY_RD_DGM_SIMPLE
SIMPLE_RD_TYPE = "LWbnd2+SWbnd1",           ; Two LW bands and one SW band
V2018_LWbnd2SWbnd1_OPTDEPTH_PARAMS_TYPE = "V2018", ; Optical-depth parameter set
V2018_LWbnd2SWbnd1_OPTDEPTH_SW_kadd = 4,      ; Number of layers added above the
                                                ; model top
V2018_LWbnd2SWbnd1_OPTDEPTH_SW_QV_above_MOT = 1.D-3, ; Specific humidity above the model
                                                ; top
DiffusivityFactor = 1.D0,                     ; Diffusivity factor
pCO2 = 360.D0,                                ; CO2 concentration [ppmv]
/
```

For this scheme, (`V2018_LWbnd2SWbnd1_OPTDEPTH_PARAMS_TYPE`) can be set to "V2018" or "USER". When "V2018" is selected, the optical-depth coefficients and the longwave spectral fractions are prescribed internally following Vallis et al. (2018). When "USER" is selected, the user specifies (`V2018_LWbnd2SWbnd1_OPTDEPTH_{SW,LW,LW_window}_PARAMS`) and (`V2018_LWbnd2SWbnd1_LW_window_FRAC`).

The shortwave coefficients contain a_{SW} and c_{SW} ; the non-window longwave coefficients contain a_{LW} , b_{LW} , c_{LW} , d_{LW} ; and the longwave-window coefficients contain a_{win} , b_{win} , c_{win} , d_{win} . The window fraction specifies the fraction of the longwave spectrum assigned to the 8 – 14 μm window band.

The vertical coordinate in SCALE-DG is based on the height-based terrain-following coordinate. Thus, the optical depth above the model top is not well defined. To address this, the two parameters (`V2018_LWbnd2SWbnd1_OPTDEPTH_SW_kadd`) and (`V2018_LWbnd2SWbnd1_OPTDEPTH_SW_QV_above_MOT`) control the treatment of shortwave optical thickness above the model top; their default values are 4 and 10^{-3} , respectively.

Common parameters for simplified radiation schemes

(`DiffusivityFactor`) is the diffusivity factor used in the two-stream approximation. Although a value around 1.66 is commonly used for a plane-parallel atmosphere with isotropic scattering, the present idealized scheme uses 1.0 following Vallis et al. (2018). (`pCO2`) specifies the CO₂ concentration in ppmv and has a default value of 360.

4.4.6 Surface flux

Surface fluxes represent the exchange of momentum, heat, and moisture between the surface and the atmosphere. Surface fluxes are configured by two namelists. First, the surface flux component is activated by (`ATMOS_PHY_SF_DO`) in [`PARAM_ATMOS`]. Then, the surface flux scheme and its calling interval are specified in [`PARAM_ATMOS_PHY_SFC`].

```
&PARAM_ATMOS
ATMOS_PHY_SF_DO = .true., ; Enable surface flux process
/
```

```
&PARAM_ATMOS_PHY_SFC
SFCFLX_TYPE = "CONST", ; Choose from surface flux schemes in Table 4.4.7
TIME_DT = 60.DO, ; Time interval of surface flux process
TIME_DT_UNIT = "SEC", ; Unit for TIME_DT
/
```

The available choices of (`SFCFLX_TYPE`) are listed in Table 4.4.7. The constant surface-flux scheme uses the corresponding SCALE library component, and its parameters are configured as in SCALE-RM. (`TIME_DT`) and (`TIME_DT_UNIT`) specify the time interval of the surface-flux process.

Table 4.4.7: Surface flux schemes available in SCALE-DG .

SFCFLX_TYPE	Description
CONST	Constant surface-flux scheme
SIMPLE	Simple bulk surface-flux scheme with constant bulk coefficients

The surface temperature used in the surface-flux calculation is stored as (`SFC_TEMP`). If no other component updates it, its default value can be specified by (`ATMOS_PHY_SFC_DEFAULT_SFC_TEMP`) in [`PARAM_ATMOS_PHY_SFC`] as

