SCALE-DG USERS GUIDE

Version 1.0.0

Yuta Kawai, Team SCALE

October 3, 2025

SCALE-DG

Scalable Computing for Advanced Library and Environment - Discontinuous Galerkin

Contents

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 1.0.0. 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-1.0.0/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

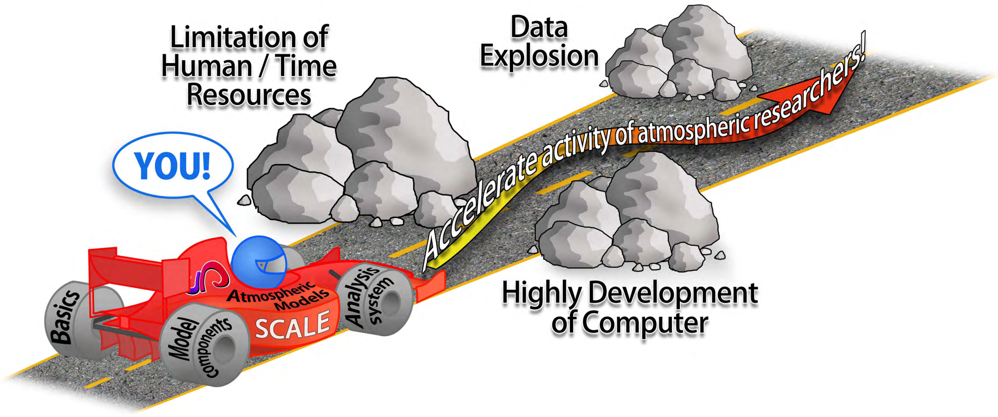
Limitation of Human / Time Resources
Data Explosion
YOU!
Accelerate activity of atmospheric researchers!
Highly Development of Computer
Basics
Model components
Atmospheric Models
SCALE
Analysis system

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

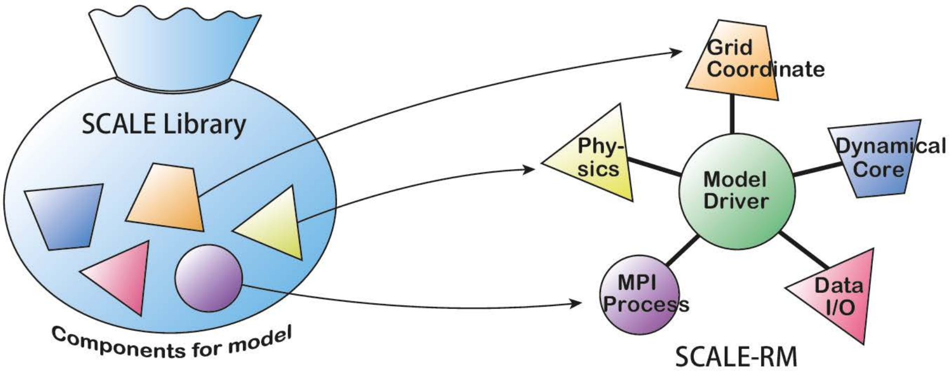

Figure 1.1.2: Relationship between the library SCALE and the model SCALE-RM (cite: Fig. 1.1.2 of SCALE users guide)

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 mange 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.

