SPRING

Version 2.4.2

User's Guide

November 2025

Akira Takeshima, Chiba University

Index

1 Introduction

(To be updated)

2 SPRING package and instruction

2.1 Contents of the package

SPRING consists of several kinds of programs:

(1) Remapping or some related processes such as merging remapping tables or calculating grid area

(2) Extended functions which are mainly for ILS users (especially for advanced use of ILS) including external programs

(3) Scripts for specific aims which are also mainly for ILS

(4) Visualizing meshes or field data

2.2 Quick instruction

(1) Get the package

Get the package of the software from the website or GitHub and sample data from the website. You can get them from Yoshimura lab's server isotope3 if you have an account.

User's guide (this document)

Website: http://hydro.iis.u-tokyo.ac.jp/~akira/pub/SPRING/doc/SPRING_doc.v02.04.02_251111.pdf

Source code

Website: http://hydro.iis.u-tokyo.ac.jp/~akira/pub/SPRING/pkg/SPRING.v02.04.02.tar.gz

GitHub: https://github.com/AkiraTAKESHIMA/SPRING/

isotope3 server: /data7/akira/SPRING/pkg/SPRING.v02.04.02.tar.gz

Sample data

Website: http://hydro.iis.u-tokyo.ac.jp/~akira/pub/SPRING/dat/SPRING_data.v02.04.02.tar.gz

isotope3: /data7/akira/SPRING/dat/SPRING_data.v02.04.02.tar.gz

Environments

Source codes and scripts are written in Fortran, C and Python. Main programs are written in Fortran and C. Python is used for visualization and running the scripts in run/.

Programs and scripts have been tested under the following environments.

(i) Debian GNU/Linux 11 (bullseye) (laptop, Windows Subsystem Linux 2)

- Fortran compiler

gfortran version 10.2.1

- C compiler

gcc version 10.2.1

- Python

Python version 3.9.2

- Python libraries

os

sys

json version 2.0.9

numpy version 1.19.5

matplotlib version 3.5.1

cartopy version 0.21.1

(ii) CentOS Linux 7 (Core)

- Fortran compiler

ifort version 19.0.3.199

- C compiler

icc version 19.0.3.199

- Python

Python version 3.8.0

- Python libraries

os

sys

json version 2.0.9

numpy version 1.24.4

matplotlib version 3.7.4

cartopy version 0.21.1

(iii) Ubuntu 18.04.5 LTS

- Fortran compiler

ifort version 2021.3.0

- C compiler

icc version 2021.3.0

- Python

Python version 3.9.6

- Python libraries

os

sys

json version 2.0.9

numpy version 1.26.4

matplotlib version 3.9.2

cartopy version 0.23.0

(2) Unzip packages and make symbolic links

```
$ ls
SPRING.v02.04.02.tar.gz  SPRING_data.v02.04.02.tar.gz
$ tar xvzf SPRING.v02.04.02.tar.gz
$ tar xvzf SPRING_data.v02.04.02.tar.gz
$ cd SPRING.v02.04.02
$ ls
adm/  README.md  run/  set/  src/
$ ln -s ../SPRING_data.v02.04.02 dat
$ ls
adm/  'dat/  README.md  run/  set/  src/
```

- ✧ A single quote before a path (e.g., ‘dat/) indicates that it is a symbolic link.
- ✧ The absolute path of the top directory of SPRING (SPRING.v02.04.02/) is written as $(SPRING) hereafter.

(3) Compile the source codes

Prepare a Mkinclude file and compile the source codes.

```
$ cd src
### select the Mkinclude file suitable for your environment
$ cp ../adm/Mkinclude.intel Mkinclude
### edit it if necessary
$ vim Mkinclude
### Compile and install
$ make clean && make && make install
$ cd ../
$ ls
adm/  'dat/  bin/  README.md  run/  set/  src/
$ ls bin/*
bin/ext:
cpl_aogcm-ils_define_mat.exe  cpl_aogcm-ils_make_rt_for_ogcm.exe  make_cmf_mat.exe

bin/std:
make_grid_data.exe  merge_remapping_tables.exe  rasterize.exe  remap.exe
mkfig_remap.py
```

[Note]

Compiling error may occur on Mac computers:

```
lib/Makefile
$(TARGET): $(OBJ)
	$(RM) $(RMFLAGS) $@
```

$(AR) cqT $@ $^
echo "create $@\naddlib $@\nsave\nend" | ar –M ← ERROR

(4) Test run

Test the program with a simple remapping from T85 latlon mesh to 1-degree COCO mesh as follows:

```
$ cd $(SPRING)
$ ./bin/std/remap.exe set/remap/T85_to_COCO-1deg.conf
```

Then visualize the results as follows:

```
$ python bin/std/mkfig_remap.py set/remap/T85_to_COCO-1deg.conf
```

The figure in Figure 2-1 will be generated.

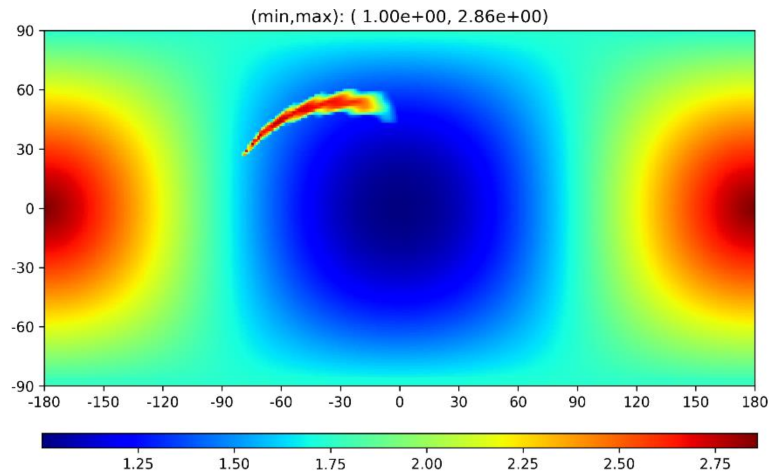

Figure 2-1 Grid values of COCO 1 degree-mesh remapped from T85 mesh.

3 Definitions and equations

3.1 Spherical coordinates and cartesian coordinates

Spherical coordinates are represented by longitude and latitude. Cartesian coordinates are represented by x-, y- and z-coordinates, and the origin is at the center of the Earth.

In SPRING, relationship between spherical coordinates and cartesian coordinates is as follows.

Suppose that the Earth is a perfect sphere having a diameter of 1.0. (x=1, y=0, z=0) is identical to (lon=0, lat=0) and (x=0, y=1, z=0) is identical to (lon=90, lat=0), and thus (x=0, y=0, z=1) is identical to (lat=90) (Figure 3-1).

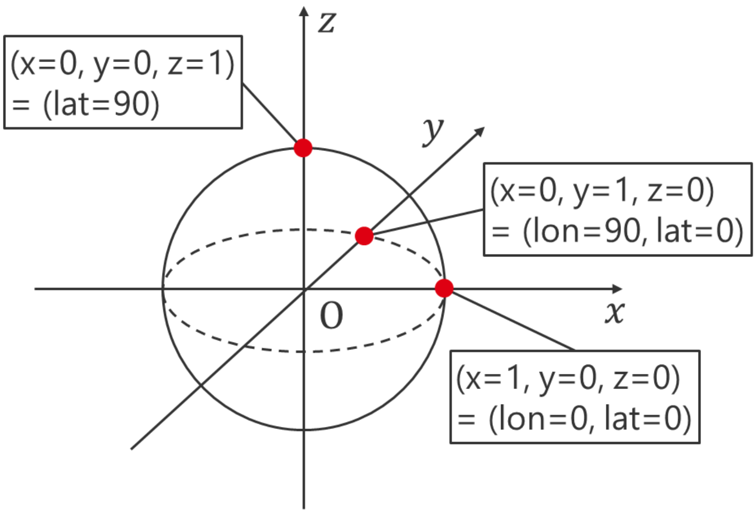

Figure 3-1 Cartesian coordinate in SPRING.

3.2 Remapping coefficients

In the normal remapping, 1st order conservative remapping is applied.

$$\boldsymbol{x}_j^T = \sum_{i=1}^{N} \boldsymbol{c}_{i,j} \boldsymbol{x}_i^S, \tag{3-1}$$

where x is grid value and c is remapping coefficient. Superscripts S and T indicate source mesh S and target mesh T, respectively, and subscripts i and j indicate the grid index in meshes S and T, respectively. N is the number of grids in mesh S.