```
&PARAM_ATMOS_PHY_SFC
ATMOS_PHY_SFC_DEFAULT_SFC_TEMP = 300.DO, ; Default surface temperature [K]
/
```

The surface flux variables can be output as history variables. For example, `SFLX_MU`, `SFLX_MV`, and `SFLX_MW` are the momentum fluxes, `SFLX_SH` and `SFLX_LH` are the sensible and latent heat fluxes, and `SFLX_QV` is the water vapor flux. The surface temperature is output as `SFC_TEMP`.

4.4.6.1 Constant surface-flux scheme

This scheme prescribes horizontally uniform sensible and latent heat fluxes. The surface momentum flux is calculated using prescribed parameters such as a bulk coefficient for momentum or friction velocity. This option is useful for idealized experiments where simple surface forcing is sufficient. This scheme uses the SCALE library, and its parameters are configured by [`PARAM_ATMOS_PHY_SF_CONST`] as in SCALE-RM.

4.4.6.2 Simple bulk surface-flux scheme

This scheme calculates surface momentum, sensible heat, and latent heat fluxes using constant bulk coefficients. The surface moisture is diagnosed from the surface temperature and surface pressure

with an evaporation efficiency. This option is useful for idealized experiments where the surface fluxes should depend on the near-surface atmospheric state while keeping the parameterization simple.

The scheme-dependent parameters are specified in `[PARAM_ATMOS_PHY_SF_SIMPLE]`.

```
&PARAM_ATMOS_PHY_SF_SIMPLE
ATMOS_PHY_SF_CONST_CM = 1.1D-3, ; Constant bulk coefficient for momentum
ATMOS_PHY_SF_CONST_CH = 4.4D-3, ; Constant bulk coefficient for heat
ATMOS_PHY_SF_CONST_CE = 4.4D-3, ; Constant bulk coefficient for evaporation
ATMOS_PHY_SF_BULK_BETA = 1.0D0, ; Evaporation efficiency
ATMOS_PHY_SF_U_MINM = 0.0D0, ; Minimum wind speed for surface fluxes [m/s]
/
```

4.4.7 Ocean model

The ocean component is activated and configured in `[PARAM_OCEAN]`. The computational mesh for the ocean component is selected by `(OCEAN_MESH_TYPE)`, and the ocean dynamics is enabled by `(OCEAN_DYN_DO)`.

```
&PARAM_OCEAN
  ACTIVATE_FLAG = .true.,      ; Enable the ocean component
  OCEAN_MESH_TYPE = "GLOBAL",  ; Type of ocean mesh: REGIONAL or GLOBAL
  OCEAN_DYN_DO = .true.,      ; Enable ocean dynamics
  TIME_DT = 3600.D0,          ; Time step of the ocean component
  TIME_DT_UNIT = "SEC",       ; Unit for TIME_DT
/
```

`(OCEAN_MESH_TYPE)` can be set to "REGIONAL" or "GLOBAL".

4.4.7.1 Setting grids for ocean model

The computational grid of the ocean model is configured in `[PARAM_OCEAN_MESH]`. The available parameters depend on the mesh type selected by `(OCEAN_MESH_TYPE)` in `[PARAM_OCEAN]`.

At present, the horizontal mesh configuration of the ocean component must be identical to that of the atmospheric component. Therefore, the horizontal domain size, MPI decomposition, number of horizontal elements, and horizontal polynomial order should be specified consistently for both components. This restriction arises from the present atmosphere–ocean coupling, which exchanges two-dimensional fields without horizontal remapping.

Regional mesh

For the regional ocean model using the Cartesian coordinates, the computational domain and DG mesh are specified in `[PARAM_OCEAN_MESH]`.