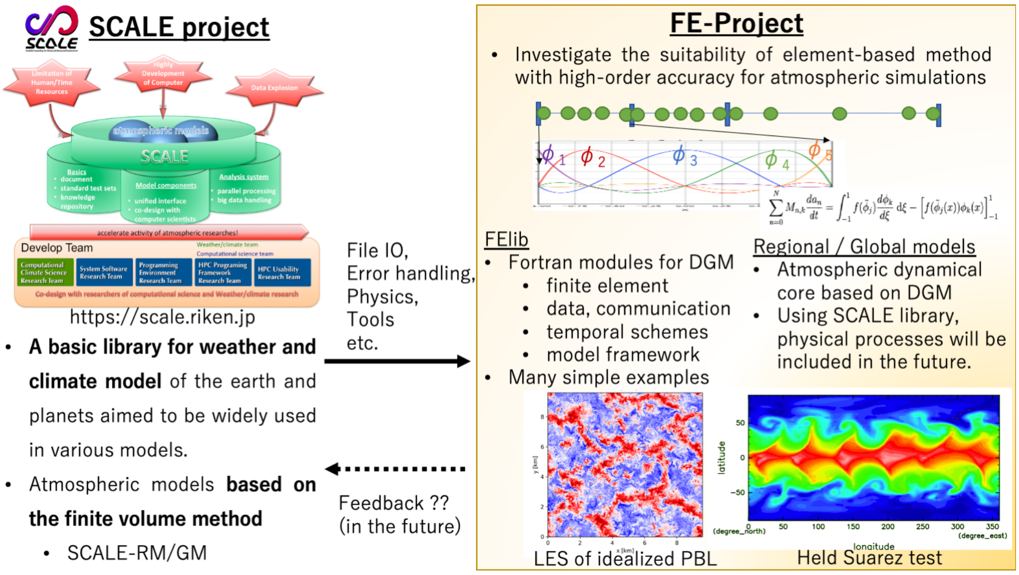
SCALE project
Develop Team
https://scale.riken.jp
A basic library for weather and climate model of the earth and planets aimed to be widely used in various models.
Atmospheric models based on the finite volume method
SCALE-RM/GM
File IO,
Error handling,
Physics,
Tools
etc.
Feedback ??
(in the future)
FE-Project
Investigate the suitability of element-based method with high-order accuracy for atmospheric simulations
FElib
Fortran modules for DGM
finite element
data, communication
temporal schemes
model framework
Many simple examples
Regional / Global models
Atmospheric dynamical core based on DGM
Using SCALE library, physical processes will be included in the future.
LES of idealized PBL
Held Suarez test

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 an ideal experiment

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)

- Cloud microphysics: selectable from among the following
  - 3-class 1 moment bulk scheme (Kessler, 1969)
  - 6-class 1 moment bulk scheme (Tomita, 2008)

- Surface models
  - 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.

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.

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-1.0.0/` can be seen when the tarball file of the source code is extracted.

```
$ tar -zxvf FE-Project-1.0.0.tar.gz
$ ls ./FE-Project-1.0.0/
```

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-1.0.0/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-ompi" (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-ompi
	Intel OneAPI	Intel MPI	Makedef.Linux64-llvm-intel-impi
Mac OS	GNU	Open MPI	Makedef.MacOSX-gnu-ompi
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_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 -lnetcdff -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-1.0.0/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-1.0.0/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-1.0.0/lib` directory. To compile regrid_tool , execute the following command after the compilation of SCALE-DG :

```
$  cd FE-Project-1.0.0/model/atm_nonhydro3d/util/regrid_tool/
$  make
```

If the compilation is succeeded, the executable binary file is generated under the `FE-Project-1.0.0/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-1.0.0/model/atm_nonhydro3d/test/case/mountain_wave/scale-rm
FE-Project-1.0.0/model/atm_nonhydro3d/test/case/mountain_wave/scale-rm_init
FE-Project-1.0.0/model/atm_nonhydro3d/test/case/mountain_wave/scale-rm_pp
FE-Project-1.0.0/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 × 8 DOF ($p = 7$): x-direction, 1 element × 8 DOF ($p = 7$): y-direction	
Vertical DOF	8 elements × 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.pe00000[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.pe00000[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.pe00000*.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.