In this method, remapping coefficients are calculated from ratio of the intersection area to the grid area. For exchanging fluxes such as latent heat flux or precipitation, remapping coefficients are calculated by:

$$c_{i,j} = \frac{a_{i,j}}{A_j^T}, \tag{3-2}$$

where a is intersection area and A is grid area. For exchanging mass or volume such as amount of water or other materials, coefficients are calculated by:

$$c_{i,j} = \frac{a_{i,j}}{A_i^S}. \tag{3-3}$$

For exchanging fluxes such as latent heat flux or precipitation between meshes, grid area of the target mesh will be denominator of equation remapping coefficients are calculated

Users can specify grid weight. If grid weights $\boldsymbol{W}^S$ for mesh S were specified, remapping coefficients for flux exchange are calculated by

$$c_{i,j} = \frac{a_{i,j}}{A_j^T} \; W_i^S. \tag{3-4}$$

If grid weights $\boldsymbol{W}^T$ for mesh T were specified, they do not affect remapping coefficients since they are multiplied to both A_j^T and $a_{i,j}$ as follows:

$$c_{i,j} = \frac{a_{i,j} W_j^T}{A_j^T W_j^T} \; W_i^S = \frac{a_{i,j}}{A_j^T} \; W_i^S.$$

3.3 Meshes

SPRING classifies meshes into three categories: latlon mesh, polygon mesh and rater mesh.

Latlon mesh is the most standard type of mesh, that divides the sphere by meridians and parallels. Polygon mesh consists of polygons such as triangles, rectangles, pentagons, hexagons, or polygons with more complex shapes. Latlon mesh is a special case of polygon mesh, but it is usually distinguished from polygon mesh due to its simplicity. Raster mesh represents each grid with an assembly of small pixels.

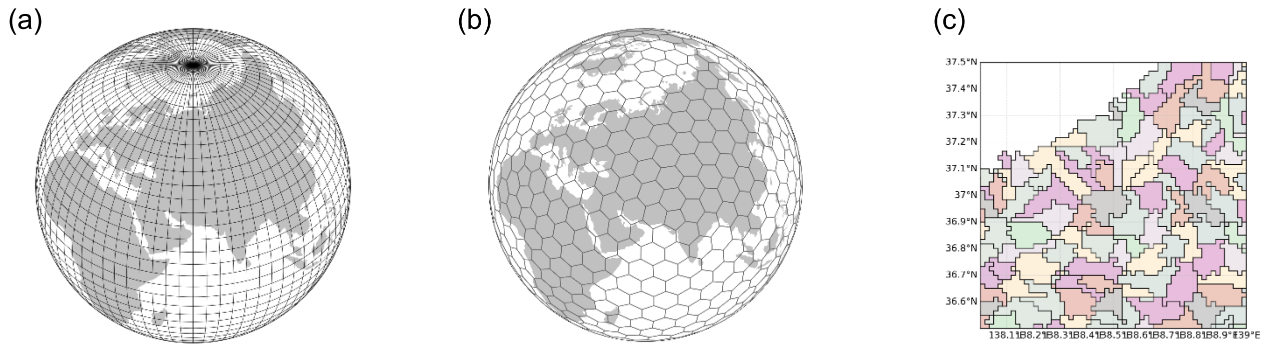

Figure 3-2 Examples of three types of meshes. (a) latlon mesh, (b) polygon mesh and (c) raster mesh.

4 Format of configuration files

SPRING uses configuration files written in a unique format.

4.1 Structure

Configuration files for SPRING consist of the path of a report file followed by blocks, and each block consists of pairs of keys and values. Path of a report file is specified with the key *path_report*, and the pair of a key and a value are divided by a colon. Each block starts with its name enclosed in square brackets and ends with *[end]*. Each pair of a key and a value or a set of values in blocks are divided by a colon, similarly to the path of a report file. Lines which start with a hash sign "#" and words after this sign are commented out.

Following sample shows the structure of configuration files, where changeable variables are written in Italic and commented out lines or phrases are gray-colored.

```
path_report: 'path_of_report_file'
# Any comment does not affect the performance of programs

[block_name]
  key: value  # Comments can be added to lines
  key: value
  …
[end]

[block_name]
  key: value
  key: value
  …
[end]
```

4.2 Data types

Data type of each input/output data has a default value and can be specified explicitly. Usually it is specified as follows:

```
f_lon_bound: ".../dat/T85/T85_lon_bound.bin", dble, 1, little
```

In this example, *dble* is the data type of *f_lon_bound*. Its format is described in detail below (4.4). Usable data types are listed in Table 4-1.

Table 4-1 Values of data type.

Value	Description

int1	1-byte integer
int2	2-byte integer
int4	4-byte integer
int8	8-byte integer
real	4-byte float
dble	8-byte float
log1	1-byte boolean
log2	2-byte boolean
log4	4-byte boolean
log8	8-byte boolean

* float = floating-point number

4.3 Endian

Endian of input/output data can be specified explicitly. Details are described in the section *file* below.

4.4 Types of setting variables

Each value in a configuration file has a data type such as integer or character. SPRING configuration files use following 8 data types: integer, float, boolean, character, path, tuple and file.

- integer

An integer.

Example(s):

```
nx: 720
```

- float

An 8-byte floating-point number (called “double precision” in Fortran language). Format of this type of variables have several options; any of “1.0”, “1e0”, “1.e0”, “1d0”, “1.d0” and “1” indicates completely same value, 1.0. Capital letters and small letters are not distinguished.

Example(s):

```
west: -180.d0
east: 1.8e2
```

- boolean

A boolean value (called “logical” in Fortran language). “True”, “T” and “.true.” mean true and “False”, “F” and “.false.” mean false. Capital letters and small letters are not distinguished.

Example(s):

```
is_south_to_north: false
```

- string

A string value (called “character” in Fortran language) without space. Whether or not the value is enclosed in quotes does not matter.

Example(s):

```
name: global_30min

coord_unit: degree
```

- path

A path of a file. The variable must be enclosed in single quotes or double quotes.

Example(s):

```
path_report: “../out/remapping/T85_to_COCO-1deg/report.txt”
```

Base directory of relative paths is the current directory in default. Most blocks have a variable “dir”, by which the base directory can be changed. Paths written as absolute path are not affected. An example is shown below.

```
dir: “../dat/T85”
f_lon_bound: “T85_lon_bound.bin”
f_lat_bound: “T85_lat_bound.bin”
dir: “”
fin_grdidx: ../dat/T85/grdidx.bin
fin_grdara: “/data7/user/SPRING/dat/T85/grdara.bin”
```

In this case, the path of each file is as follows:

```
f_lon_bound: “../dat/T85/T85_lon_bound.bin”
f_lat_bound: “../dat/T85/T85_lat_bound.bin”
fin_grdidx: “../dat/T85/grdidx.bin”
fin_grdara: “/data7/user/SPRING/dat/T85/grdara.bin”
```

- tuple

A sequence of variables arranged in a specific order and divided by commas.

Example(s):

```
in_grid_sz: 720, 360
```

- file

A set of variables that provides information on a file in which data has been or will be recorded. For plain binary format-files, the following values are specified with the variable of this data type: (1) path, (2) data type, (3) record number and (4) endian. If all of these four items of information are specified, they can be written as follows:
key: *"path"*, *data_type*, *record_number*, *endian*

The variables other than *path* can be omitted if they are same to the default values, that are predefined for each file, as follows:
key: *"path"*, *data_type*

Also, roles of variables can be explicitly specified using correspondent keywords shown in Table 4-2.

Table 4-2 Variables and correspondent keywords.

Variable	Keyword	Data type
path	path	path
data_type	dtype	integer
record_number	rec	integer
endian	endian	string. {"little", "big"}

Data type can take any of values in Table 4-1 (some of them are invalid depending on the data).
Endian can take either "little" or "big". "little" is for little endian and "big" is for big endian.

The order of variables can be changed using keywords. Following examples are identical:
key: path=*"path"*, dtype=*data_type*, rec=*record_number*, endian=*endian*
key: *"path"*, *data_type*, rec=*record_number*, endian=*endian*
key: rec=*record_number*, endian=*endian,* path=*"path"*, dtype=*data_type*

If *data_type* is same to the default value, it can be omitted, and the settings above can be written as:
key: path=*"path"*, rec=*record_number*, endian=*endian*
key: *"path"*, rec=*record_number*, endian=*endian*
key: rec=*record_number*, endian=*endian,* path=*"path"*

In most cases, default values of variables of files are as follows:
data_type = *int4* or *dble*, *record_number* = 1, *endian* = *little*.

Example(s):

```
f_lon_bound: "../dat/T85/T85_lon_bound.bin"

f_lon_vertex: "../dat/COCO/1deg/COCO_lon.bin", dble, 1, little
```

5 Making a remapping table and remapping grid values

5.1 Setting variables

Configuration files consist of:

- Path of report file
- Two of mesh blocks in *mesh_latlon*, *mesh_polygon* and *mesh_raster* (duplication allowed)
- Block *remapping*
- Block *options* (optional)
- Block *figures* (optional)

The first mesh block is considered the source mesh, and the second mesh block is considered the target mesh. The block “remapping” defines parameters related to the remapping table such as output files of the remapping table, options for calculation, modification or thresholds of interpolation coefficients. This block is also used to remap mesh data.