```
&PARAM_OCEAN_MESH
  NprcX = 1,      ; Number of MPI processes in the x-direction
  NprcY = 1,      ; Number of MPI processes in the y-direction
  NeX = 2,        ; Number of elements in x per MPI process
  NeY = 2,        ; Number of elements in y per MPI process
  NeZ = 1,        ; Number of vertical elements
  dom_xmin = 0.D0, ; Minimum x-coordinate [m]
  dom_xmax = 100.D3, ; Maximum x-coordinate [m]
  dom_ymin = 0.D0, ; Minimum y-coordinate [m]
  dom_ymax = 100.D3, ; Maximum y-coordinate [m]
  dom_zmin = 0.D0, ; Minimum vertical coordinate [m]
  dom_zmax = 50.D0, ; Maximum vertical coordinate [m]
  PolyOrder_h = 2, ; Horizontal polynomial order
  PolyOrder_v = 1, ; Vertical polynomial order
/
```

An arbitrary non-uniform element spacing can also be specified in the vertical direction by giving the locations of all element boundaries using `(FZ)`.

```
&PARAM_OCEAN_MESH
  FZ = 0.D0, 10.D0, 50.D0, ; Vertical element boundaries [m]
/
```

Global mesh

For the global ocean model using the cubed-sphere coordinates, the grid is also configured in `[PARAM_OCEAN_MESH]`.

```
&PARAM_OCEAN_MESH
Nprc = 6,                ; Total number of MPI processes
NeGX = 6,                ; Number of elements in each cubed-sphere panel
NeGY = 6,                ; Number of elements in each cubed-sphere panel
NeZ = 1,                 ; Number of vertical elements
PolyOrder_h = 2,         ; Horizontal polynomial order
PolyOrder_v = 1,         ; Vertical polynomial order
dom_zmin = 0.D0,         ; Minimum vertical coordinate [m]
dom_zmax = 50.D0,        ; Maximum vertical coordinate [m]
SHALLOW_OCN_APPROX_FLAG = .true., ; Apply shallow-ocean approximation
/
```

4.4.7.2 Ocean-model configurations

Currently, SCALE-DG provides two ocean-model options: the slab ocean model and a fixed sea-surface temperature option. The slab ocean model predicts the mixed-layer temperature by solving the heat budget of a prescribed mixed layer, while the fixed-temperature option keeps the initial sea-surface temperature unchanged throughout the simulation. These are configured in `[PARAM_OCEAN_DYN]`.

A typical configuration is

```
&PARAM_OCEAN_DYN
EQS_TYPE = "SLAB",       ; Ocean model; SLAB or FIXED_INIT_TEMP
TIME_DT = 3600.D0,       ; Time interval of ocean dynamics
TIME_DT_UNIT = "SEC",    ; Unit for TIME_DT
/
```

When `EQS_TYPE="SLAB"` is selected, the ocean temperature evolves according to the heat budget of the slab ocean. When `EQS_TYPE="FIXED_INIT_TEMP"` is selected, the initially specified ocean temperature is kept fixed. For the slab ocean model, the temperature is advanced with a forward Euler method using `(TIME_DT)`.

Note

For the slab ocean model, the ocean temperature is assumed to be vertically uniform. Therefore, a zeroth-order polynomial would be the natural choice in the vertical direction. However, the current mesh implementation does not support `PolyOrder_v=0`. Thus, `(PolyOrder_v)` must be set to one or larger.

Although multiple vertical nodal values are therefore retained redundantly, the slab-ocean temperature tendency is applied uniformly to all vertical degrees of freedom, so that the slab-ocean calculation is effectively vertically uniform.

4.4.7.3 Surface fluxes and atmosphere–ocean exchange

For the surface-flux calculation, the ocean component receives the near-surface atmospheric state from the atmosphere through the coupler, including density, pressure, temperature, wind velocity, specific humidity, and the thickness of the lowest atmospheric layer. The downward direct

shortwave and downward diffuse longwave radiative fluxes are also passed from the atmospheric radiation component to the ocean component.

Using these atmospheric quantities together with the ocean surface temperature, the ocean component calculates the surface momentum, sensible-heat, latent-heat, and water-vapor fluxes using surface-flux schemes. The ocean surface temperature and these surface fluxes are then returned to the atmospheric component through the coupler and used as the lower boundary condition of the atmosphere.

In the current SCALE-DG, the ocean surface fluxes are calculated using the simple bulk surface-flux scheme described in Section 4.4.6.2. The same module and its parameters are shared with the atmospheric surface-flux component. Therefore, the atmospheric surface-flux component should be enabled and the simple bulk scheme should be selected as follows:

```
&PARAM_ATMOS
ATMOS_PHY_SF_DO = .true.,    ; Enable the surface-flux component
/
```

```
&PARAM_ATMOS_PHY_SFC
SFCFLX_TYPE = "SIMPLE",      ; Use the simple bulk surface-flux scheme
TIME_DT = 600.DO,           ; Time interval of the atmospheric surface process
TIME_DT_UNIT = "SEC",       ; Unit for TIME_DT
/
```

The parameters of the simple bulk surface-flux scheme are specified in `[PARAM_ATMOS_PHY_SF_SIMPLE]`. See Section 4.4.6.2 for details.

Chapter 4.5

Common settings

Chapter 4.6

Post-processing

4.6.1 Regrid tool

regrid.tool has the following features to reduce the annoyance of handling netCDF files output from SCALE-DG .

- Combine the decomposed multiple files into a single file.
- Convert a single file or multiple files into multiple files with different number of decomposition.
- Regrid data from the original resolution to different resolutions.
- Regrid data from the model-level to the different vertical coordinates such as hight-level and pressure-level.
- Regrid data from the model grid to the geodesic (latitude-longitude) coordinates.

4.6.1.1 Basic usage

The mesh settings for regrid.tool are specified in [\[PARAM_REGRID_MESH\]](#).

```
&PARAM_REGRID_MESH
  in_MeshType = CUBEDSPHERE3D, ; Mesh type of input data
  out_MeshType = LONLAT3D, ; Mesh type of output data
/
```

The settings of data interpolation for regrid.tool are specified in [\[PARAM_REGRID_INTERP_FIELD\]](#).

```
&PARAM_REGRID_INTERP_FIELD
  in_basename = history, ; Base name of input file(s)
  vars = "W", "U", "V", ; Name of target variables
  out_tinterval = 1, ; Time interval when outputting data (default: 1)
/
```

The settings of output data for regrid.tool are specified in [\[PARAM_REGRID_FILE\]](#).

```
&PARAM_REGRID_FILE
  out_basename = history, ; Base name of output file(s)
  out_UniformGrid = .false., ; Flag whether data is interpolated to a uniform grid when outputting files
/
```

The detail settings of input or output mesh for regrid_tool are specified as follows: The items in namelists are similar with that in configuration files for running SCALE-DG .

For the 3D cubed-sphere mesh, the settings are specified as

```
&PARAM_REGRID_(IN/OUT)MESH3D_CUBEDSPHERE
  NLocalMeshPerPrc = 1,
  Nprc = 24,
  NeGX = 8,
  NeGY = 8,
  NeGZ = 4,
  dom_zmin = 0.0D0,
  dom_zmax = 30.0D3,
  PolyOrder_h = 7,
  PolyOrder_v = 7,
  Fz = 0.0D0, 3000D0, 8000.0D0, 15000.0D0, 30000.0D0,
/
```

For the 2D cubed-sphere mesh,

```
&PARAM_REGRID_(IN/OUT)MESH2D_CUBEDSPHERE
  NLocalMeshPerPrc = 1,
  Nprc = 24,
  NeGX = 8,
  NeGY = 8,
  PolyOrder_h = 7,
/
```

For the 3D structured mesh such as the Cartesian coordinates and longitude-latitude coordinates, the settings are specified as

```
&PARAM_REGRID_(IN/OUT)MESH3D_STRUCTURED
  NprcX = 4,
  NeX = 16,
  NprcY = 2,
  NeY = 16,
  NeGZ = 4,
  PolyOrder_h = 3,
  PolyOrder_v = 7,
  dom_xmin = 0.0D0,
  dom_xmax = 360.0D0,
  dom_ymin = -90.0D0,
  dom_ymax = 90.0D0,
  dom_zmin = 0.0D0,
  dom_zmax = 30.0D3,
  Fz = 0.0D0, 3000D0, 8000.0D0, 15000.0D0, 30000.0D0,
/
```

For the 2D structured mesh,