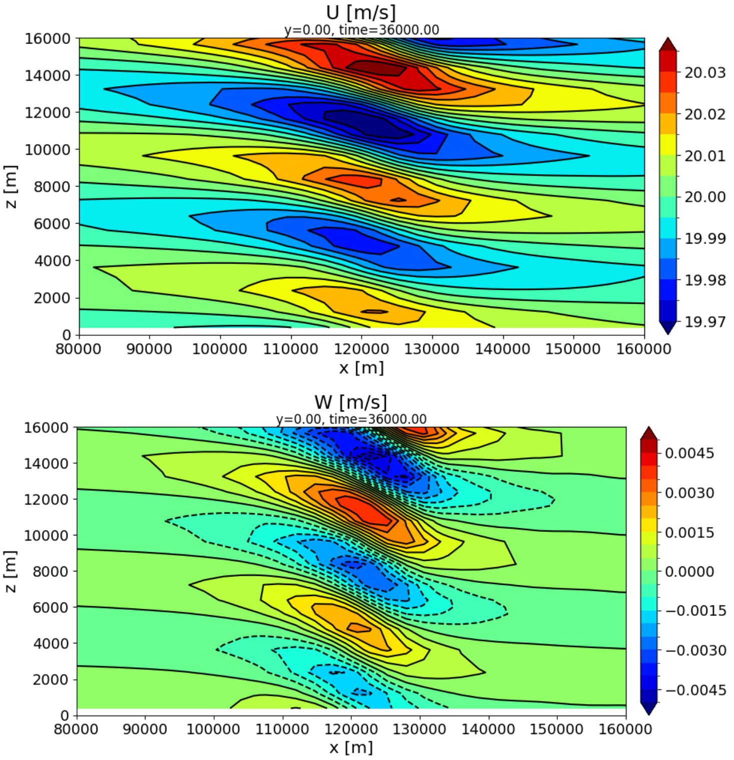

Figure 3.1.1: The vertical section of horizontal and vertical winds after $t = 10$ hours

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, various idealized experimental settings have been prepared in the directory "`model/atm_nonhydro3d/test/case`." 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.

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.D0,             ; Time interval of history output
 FILE_HISTORY_DEFAULT_TUNIT = "SEC",                   ; Unit of DEFAULT_TINTERVAL
 FILE_HISTORY_DEFAULT_TSTATS_OP = "none",              ; Statistics operation
 FILE_HISTORY_DEFAULT_DATATYPE = "REAL4",              ; Output data type: REAL4 or REAL8
 FILE_HISTORY_OUTPUT_STEP0 = .true.,                   ; Output data at initial time (t=0)?
 FILE_HISTORY_OUTPUT_WAIT = 0.D0,                      ; Time to suppress output
 FILE_HISTORY_OUTPUT_WAIT_TUNIT = "SEC",               ; Unit of OUTPUT_WAIT
 FILE_HISTORY_OUTPUT_SWITCH_TINTERVAL = -1.D0,         ; Time interval to switch the file
 FILE_HISTORY_OUTPUT_SWITCH_TUNIT = "SEC",             ; Unit of OUTPUT_SWITCH_TINTERVAL
 FILE_HISTORY_ERROR_PUTMISS = .true.,                  ; Check missing preparation of the
                                                       data?
/
```

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_STEP0) is set to .true., the variables at the time before time integration (initial state) are output to history file. You can suppress history output during the time in the simulation defined by (FILE_HISTORY_OUTPUT_WAIT) and (FILE_HISTORY_OUTPUT_WAIT_TUNIT). If the value is negative, no suppression occurs. (FILE_HISTORY_OUTPUT_SWITCH_TINTERVAL) is time interval of switching output file and its unit is defined by (FILE_HISTORY_OUTPUT_SWITCH_TUNIT). If the value is negative only one file per process is used for history output. If this option is enabled, the time label is added to the file name.

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.D0,           (OPTION) same as FILE_HISTORY_DEFAULT_TINTERVAL
 TUNIT = "SEC",                (OPTION) same as FILE_HISTORY_DEFAULT_TUNIT
 TSTATS_OP = "mean",           (OPTION) same as FILE_HISTORY_DEFAULT_TSTATS_OP
 DATATYPE = "REAL4",           (OPTION) same as FILE_HISTORY_DEFAULT_DATATYPE
/
```

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

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

```
&HISTORY_ITEM NAME="T" /
&HISTORY_ITEM NAME="U" /
&HISTORY_ITEM NAME="V" /
&HISTORY_ITEM NAME="RAIN", TINTERVAL=600.D0, 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-
                                      tion4.3.2)
/
```

(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 $\Delta t/\alpha_m$.

The switch of modal filtering exists in [PARAM_ATMOS_DYN] and the parameters are specified in [PARAM_ATMOS_DYN_MODALFILTER] as

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

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

We set the order p_m, and decay coefficient α_m such that the strength of filter should ensure numerical stability while being as weak as possible.

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,             ; f0
CORIOLIS_beta = 0D0,           ; β0
CORIOLIS_y0 = 0D0,             ; y0
/
```

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.

Chapter 4.4

Setting the physical process

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 outputing 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 gird when output-
                                 ing 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 regriding 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_*].

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.

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 Higueras 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.

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.

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.

Part 6

Appendix

SCALE-DG USERS GUIDE

Author & Editor

Yuta Kawai, Team SCALE User's guide working group