Block *mesh_latlon*

Table 5-1 Setting variables for block *mesh_latlon*

Variable	Data type	Description	Default value
name	string	Mesh name	‘source’ or ‘target’
nx	integer	The number of grids in longitudinal direction	
ny	integer	The number of grids in latitudinal direction	
west	float	West edge of the mesh	
east	float	East edge of the mesh	
south	float	South edge of the mesh	
north	float	North edge of the mesh	
is_south_to_north	boolean	Grid data is arranged from south to north	True
dir	path	Parent directory of the files below	‘’
f_lon_bound	file	Input file of longitudes of latitude-direction grid lines	‘’, dble, 1, little
f_lat_bound	file	Input file of latitudes of longitude-direction grid lines	‘’, dble, 1, little
coord_unit	string	Unit of coordinates in *f_lon_bound* and *f_lat_bound*. {‘degree’, ‘radian’}	degree
idx_bgn	integer	Initial number of grid index automatically set	1
fin_grdidx	file	Input file of grid indices	‘’, int4, 1, little
fin_grdara	file	Input file of grid areas	‘’, dble, 1, little
fin_grdwgt	file	Input file of grid weights	‘’, dble, 1, little
in_grid_sz	tuple	Size of input grid data	*nx, ny*
in_grid_lb	tuple	Lower bounds of grid data to use	1, 1
in_grid_ub	tuple	Upper bounds of grid data to use	*nx, ny*
idx_miss	integer	Missing value of grid index	-9999
ara_miss	float	Missing value of grid area	-1e20
wgt_miss	float	Missing value of grid weight	-1e20
val_miss	float	Missing value of grid value	-1e20

Details

[Utilities]

name
User-defined mesh name. It is used in the log messages and report file but does not affect the result of calculation. ‘source’ for the source mesh and ‘target’ for the target mesh.

[Coordinates]

nx, west, east, f_lon_bound
Coordinates of latitude-direction grid lines are defined by either the pair of (*west*, *east*) or *f_lon_bound*. If the grid size is uniform in longitude-direction, coordinates of the grid lines can be defined by *west*, *east* and *nx*. If not, use *f_lon_bound*.
f_lon_bound must have coordinates from the west edge to the east edge, thus *nx*+1 values. If the mesh covers the whole area in the longitudinal direction, the 1st element and the (*nx*+1)th element will be identical.
Coordinates, namely *west*, *east* and values in *f_lon_bound*, must be in the range of [-180, 180] or [0, 360].

ny, south, north, f_lat_bound
Similarly to the latitude-direction grid lines, coordinates of longitude-direction lines are defined by either the pair of (*south*, *north*) or *f_lat_bound*. If the grid size is uniform in latitude-direction, coordinates of the grid lines can be defined by *south*, *north* and *ny*. If not, use *f_lat_bound*.
f_lat_bound must have coordinates from the south edge to the north edge (north to south also works), thus *ny*+1 values.

[Direction of y-axis]

is_south_to_north
If *True*, the positive direction of the y-axis (the axis in the latitudinal direction) of the mesh is from south to north, and vice versa.
Grid indices automatically set increase in the positive direction of y-axis (Figure エラー**!** 参照元が見つかりません。.2).
It is assumed that each grid data is recorded in the same order as this grid indices. When y-axis is from south to north, recorded data must be also arranged from south to north, and vice versa.

[Grid data]

Grid data usually has a shape of (*nx, ny*).

idx_bgn

Grid indices start from this value when they are automatically set (= *fin_grdidx* were not specified). Figure 5-1 is the case that *idx_bgn* is 1 (default).

fin_grdidx

Input file of grid indices.

If this is not specified, grid indices are determined automatically. The order of grid indices relies on the direction of y-axis, which is specified by *is_south_to_north*. The indices start from the value of *idx_bgn* (1 in default).

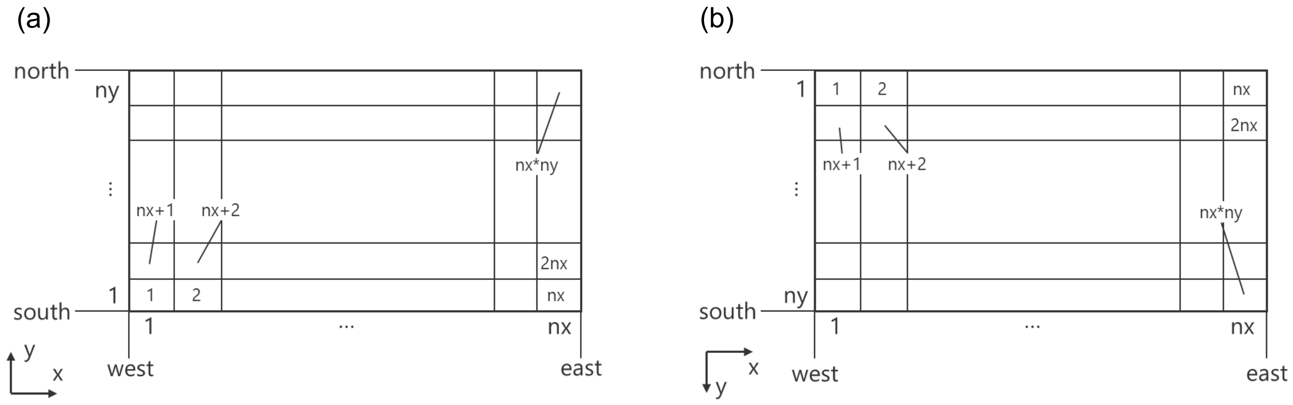

Figure 5-1 Direction of y-axis and grid indices automatically set. (a) The case when y-axis is from south to north and (b) the opposite case.

fin_grdara, fin_grdwgt

These variables are used to give grid weights. They cannot be specified at the same time.

If *fin_grdara* was specified, grid weights $\boldsymbol{w} = \{w_i, i = 1, \dots, N\}$ are calculated by

$$w_i = \frac{A_i^{in}}{A_i^{true}} \ (i = 1, \dots, N),$$

where A_i^{in} is grid area given via *fin_grdara*, A_i^{true} is grid area calculated from its shape and N is the number of grids. *fin_grdwgt* directly gives $\boldsymbol{w}$.

[Using part of data (advanced)]

in_grid_sz, in_grid_lb, in_grid_ub

These variables are used when the size of input data and the number of grids of the mesh do not match. in_grid_sz gives the size of input grid data, in_grid_lb and in_grid_ub gives lower and upper bounds of the grid data to use, respectively.

Data in the range (*lb1*:*ub1*, *lb2*:*ub2*) in the input data (*sz1*, *sz2*) is considered to be the mesh defined here. *sz1* and *sz2* are the first and the second values of *in_grid_sz*, respectively, *lb1* and *lb2* are the first and the second values of *in_grid_lb*, respectively, and *ub1* and *ub2* are the first and the second values of *in_grid_ub*, respectively. Figure 5-2 shows the relations among mesh, input data and these values.

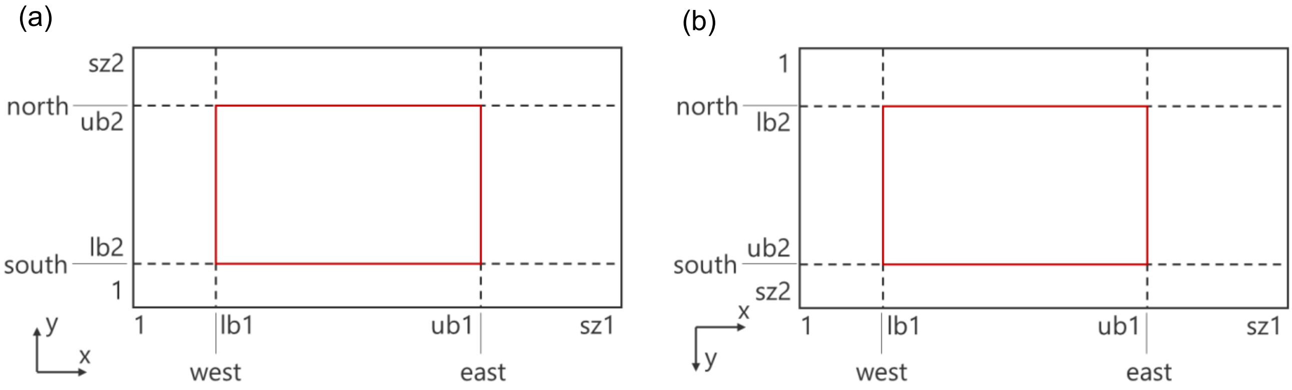

Figure 5-2 Relations of mesh, in_grid_sz, in_grid_lb and in_grid_ub. sz1 and sz2 are the first and the second values of in_grid_sz, respectively. (a) the case when the y-axis is from south to north and (b) the opposite case. Black solid line is a 2 dimension-array in the file and red line is the range correspondent to the mesh.