```
&PARAM_REGRID_(IN/OUT)MESH2D_STRUCTURED
```

```
NprcX = 4, ;
NeX = 16, ;
NprcY = 2, ;
NeY = 16, ;
PolyOrder_h = 3, ;
dom_xmin = 0.0D0, ;
dom_xmax = 360.0D0, ;
dom_ymin = -90.0D0, ;
dom_ymax = 90.0D0, ;
/
```

4.6.1.2 Regrid data from model-level coordinate to other vertical coordinate

For regridding data from model-level coordinate to other vertical coordinate, we need add `[PARAM_REGRID_VCOORD]` to configuration files for regrid.tool .

If we want to interpolate data to pressure coordinate, the namelist is specified as

```
&PARAM_REGRID_VCOORD
```

```
vintrp_name = 'PRESSURE', ;
out_NeZ = 10, ;
out_PolyOrder_v = 3, ;
out_dom_vmin = 1000D0, ;
out_dom_vmax = 20D2, ;
out_Fz = 1000D2, 950D2, 850D2, 790D2, 680D2, 550D2, 400D2, 250D2, 100D2, 50D2, 30D2, ;
PolyOrder_v = 7, ;
extrapolate = .true., ;
/
```

If we want to interpolate data to actual height coordinate, the namelist is specified as

```
&PARAM_REGRID_VCOORD
```

```
vintrp_name = 'HEIGHT', ;
out_NeZ = 6, ;
out_PolyOrder_v = 7, ;
out_dom_vmin = 0D0, ;
out_dom_vmax = 40D2, ;
out_Fz = 0D0, 4.0D3, 9.D3, 15D3, 22.D3, 30.D3, 40.D3, ;
PolyOrder_v = 7, ;
in_topofile_basename = "outdata/topo", ;
topo_varname = "topo", ;
/
```

Note that topography data specified by `in_topofile_basename` need to be interpolated to a 2D mesh consistent to `[PARAM_REGRID_(IN/OUT)MESH3D_*]`.

4.6.2 Spectral analysis tool

`spectral_analysis` is a post-processing tool for calculating horizontal spectra from the output files of SCALE-DG. It reads variables in the SCALE-netCDF files and outputs spectral coefficients and one-dimensional spectra to a NetCDF file. This tool is useful, for example, for analyzing kinetic energy spectra at specified vertical levels. Currently, this tool is available only for the output data of regional simulations on the Cartesian coordinates.

To compile `spectral_analysis`, execute the following command under the top directory after the compilation of SCALE-DG :

```
$ cd model/atm_nonhydro3d/util/spectral_analysis/
$ make
```

If the compilation succeeds, the executable binary file is generated under the `bin` directory.

The basic settings are specified in `[PARAM_SPECTRAL_ANALYSIS]`. For example,

```
&PARAM_SPECTRAL_ANALYSIS
in_bs_filebase = "init_00010101-000000.000", ; Base name of input file for basic state
in_filebase = "history", ; Base name of input history files
out_filebase = "energy_spectra", ; Base name of output file
target_proc_num_tot = 32, ; Total number of MPI processes used in the
; original run
LevelNum = 1, ; Number of target vertical levels
TARGET_LEVELS = 800.D0, ; Target vertical levels
level_units = "m", ; Unit of target vertical levels
VARS = "U", "V", "W", ; Target 3D variables
SpectralAnalysisType = "Uniform_Sampling", ; Type of spectral analysis
ST_NsamplePtPerElem1D = 8, ; Number of sampling points per element
ST_ks = -64, ST_ke = 64, ; Range of wavenumber in the first direction
ST_ls = -64, ST_le = 64, ; Range of wavenumber in the second direction
KinEnergyAnalysisFlag = .true., ; Output kinetic energy spectra
/
```

`(in_filebase)` specifies the base name of the input history files, and `(out_filebase)` specifies the base name of the output NetCDF file. `(target_proc_num_tot)` should be set to the total number of MPI processes used in the original simulation. `(TARGET_LEVELS)` and `(level_units)` specify the vertical levels at which three-dimensional variables are analyzed. `(VARS)` specifies the target three-dimensional variables. Two-dimensional variables can also be analyzed by specifying `(VARS2D)`. `(SpectralAnalysisType)` can be set to `Uniform_Sampling` or `L2_Projection`.

The mesh information of the original output is specified in `[PARAM_SPECTRAL_ANALYSIS_MESH]`.