Block *mesh_polygon*

Table 5-2 Setting variables for block *mesh_polygon*

Variable	Data type	Description	Default value
name	string	Mesh name	'source' or 'target'
np	integer	The maximum number of grid vertices	
nij	integer	The number of grids	
dir	path	Parent directory of the files below	''
f_lon_vertex	file	Input file of longitudes of vertices	'', dble, 1, little
f_lat_vertex	file	Input file of latitudes of vertices	'', dble, 1, little
f_x_vertex	file	Input file of x coordinates of vertices	'', dble, 1, little
f_y_vertex	file	Input file of y coordinates of vertices	'', dble, 1, little
f_z_vertex	file	Input file of z coordinates of vertices	'', dble, 1, little
coord_unit	string	Unit of coordinates. {'degree', 'radian'} for spherical and {'m', 'km'} for cartesian	'degree' for spherical and 'm' for cartesian
coord_miss	float	Missing value of coordinates	-1e20
f_arctyp	file	Input file of types of grid sides	'', int4, 1, little
arc_parallel	boolean	Grid sides having vertices with the same latitude are considered as horizontal	False
idx_bgn	integer	Initial number of grid index automatically set	1
fin_grdidx	file	Input file of grid indices	'', int4, 1, little
fin_grdara	file	Input file of grid areas	'', dble, 1, little
fin_grdwgt	file	Input file of grid weights	'', dble, 1, little
in_grid_sz	tuple	Size of input grid data	*np, nij*

in_grid_lb	tuple	Lower bounds of grid data to use	1, 1
in_grid_ub	tuple	Upper bounds of grid data to use	*np, nij*
idx_miss	integer	Missing value of grid index	-9999
ara_miss	float	Missing value of grid area	-1e20
wgt_miss	float	Missing value of grid weight	-1e20
val_miss	float	Missing value of grid value	-1e20

Details

[Utilities]

name

User-defined mesh name. It is used in the log messages and report file but does not affect the result of calculation. *'source'* for source mesh and *'target'* for target mesh in default.

[Shapes of grids]

np

The number of vertices of grids are not necessarily uniform. To input data of vertices and sides with plain binary format files, arrays that have a shape of (*np, nij*) are required. If the number of vertices *n* of a grid is smaller than *np*, from *n*th to *np*th elements are filled by missing values.

f_lon_vertex, f_lat_vertex, f_x_vertex, f_y_vertex, f_z_vertex

Coordinates of grid vertices are represented in spherical coordinates or Cartesian coordinates. *f_lon_vertex* and *f_lat_vertex* are used for the former case and *f_x_vertex*, *f_y_vertex* and *f_z_vertex* are used for the latter case.

Basically, each data has a shape of (*np, nij*). Element (*p*, *ij*) is the coordinate of the *p*th vertex of the *ij*th grid.

coord_unit

If the coordinates of grid vertices were given as spherical coordinates, the value of coord_unit must be 'degree' or 'radian', and 'degree' in default. If cartesian coordinates, the value must be 'm' or 'km', and 'm' in default.

f_arctyp

Basically, the data has a shape of (*np*, *nij*). Element (*p*, *ij*) is the type of the *p*th grid side, which connects vertices P_p and P_{p+1} if $p < m$ and P_n and P_1 if $p = m$, where m is the number of vertices in the ijth grid (Figure 5-3).

1 means great arc and 2 means small arc.

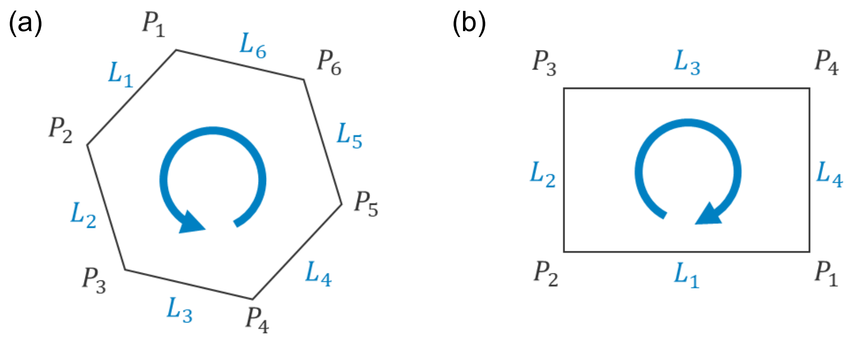

Figure 5-3 Relationship between vertices and sides. Vertices are arranged in the direction of (a) anticlockwise rotation and (b) clockwise rotation.

arc_parallel: *boolean, default: False*

If True, grid sides that have edges of same latitudes are dealt with as small arcs (= part of intersections of horizontal planes and the Earth). This variable is inactivated when *f_arctyp* was specified.

[Grid data]

Grid data usually has a shape of (*nij*).

fin_grdara, fin_grdwgt

These variables are used to give grid weights. They cannot be specified at the same time.

If *fin_grdara* was specified, grid weights $\boldsymbol{W} = \{w_1, \dots, w_N\}$ are calculated by

$$w_i = \frac{A_i^{in}}{A_i^{true}} \ (i = 1, \dots, N),$$

where A_i^{in} is grid area given via *fin_grdara*, A_i^{true} is grid area calculated from its shape and N is the number of grids. *fin_grdwgt* directly gives $\boldsymbol{W}$.

Block *mesh_raster*

Table 5-3 Setting variables for block *mesh_raster*

Variable	Data type	Description	Default value
name	string	Mesh name	‘source’ or ‘target’
nx	integer	The number of pixels in the longitudinal direction	
ny	integer	The number of pixels in the latitudinal direction	
west	float	West edge of the mesh	
east	float	East edge of the mesh	
south	float	South edge of the mesh	
north	float	North edge of the mesh	
is_south_to_north	boolean	Raster data is arranged from south to north	True
xi	integer	Lower bound of the valid range in the longitudinal direction in the mesh	1

xf	integer	Upper bound of the valid range in the longitudinal direction in the mesh	*nx*
yi	integer	Lower bound of the valid range in the latitudinal direction in the mesh	1
yf	integer	Upper bound of the valid range in the latitudinal direction in the mesh	*ny*
dir	path	Parent directory of the files below	‘’
fin_rstidx	file	Input file of raster indices	‘’, int4, 1, little
fin_rstara	file	Input file of raster areas	‘’, dble, 1, little
fin_rstwgt	file	Input file of raster weights	‘’, dble, 1, little
in_raster_sz	tuple	Size of input raster data	*nx, ny*
in_raster_lb	tuple	Lower bounds of raster data to use	1, 1
in_raster_ub	tuple	Upper bounds of raster data to use	*nx, ny*
fin_grdidx	file	Input file of grid indices	‘’, int4, 1, little
fin_grdara	file	Input file of grid areas	‘’, dble, 1, little
fin_grdwgt	file	Input file of grid weights	‘’, dble, 1, little
in_grid_sz	tuple	Size of input grid data	
in_grid_lb	tuple	Lower bounds of grid data to use	1, 1
in_grid_ub	tuple	Upper bounds of grid data to use	*in_grid_sz*
idx_condition	string	Condition that raster indices and grid indices must be fulfilled. {‘match’, ‘grid_in_raster’, ‘raster_in_grid’, ‘none’}	match
idx_miss	integer	Missing value of grid index	-9999
ara_miss	float	Missing value of grid area	-1e20
wgt_miss	float	Missing value of grid weight	-1e20
val_miss	float	Missing value of grid value	-1e20

Details

[Utilities]

name

User-defined mesh name. It is used in the log messages and report file but does not affect the result of calculation. ‘source’ for the source mesh and ‘target’ for the target mesh.

[Direction of y-axis]

is_south_to_north

If *True*, the positive direction of the y-axis (the axis in the latitudinal direction) of the mesh is from south to north, and vice versa.

It is assumed that each raster data is recorded in the same order as this grid indices. When y-axis is from south to north, recorded data must be also arranged from south to north, and vice versa.

[Raster data and grid data (advanced)]

fin_rstara, fin_rstwgt

Input file of pixel areas and weights, respectively.

fin_grdara, fin_grdwgt
Input file of grid areas and grid weights, respectively.

in_grid_sz
Tuple of two integers that represent shape of input grid data.

idx_condition
The condition that raster index data and grid index data must fulfill. K^{grid}, a set of grid indices made from grid data, and K^{raster}, a set of grid indices made from raster data, are identical in most cases when the globe is the subject of calculation. However, in some cases, especially when the calculation domain was limited, they are not necessarily identical. *idx_condition* can relax the condition for their relationship. Its values and the conditions to be fulfilled are as follows:

match (default): $K^{grid} = K^{raster}$
grid_in_raster: $K^{grid} \subseteq K^{raster}$
raster_in_grid: $K^{raster} \subseteq K^{grid}$
none: no condition

[Clipping the region and using part of data (advanced)]

xi, xf, yi, yf
The area defined by *west*, *east*, *south* and *north* is divided into *nx* columns and *ny* rows, and only the area from *xi*th to *xf*th columns and from *yi*th to *yf*th rows is considered in the process of remapping (Figure 5-4).

For example, following two settings define the same mesh.
(1) west=-180, east=180, south=-90, north=90,
  is_south_to_north=False,
  nx=21600, ny=10800,
  xi=18601, xf=19200, yi=3001, yf=3600
(2) west=130, east=140, south=30, north=40,
  is_south_to_north=False,
  nx=600, ny=600,
  xi=1, xf=600, yi=1, yf=600 (xi, xf, yi and yf have default values)
Both have the region (west, east, south, north) = (130, 140, 30, 40) divided into (nx, ny) = (600, 600) pixels.

in_raster_sz, in_raster_lb, in_raster_ub

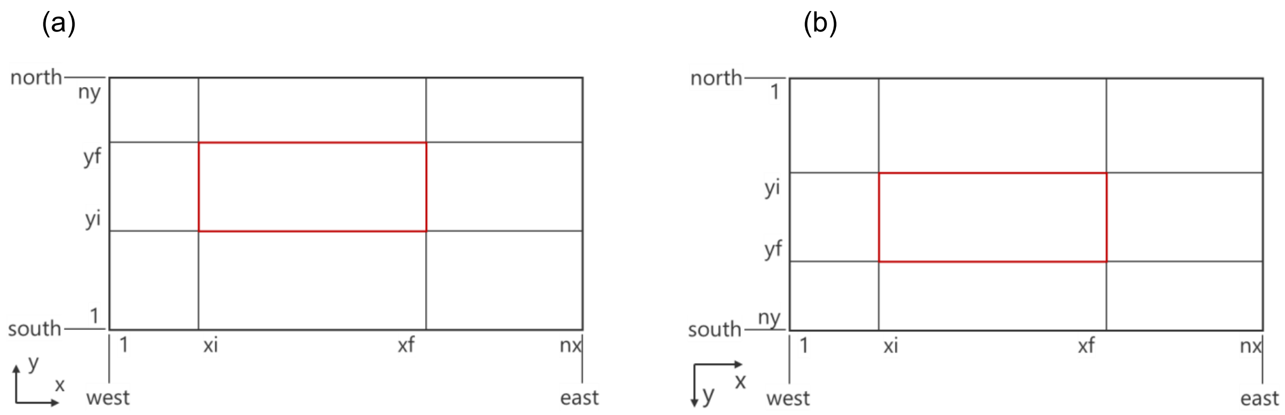

Figure 5-4 Clipping the region using *xi*, *xf*, *yi* and *yf*. Red box indicates the clipped region. (a) the case when the y-axis is from south to north and (b) the opposite case.

These variables are used when the size of input data and the number of pixels of the mesh do not match. *in_raster_sz* gives the size of input raster data, *in_raster_lb* and *in_raster_ub* give lower and upper bounds of the raster data to use, respectively.

in_grid_sz, in_grid_lb, in_grid_ub

Same to in_raster_sz, in_raster_lb, in_raster_ub, respectively, but for grid data.

Block *remapping*

Table 5-4 Setting variables for block *remapping*

Variable	Data type	Description	Default value
dir	path	Parent directory of the files below	‘’
fin_grdval	file	Input file of source grid values	‘’, dble, 1, little
fout_grdval	file	Output file of target grid values	‘’, dble, 1, little
length_rt	integer	Length of remapping table	
fin_rt_sidx	file	Output file of source grid indices of remapping table	‘’, int4, 1, little
fin_rt_tidx	file	Output file of target grid indices of remapping table	‘’, int4, 2, little
fin_rt_area	file	Output file of intersection areas of remapping table	‘’, dble, 1, little
fin_rt_coef	file	Output file of remapping coefficients of remapping table	‘’, dble, 1, little
fout_rt_sidx	file	Output file of source grid indices of remapping table	‘’, int4, 1, little
fout_rt_tidx	file	Output file of target grid indices of remapping table	‘’, int4, 2, little
fout_rt_area	file	Output file of intersection areas of remapping table	‘’, dble, 1, little

fout_rt_coef	file	Output file of remapping coefficients of remapping table	'', dble, 1, little
mesh_coef	string	Mesh of denominator in calculation of coefficients	target
mesh_sort	string	Mesh by whose grid indices the remapping table is sorted	target
allow_empty	boolean	Allow an empty remapping table	False
opt_coef_sum_modify	float	Summations of coefficients are modified to this value	
opt_coef_sum_modify_ulim	float	Summations of coefficients are limited to this value	
opt_coef_zero_positive	float	Positive coefficients lower than this value are removed	
opt_coef_zero_negative	float	Negative coefficients greater than this value are removed	
opt_coef_error_excess	float	Coefficients above 1.0+(this value) are not allowed	
opt_coef_sum_error_excess	float	Summations of coefficients above 1.0+(this value) are not allowed	
mesh_vrf	string	Mesh to make verification data	
fout_vrf_grdidx	file	Output file of grid indices correspondent to verification data	'', int4, 1, little
fout_vrf_grdara_true	file	Output file of true grid areas	'', dble, 1, little
fout_vrf_grdara_rt	file	Output file of grid areas calculated from the remapping table	'', dble, 1, little
fout_vrf_rerr_grdara	file	Output file of relative errors of grid areas	'', dble, 1, little
fout_vrf_grdnum	file	Output file of the number of intersecting grids	'', int4, 1, little
fout_vrf_iarea_sum	file	Output file of summation of intersection areas	'', dble, 1, little
fout_vrf_ifrac_sum	file	Output file of summation of coefficients	'', dble, 1, little
vrf_val_miss	float	Missing value of verification data	-1e20

Details

[Remapping tables]

length_rt, fin_sidx, fin_tidx, fin_area, fin_coef

Input files of remapping table. If you use the remapping table you have, specify them with these variables.

fout_sidx, fout_tidx, fout_area, fout_coef

Output file of remapping table. If you make a new remapping table, specify them with these variables.

[Grid values]

fin_grdval, fout_grdval

fin_grdval is grid point values of source mesh and *fout_grdval* is those of target mesh.

They can be specified more than once and the number of times they are specified must be the same. The remapped values of the *i*th *fin_grdval* are output to the *i*th *fout_grdval*.

[Options for calculation of coefficients]

mesh_coef

If mesh_coef is “target”, remapping coefficients are calculated by equation (3-2). If mesh_coef is “source”, they are calculated by equation (3-3).

allow_empty

If this is False (default) and the remapping table is empty (e.g. two meshes have no intersection), the program stops outputting an error. If this is True and the remapping table is empty, empty files are generated for *fout_rt_* (*=sidx, tidx, area or coef)*.

[Options for modification or thresholds of coefficients]

opt_coef_sum_modify

Summation of weights for each grid of *mesh_coef* is modified to this value.

opt_coef_sum_modify_ulim

Summation of weights for each grid of *mesh_coef* is modified to this value if it exceeds this value.

opt_coef_error_excess

Raise error if any of weights exceeded 1.0 + (this value).

opt_coef_sum_error_excess

Raise error if any of summations of weights for grids of *mesh_coef* exceeded 1.0 + (this value).

[Verification data]

mesh_vrf

Verification data below this variable are made for this mesh.

Following variables can be specified as verification data:

fout_vrf_grdidx

fout_vrf_grdara_true

fout_vrf_grdara_rt

fout_vrf_rerr_grdara

fout_vrf_grdnum

fout_vrf_iarea_sum (only for raster mesh)

fout_vrf_iratio_sum (only for raster mesh)

Example:

```
[remapping]
…

mesh_vrf: source
fout_vrf_grdidx     : "vrf/src_idx.bin", int4, 1, little
fout_vrf_grdara_true: "vrf/src_val.bin", dble, 1, little
fout_vrf_grdara_rt  : "vrf/src_val.bin", dble, 2, little
fout_vrf_rerr_grdara: "vrf/src_val.bin", dble, 3, little
fout_vrf_grdnum     : "vrf/src_num.bin", int4, 1, little
fout_vrf_iarea_sum  : "vrf/src_raster_area.bin" , dble, 1, little
fout_vrf_iratio_sum : "vrf/src_raster_ratio.bin", dble, 1, little
```

verification data for the source mesh

```
mesh_vrf: target
fout_vrf_grdidx     : "vrf/tgt_idx.bin", int4, 1, little
fout_vrf_grdara_true: "vrf/tgt_val.bin", dble, 1, little
fout_vrf_grdara_rt  : "vrf/tgt_val.bin", dble, 2, little
fout_vrf_rerr_grdara: "vrf/tgt_val.bin", dble, 3, little
fout_vrf_grdnum     : "vrf/tgt_num.bin", int4, 1, little
```

verification data for the target mesh

```
…
[end]
```

fout_vrf_grdidx

Output file of grid indices correspondent to verification data in *fout_vrf_grdara_true*, *fout_vrf_grdara_rt*, *fout_vrf_rerr_grdara* and *fout_vrf_grdnum*.

fout_vrf_grdara_true:

Output file of verification data, true grid areas A_i^{true} $(i = 1, \ldots, N)$, where *N* is the number of grids.

fout_vrf_grdara_rt

Output file of verification data, grid areas calculated from remapping table A_i^{rt} $(i = 1, \ldots, N)$.

fout_vrf_rerr_grdara

Output file of verification data, relative error A^{err} of A^{rt} defined by

$$A_i^{err} = \frac{A_i^{rt} - A_i^{true}}{A_i^{true}} \ (i = 1, \ldots, N).$$

fout_vrf_grdnum

Output file of verification data, the number of grids of the other mesh intersecting with each grid of the mesh of *mesh_vrf*.

fout_vrf_iarea_sum, fout_vrf_iratio_sum
Output file of verification data, summation of intersection area and its ratio to the pixel area for each pixel. Data shape is correspondent to the raster mesh defined in the block of *mesh_raster*.