```
&PARAM_SPECTRAL_ANALYSIS_MESH
dom_xmin = -1.6D3, dom_xmax = 1.6D3, ; Domain range in the x-direction
dom_ymin = -1.6D3, dom_ymax = 1.6D3, ; Domain range in the y-direction
dom_zmin = 0.D0, dom_zmax = 1.6D3, ; Domain range in the z-direction
NprcX = 8, ; Number of MPI processes in the x-direction in the
; original run
NeX = 2, ; Number of elements in the x-direction in each MPI
; process
NprcY = 4, ; Number of MPI processes in the y-direction in the
; original run
NeY = 4, ; Number of elements in the y-direction in each MPI
; process
NeZ = 8, ; Number of elements in the z-direction
PolyOrder_h = 7, ; Polynomial order in the horizontal direction
PolyOrder_v = 7, ; Polynomial order in the vertical direction
/
```

The execution of `spectral_analysis` is as follows:

```
$ mpirun -n 4 ./spectral_analysis spectral_analysis.conf
```

In this example, `spectral_analysis` is executed with four MPI processes. The last argument specifies the configuration file. Note that the number of MPI processes used is constrained by the current implementation.

- The number of MPI processes used for `spectral_analysis` must be a divisor of `(target_proc_num_tot)` and must not exceed `(target_proc_num_tot)`.
- When `(SpectralAnalysisType)` is set to `Uniform_Sampling`, the number of MPI processes used for `spectral_analysis` should be the same as `(NprcY)`.

The output file is generated as `[out_filebase].nc`. The output contains wavenumber coordinates and spectra of the specified variables. When `(KinEnergyAnalysisFlag)` is set to `.true.`, horizontal and vertical kinetic energy spectra are also output.

Part 5

Detailed explanation for advanced use

Chapter 5.1

How to use user defined program

Bibliography

- 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(520):1485–1512, 1994. doi: 10.1002/qj.49712052004.
- Michael P. Byrne and Paul A. O’Gorman. Land–ocean warming contrast over a wide range of climates: Convective quasi-equilibrium theory and idealized simulations. *Journal of Climate*, 26(12):4000–4016, 2013.
- Dargan M. W. Frierson, Isaac M. Held, and Pablo Zurita-Gotor. A gray-radiation aquaplanet moist GCM. part i: Static stability and eddy scale. *Journal of the Atmospheric Sciences*, 63(10):2548–2566, 2006.
- Ruth Geen, Arnaud Czaja, and Joanna D. Haigh. The effects of increasing humidity on heat transport by extratropical waves. *Geophysical Research Letters*, 43(15):8314–8321, 2016. doi: 10.1002/2016GL070214.
- F. X. Giraldo, J. F. Kelly, and E. M. Constantinescu. Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (numa). *SIAM Journal on Scientific Computing*, 35(5):B1162–B1194, 2013. doi: 10.1137/120876034.
- Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007. doi: 10.1007/978-0-387-72067-8.
- Inmaculada Higuera and Teo Roldán. New third order low-storage ssp explicit runge–kutta methods. *Journal of Scientific Computing*, 79(3):1882–1906, 2019.
- Yuta Kawai and Hirofumi Tomita. Numerical Accuracy Necessary for Large-Eddy Simulation of Planetary Boundary Layer Turbulence Using the Discontinuous Galerkin Method. *Monthly Weather Review*, 151(6):1479 – 1508, 2023. doi: 10.1175/MWR-D-22-0245.1.
- Yuta Kawai and Hirofumi Tomita. Development of a high-order global dynamical core using the discontinuous Galerkin method for an atmospheric large-eddy simulation (LES) and proposal of test cases: SCALE-DG v0.8.0. *Geoscientific Model Development*, (18):725 – 762, 2025. doi: 10.5194/gmd-18-725-2025.