Block *options*

Table 5-5 Setting variables for block *options*

Variable	Data type	Description	Default value
old_files	string	Options for dealing with old files. {‘stop’, ‘remove’, ‘overwrite’}	stop
dir_intermediates	path	Output directory for intermediate data	Directory of report file
remove_intermediates	string	Remove intermediate data at the end of the execution	True
earth_shape	string	Shape of the Earth. {‘sphere’, ‘ellips’}	sphere
earth_r	float	Radius of the Earth in meter	371000.7900
earth_e2	float	Square of eccentricity of the Earth	

Details

[File handling]

old_files
This variable controls how the program behaves when the file specified as an output file already exists.
‘stop’: The program raises an error to stop.
‘remove’: Remove existing files at the beginning of the program.
‘overwrite’: Overwrite the file without removing it. Not recommended.

dir_intermediates
When intermediate data were generated, they are output into this directory.

remove_intermediates
Remove intermediate data before stopping the program. If the program stopped by any error or was stopped manually, it would not work.

[Shape of the Earth]

earth_shape, earth_r, earth_e2

The Earth's shape is approximated by a perfect sphere or an ellipsoid.

In case of a perfect sphere, *earth_shape* = *'sphere'* and *earth_r* is the radius. *earth_e2* is inactive. In default, *earth_r* takes the volumetric radius, with which the sphere has the volume same as the WGS84 Earth ellipsoid.

In case of an ellipsoid, *earth_shape* = *'ellips'*, *earth_r* is the major axis and *earth_e2* is the square of eccentricity. In default, the values of *earth_r* and *earth_e2* is those of WGS84 Earth ellipsoid; *earth_r* = 6378137.d0 and *earth_e2* = 0.00669437999014d0.

5.2 Executing the program

Remapping is done by executing a program file remap.exe in bin/std with a configuration file as a command line argument as follows (and as shown in the quick start):

```
$ cd $(SPRINGDIR)
$ ./bin/std/remap.exe set/remap/T85_to_COCO-1deg.conf
$ ls out/remap/T85_to_COCO-1deg
area.bin  conf.bin  grid.bin  remap/  report.txt  vrf/
```

grid.bin ... Grid number of the remapping table

area.bin ... Intersection area of the remapping table

coef.bin ... Interpolation coefficient of the remapping table

vrf/ ... Check data

remap/ ... Remapped data

5.3 Drawing figures

Table 5-6 Setting variables for block *figures*

Variable	Data type	Description	Default value
mesh	string	For which mesh the figure is	
path_fig	path	Path of the figure	''
f_grdval	file	Input file of grid values	'', dble, 1, little
figsize	tuple	Size of the figure	10, 6
dpi	integer	Resolution of the figure	300
linewidth	float	Width of grid lines	0.0
edgecolor	string	Color of grid lines	gray
cmap	string	Colormap of grid values	jet
color_miss	string	Color for missing value	none
vmin	float	Lower limit value in the color map	
vmax	float	Upper limit value in the color map	
title	string	Title of the figure	
dir	path	Parent directory of the paths below	''
show	boolean	Whether or not to display the figure	true

[Format]

Each figure starts with *mesh*. File of input data of grid values, *f_grdval*, is specified for each figure. Other settings for drawing figures such as *path_fig*, *figsize*, *colormap*, *edgecolor* can also be specified. Variables other than *path_fig* are succeeded to the following figures.

Here shows an example of remapping from GSWP3 mesh (latlon mesh) to NICAM mesh (polygon mesh). See also set/remap/GSWP3_nc_to_NICAM-gl05rl00-nohalo.conf for the settings of meshes.

```
[figures]
  cmap: YlGn
  vmin:  50000
  vmax: 100000

  mesh: source
  path_fig: "out/remap/GSWP3_nc_to_NICAM-gl05rl00-nohalo/field/PSurf_src.png"
  f_grdval: "dat/field/GSWP3_nc/GSWP3.BC.PSurf.3hrMap.ILS.2000010100.bin", real
  linewidth: 0
  title: "source"

  mesh: target
  path_fig: "out/remap/GSWP3_nc_to_NICAM-gl05rl00-nohalo/field/PSurf_tgt.png"
  f_grdval: "out/remap/GSWP3_nc_to_NICAM-gl05rl00-nohalo/field/PSurf.bin", real
  linewidth: 0.5
  edgecolor: white
  title: "target"
[end]
```

The settings on the head, *cmap*, *vmin*, *vmax* are used until they are updated. The settings after *mesh* are applied to the current figure and succeeded by the following figures. Figures made by these settings are shown in Figure 5-5. Note that drawing figures for high resolution-polygon meshes may take a long time since the script fills colors for each grid for polygon meshes.

[Details]

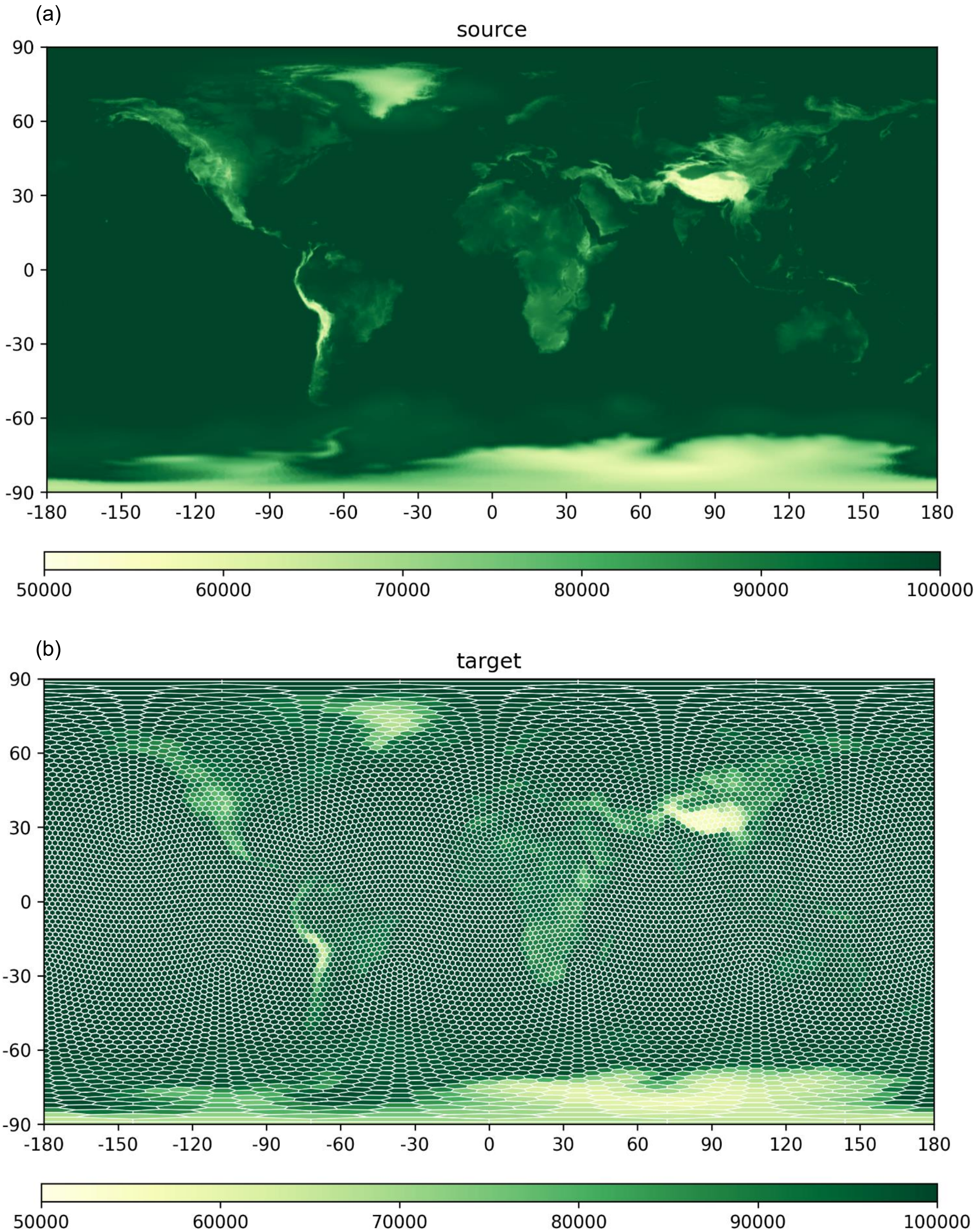

Figure 5-5 Grid values of (a) the source mesh and (b) the target mesh.

6 Merging remapping tables

6.1 Setting variables

Configuration files consist of:

- Path of report file
- Block *input*
- Block *output*
- Block *options*

Block *input*

Table 6-1 Setting variables for block *input*

Variable	Data type	Description	Default value
dir	path	Parent directory of the files below	‘’
length_rt	integer	Length of remapping table	
f_rt_sidx	file	Input file of source indices of remapping table	‘’, int4, 1, little
f_rt_tidx	file	Input file of target indices of remapping table	‘’, int4, 1, little
f_rt_area	file	Input file of intersection areas of remapping table	‘’, dble, 1, little
f_rt_coef	file	Input file of remapping coefficients of remapping table	‘’, dble, 1, little
length_grid	integer	Length of grid data	
f_grdidx	file	Input file of grid indices	‘’, int4, 1, little
f_grdara	file	Input file of grid area	‘’, dble, 1, little
idx_miss	integer	Missing value of grid index	-9999
opt_idx_duplication	string	Options for duplication of grid indices between/among remapping tables. {‘stop’, ‘sum’}	stop

[Details]

length_rt, f_rt_* (*=sidx, tidx, area, coef)

Information on each remapping table consists of *length_rt* followed by correspondent files, *f_rt_sidx*, *f_rt_tidx*, *f_rt_area* and *f_rt_coef*.

Example:

```
dir: “../out/remapping/io-met_to_mat-river”
length_rt: 77831
f_rt_sidx: “grid.bin”, rec=1
f_rt_tidx: “grid.bin”, rec=2
```

```
f_rt_area: “area.bin”
f_rt_coef: “coef.bin”

dir: “../out/remapping/io-met_to_mat-noriv”
length_rt: 24310
f_rt_sidx: “grid.bin”, rec=1
f_rt_tidx: “grid.bin”, rec=2
f_rt_area: “area.bin”
f_rt_coef: “coef.bin”
```

Block *output*

Table 6-2 Setting variables for block *output*

Variable	Data type	Description	Default value
dir	path	Parent directory of the files below	
mesh_coef	string	Mesh of denominator in calculation of coefficients	target
mesh_sort	string	Mesh by whose grid indices the remapping table is sorted	target
f_rt_sidx	file	Output file of source grid indices of remapping table	‘’, int4, 1, little
f_rt_tidx	file	Output file of target grid indices of remapping table	‘’, int4, 1, little
f_rt_area	file	Output file of intersection areas of remapping table	‘’, dble, 1, little
f_rt_coef	file	Output file of remapping coefficients of remapping table	‘’, dble, 1, little
opt_coef_sum_modify	float	Summations of coefficients are modified to this value	
opt_coef_sum_modify_ulim	float	Summations of coefficients are limited to this value	
opt_coef_zero_positive	float	Positive coefficients lower than this value are removed	
opt_coef_zero_negative	float	Negative coefficients greater than this value are removed	
opt_coef_error_excess	float	Coefficients above 1.0+(this value) are not allowed	
opt_coef_sum_error_excess	float	Summations of coefficients above 1.0+(this value) are not allowed	
f_grdidx	file	Output file of merged grid index data	‘’, int4, 1, little
f_grdara	file	Output file of merged grid area data	‘’, dble, 1, little

Details

[Calculation of coefficients]

mesh_coef

See *Block remapping* in 5.

[Options for modification or thresholds of coefficients]

opt_coef_sum_modify, opt_coef_sum_modify_ulim,
opt_coef_error_excess, opt_coef_sum_error_excess
See *Block remapping* in 5.

[Grid data]

f_grdidx, f_grdara
These are used to calculate remapping coefficients. It opt_coef_sum_modify was not specified, remapping coefficients are calculated from intersection area in the remapping table specified by *f_rt_area* and grid area specified by *f_grdara*.
Grid data of f_grdidx and grid area of f_grdara must be correspondent to each other. Grid area data can be made using the program bin/std/make_grid_data.exe (see Merging remapping tables for details).

Block *options*

Table 6-3 Setting variables for block *options*

Variable	Data type	Description	Default value
old_files	string	Options for dealing with old files. {‘stop’, ‘remove’, ‘overwrite’}	stop
dir_intermediates	path	Output directory for intermediate data	Directory of report file
remove_intermediates	string	Remove intermediate data at the end of the execution	True

Details

See *Block options* in 5.1.

6.2 Executing the program

Merging remapping tables is done by executing a program file merge_remapping_tables.exe in bin/std with a configuration file as a command line argument as follows:

```
$ cd $(SPRINGDIR)
### Make remapping tables
$ ./bin/std/remap.exe set/remap/NICAM-gl05rl00-nohalo_to_MATSIRO-glb30min-river.conf
$ ./bin/std/remap.exe set/remap/NICAM-gl05rl00-nohalo_to_MATSIRO-glb30min-noriv.conf
### Make grid data
```

```
$ ./bin/std/make_grid_data.exe set/make_grid_data/MATSIRO_glb30min_river.conf
$ ./bin/std/make_grid_data.exe set/make_grid_data/MATSIRO_glb30min_noriv.conf
### Merge remapping tables
$ ./bin/std/merge_remapping_tables.exe set/merge_remapping_tables/NICAM-gl05rl00-
ohalo_to_MATSIRO-glb30min.conf
$ ls out/remap/NICAM-gl05rl00-nohalo_to_MATSIRO-glb30min/merged
area.bin  conf.bin  grid.bin  report.txt
```

7 Making grid data

7.1 Setting variables

Configuration files consist of:

- Path of report file
- One mesh block in *mesh_latlon*, *mesh_polygon* and *mesh_raster*
- Block *options* (optional)

Block *mesh_latlon*

Table 7-1 Setting variables for block *mesh_latlon*

Variable	Data type	Description	Default value
name	string	Mesh name	mesh
nx	integer	The number of grids in longitudinal direction	
ny	integer	The number of grids in latitudinal direction	
west	float	West edge of the mesh	
east	float	East edge of the mesh	
south	float	South edge of the mesh	
north	float	North edge of the mesh	
is_south_to_north	boolean	Grid data is arranged from south to north	True
dir	path	Parent directory of the files below	‘’
f_lon_bound	file	Input file of longitudes of latitude-direction grid lines	‘’, dble, 1, little
f_lat_bound	file	Input file of latitudes of longitude-direction grid lines	‘’, dble, 1, little
coord_unit	string	Unit of coordinates in *f_lon_bound* and *f_lat_bound*. {‘degree’, ‘radian’}	degree
idx_bgn	integer	Initial number of grid index automatically set	1
fin_grdidx	file	Input file of grid indices	‘’, int4, 1, little
fin_grdara	file	Input file of grid areas	‘’, dble, 1, little
fin_grdwgt	file	Input file of grid weights	‘’, dble, 1, little
in_grid_sz	integer tuple	Size of input grid data	*nx, ny*
in_grid_lb	integer tuple	Lower bounds of grid data to use	1, 1
in_grid_ub	integer tuple	Upper bounds of grid data to use	*nx, ny*
in_unit_ara	string	Unit of input grid areas. {‘m2’, ‘km2’}	m2
out_form	string	Method of formatting output grid data. {‘auto’, ‘index’}	auto
fout_grdmsk	file	Output file of grid mask	‘’, int4, 1, little
fout_grdidx	file	Output file of grid indices	‘’, int4, 1, little
fout_grdara	file	Output file of grid areas	‘’, dble, 1, little
fout_grdwgt	file	Output file of grid weights	‘’, dble, 1, little
fout_grdx	file	Output file of x coordinates of grid centers	‘’, dble, 1, little

fout_grdy	file	Output file of y coordinates of grid centers	‘’, dble, 1, little
fout_grdz	file	Output file of z coordinates of grid centers	‘’, dble, 1, little
fout_grdlon	file	Output file of longitudes of grid centers	‘’, dble, 1, little
fout_grdlat	file	Output file of latitudes of grid centers	‘’, dble, 1, little
out_grid_sz	tuple integer	Size of the output grid data	*nx, ny*
out_grid_lb	tuple integer	Lower bounds of the range to write	1, 1
out_grid_ub	tuple integer	Upper bounds of the range to write	*nx, ny*
out_unit_ara	tuple integer	Unit of output grid area. {‘m2’, ‘km2’}	m2
out_unit_xyz	tuple integer	Unit of output grid center’s cartesian coordinates. {‘m’, ‘km’}	m
out_unit_lonlat	tuple integer	Unit of output grid center’s spherical coordinates. {‘degree’, ‘radian’}	degree
idx_miss	integer	Missing value of grid index	-9999
ara_miss	float	Missing value of grid area	-1e20
wgt_miss	float	Missing value of grid weight	-1e20
xyz_miss	float	Missing value of cartesian coordinates	-1e20
lonlat_miss	float	Missing value of spherical coordinates	-1e20

Details

See *Block mesh_latlon* in 5.1, since most of the variables are common to it.

[Output grid data]

out_form

‘auto’: grid data is arranged in the order of grid indices with no missing value.

‘index’: grid data is arranged in the order of grid indices given by *fin_grdidx*.

fout_grdmsk

Grid mask data. 1 for valid and 0 for invalid.

fout_grdx, fout_grdy, fout_grdz, fout_grdlon, fout_grdlat

The center of grid is defined as an intersection of the Earth’s surface and the line that passes through the origin and the point *Pc*. Pc is defined by the following equation:

$$\boldsymbol{P}_c = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{P}_i,$$

where $\boldsymbol{P} = (x, y, z)$ is cartesian coordinates of the point and n is the number of vertices of the grid. Longitude and latitude are calculated from cartesian coordinates.