- Christopher A Kennedy and Mark H Carpenter. Additive Runge–Kutta schemes for convection–diffusion–reaction equations. *Applied numerical mathematics*, 44(1-2):139–181, 2003. doi: 10.1016/S0168-9274(02)00138-1.
- E. Kessler. On the distribution and continuity of water substance in atmospheric circulation. *Meteorological Monograph*, 10:1–84, 1969.
- David I. Ketcheson. Highly Efficient Strong Stability-Preserving Runge–Kutta Methods with Low-Storage Implementations. *SIAM Journal on Scientific Computing*, 30(4):2113–2136, 2008. doi: 10.1137/07070485X.
- Devin Light and Dale Durran. Preserving nonnegativity in discontinuous galerkin approximations to scalar transport via truncation and mass aware rescaling (tmar). *Monthly Weather Review*, 144(12):4771 – 4786, 2016. doi: 10.1175/MWR-D-16-0220.1.

- Douglas K Lilly. On the numerical simulation of buoyant convection. *Tellus*, 14(2):148–172, 1962. doi: 10.1111/j.2153-3490.1962.tb00128.x.
- Syukuro Manabe, Joseph Smagorinsky, and Robert F. Strickler. Simulated climatology of a general circulation model with a hydrologic cycle. *Monthly Weather Review*, 93(12):769–798, 1965. doi: 10.1175/1520-0493(1965)093<0769:SCOAGC>2.3.CO;2.
- Mikio Nakanishi and Hiroshi Niino. An improved mellor–yamada level-3 model: Its numerical stability and application to a regional prediction of advection fog. *Boundary-Layer Meteorology*, 119(2):397–407, 2006. doi: 10.1007/s10546-005-9030-8.
- S Nishizawa, H Yashiro, Y Sato, Yoshiaki Miyamoto, and H Tomita. Influence of grid aspect ratio on planetary boundary layer turbulence in large-eddy simulations. *Geoscientific Model Development*, 8(10):3393–3419, 2015. doi: 10.5194/gmd-8-3393-2015.
- Vladimirovich V. Rusanov. Calculation of Interaction of Non-Steady Shock Waves with Obstacles. *Journal of Computational and Mathematical Physics USSR*, 1:267, 1961. doi: 10.1016/0041-5553(62)90062-9.
- Yousuke Sato, Seiya Nishizawa, Hisashi Yashiro, Yoshiaki Miyamoto, Yoshiyuki Kajikawa, and Hirofumi 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):23, 2015.
- A. Scotti, C. Meneveau, and D. K. Lilly. Generalized smagorinsky model for anisotropic grids. *Physics of Fluids A*, 5:2306–2308, 1993.
- Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988. ISSN 0021-9991. doi: [https://doi.org/10.1016/0021-9991\(88\)90177-5](https://doi.org/10.1016/0021-9991(88)90177-5).
- Joseph Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly Weather Review*, 91(3):99–164, 1963. doi: 10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.
- H. Tomita. New microphysical schemes with five and six categories by diagnostic generation of cloud ice. *Journal of the Meteorological Society of Japan*, 86:121–142, 2008.
- Geoffrey K. Vallis, Greg Colyer, Ruth Geen, Edwin Gerber, Martin Jucker, Penelope Maher, Alexander Paterson, Marianne Pietschnig, James Penn, and Stephen I. Thomson. Isca, v1.0: a framework for the global modelling of the atmospheres of earth and other planets at varying levels of complexity. *Geoscientific Model Development*, 11(3):843–859, 2018. doi: 10.5194/gmd-11-843-2018.

Part 6

Appendix

SCALE-DG USERS GUIDE

Author & Editor

Yuta Kawai, Team SCALE User's guide working group



Copyright ©Yuta Kawai, Team SCALE, RIKEN R-CCS, 2018-2026. All rights reserved.