Block *mesh_polygon*

Table 7-2 Setting variables for block *mesh_polygon*

Variable	Data type	Description	Default value
name	string	Mesh name	mesh
np	integer	The maximum number of grid vertices	
nij	integer	The number of grids	
dir	path	Parent directory of the files below	‘’
f_lon_vertex	file	Input file of longitudes of vertices	‘’, dble, 1, little
f_lat_vertex	file	Input file of latitudes of vertices	‘’, dble, 1, little
f_x_vertex	file	Input file of x coordinates of vertices	‘’, dble, 1, little
f_y_vertex	file	Input file of y coordinates of vertices	‘’, dble, 1, little
f_z_vertex	file	Input file of z coordinates of vertices	‘’, dble, 1, little
coord_unit	string	Unit of coordinates. {‘degree’, ‘radian’} for spherical and {‘m’, ‘km’} for cartesian	‘degree’ for spherical and ‘m’ for cartesian
coord_miss	float	Missing value of coordinates	-1e20
f_arctyp	file	Input file of types of grid sides	‘’, int4, 1, little
arc_parallel	boolean	Grid sides having vertices with the same latitude are considered as horizontal	False
fin_grdidx	file	Input file of grid indices	‘’, int4, 1, little
fin_grdara	file	Input file of grid areas	‘’, dble, 1, little
fin_grdwgt	file	Input file of grid weights	‘’, dble, 1, little
in_grid_sz	integer tuple	Size of input grid data	*np, nij*
in_grid_lb	integer tuple	Lower bounds of grid data to use	1, 1
in_grid_ub	integer tuple	Upper bounds of grid data to use	*np, nij*
in_unit_ara	string	Unit of input grid areas. {‘m2’, ‘km2’}	m2
out_form	string	Method of formatting output grid data. {‘auto’, ‘index’}	auto
fout_grdmsk	file	Output file of grid mask	‘’, int4, 1, little
fout_grdidx	file	Output file of grid indices	‘’, int4, 1, little
fout_grdara	file	Output file of grid areas	‘’, dble, 1, little
fout_grdwgt	file	Output file of grid weights	‘’, dble, 1, little
fout_grdx	file	Output file of x coordinates of grid centers	‘’, dble, 1, little
fout_grdy	file	Output file of y coordinates of grid centers	‘’, dble, 1, little
fout_grdz	file	Output file of z coordinates of grid centers	‘’, dble, 1, little
fout_grdlon	file	Output file of longitudes of grid centers	‘’, dble, 1, little
fout_grdlat	file	Output file of latitudes of grid centers	‘’, dble, 1, little
out_grid_sz	tuple integer	Size of the output grid data	*nx, ny*
out_grid_lb	tuple integer	Lower bounds of the range to write	1, 1
out_grid_ub	tuple integer	Upper bounds of the range to write	*nx, ny*
out_unit_ara	tuple integer	Unit of output grid area. {‘m2’, ‘km2’}	m2
out_unit_xyz	tuple integer	Unit of output grid center’s cartesian coordinates. {‘m’, ‘km’}	m
out_unit_lonlat	tuple integer	Unit of output grid center’s spherical coordinates. {‘degree’, ‘radian’}	degree
idx_miss	integer	Missing value of grid index	-9999
ara_miss	float	Missing value of grid area	-1e20
wgt_miss	float	Missing value of grid weight	-1e20
xyz_miss	float	Missing value of cartesian coordinates	-1e20
lonlat_miss	float	Missing value of spherical coordinates	-1e20

Details

See *Block mesh_polygon* in 5.1, since most of the variables are common to it.

Block *mesh_raster*

Table 7-3 Setting variables for block *mesh_raster*

Variable	Data type	Description	Default value
name	string	Mesh name	mesh
nx	integer	The number of grids in longitudinal direction	
ny	integer	The number of grids in latitudinal direction	
west	float	West edge of the mesh	
east	float	East edge of the mesh	
south	float	South edge of the mesh	
north	float	North edge of the mesh	
is_south_to_north	boolean	Grid data is arranged from south to north	True
xi	integer	Lower bound of the valid range in the longitudinal direction in the mesh	1
xf	integer	Upper bound of the valid range in the longitudinal direction in the mesh	*nx*
yi	integer	Lower bound of the valid range in the latitudinal direction in the mesh	1
yf	integer	Upper bound of the valid range in the latitudinal direction in the mesh	*ny*
dir	path	Parent directory of the files below	‘’
fin_rstidx	file	Input file of raster indices	‘’, int4, 1, little
fin_rstara	file	Input file of raster areas	‘’, dble, 1, little
fin_rstwgt	file	Input file of raster weights	‘’, dble, 1, little
in_raster_sz	integer tuple	Size of input raster data	*nx, ny*
in_raster_lb	integer tuple	Lower bounds of raster data to use	1, 1
in_raster_ub	integer tuple	Upper bounds of raster data to use	*nx, ny*
fin_grdidx	file	Input file of grid indices	‘’, int4, 1, little
fin_grdara	file	Input file of grid areas	‘’, dble, 1, little
fin_grdwgt	file	Input file of grid weights	‘’, dble, 1, little
in_grid_sz	integer tuple	Size of input grid data	
in_grid_lb	integer tuple	Lower bounds of grid data to use	1, 1
in_grid_ub	integer tuple	Upper bounds of grid data to use	*in_grid_sz*
in_unit_ara	string	Unit of input grid areas. {‘m2’, ‘km2’}	m2
idx_condition	string	Condition that raster indices and grid indices must be fulfill. {‘match’, ‘grid_in_raster’, ‘raster_in_grid’, ‘none’}	match
out_form	string	Method of formatting output grid data. {‘auto’, ‘index’}	auto
fout_grdmsk	file	Output file of grid mask	‘’, int4, 1, little
fout_grdidx	file	Output file of grid indices	‘’, int4, 1, little
fout_grdara	file	Output file of grid areas	‘’, dble, 1, little
fout_grdwgt	file	Output file of grid weights	‘’, dble, 1, little
fout_grdx	file	Output file of x coordinates of grid centers	‘’, dble, 1, little

fout_grdy	file	Output file of y coordinates of grid centers	‘’, dble, 1, little
fout_grdz	file	Output file of z coordinates of grid centers	‘’, dble, 1, little
fout_grdlon	file	Output file of longitudes of grid centers	‘’, dble, 1, little
fout_grdlat	file	Output file of latitudes of grid centers	‘’, dble, 1, little
out_grid_sz	tuple integer	Size of the output grid data	*nx, ny*
out_grid_lb	tuple integer	Lower bounds of the range to write	1, 1
out_grid_ub	tuple integer	Upper bounds of the range to write	*nx, ny*
out_unit_ara	tuple integer	Unit of output grid area. {‘m2’, ‘km2’}	m2
out_unit_xyz	tuple integer	Unit of output grid center’s cartesian coordinates. {‘m’, ‘km’}	m
out_unit_lonlat	tuple integer	Unit of output grid center’s spherical coordinates. {‘degree’, ‘radian’}	degree
idx_miss	integer	Missing value of grid index	-9999
ara_miss	float	Missing value of grid area	-1e20
wgt_miss	float	Missing value of grid weight	-1e20
xyz_miss	float	Missing value of cartesian coordinates	-1e20
lonlat_miss	float	Missing value of spherical coordinates	-1e20

Details

See *Block mesh_raster* in 5.1, since most of the variables are common to it.

7.2 Executing the program

Making grid data is done by executing a program file make_grid_data.exe in bin/std with a configuration file as a command line argument as follows:

```
$(SPRINGDIR)
$ ./bin/std/make_grid_data.exe set/make_grid_data/T85.conf
$ ls out/grid/T85
grdara.bin  grdidx.bin  grdlonlat.bin  grdxyz.bin  report_make_grid_data.txt
```

8 Rasterizing grid data

8.1 Setting variables

Configuration files consist of:

- Path of report file
- One mesh block in *mesh_latlon*, *mesh_polygon* and *mesh_raster*
- Block *output*
- Block *options* (optional)

Block *mesh_latlon*

Table 8-1 Setting variables for block *mesh_latlon*

Variable	Data type	Description	Default value
name	string	Mesh name	mesh
nx	integer	The number of grids in longitudinal direction	
ny	integer	The number of grids in latitudinal direction	
west	float	West edge of the mesh	
east	float	East edge of the mesh	
south	float	South edge of the mesh	
north	float	North edge of the mesh	
is_south_to_north	boolean	Grid data is arranged from south to north	True
dir	path	Parent directory of the files below	‘’
f_lon_bound	file	Input file of longitudes of latitude-direction grid lines	‘’, dble, 1, little
f_lat_bound	file	Input file of latitudes of longitude-direction grid lines	‘’, dble, 1, little
coord_unit	string	Unit of coordinates in *f_lon_bound* and *f_lat_bound*. {‘degree’, ‘radian’}	degree
idx_bgn	integer	Initial number of grid index automatically set	1
fin_grdidx	file	Input file of grid indices	‘’, int4, 1, little
fin_grdara	file	Input file of grid areas	‘’, dble, 1, little
fin_grdwgt	file	Input file of grid weights	‘’, dble, 1, little
fin_grdval	file	Input file of grid point values	‘’, dble, 1, little
in_grid_sz	integer tuple	Size of input grid data	*nx, ny*
in_grid_lb	integer tuple	Lower bounds of grid data to use	1, 1
in_grid_ub	integer tuple	Upper bounds of grid data to use	*nx, ny*
in_unit_ara	string	Unit of input grid areas. {‘m2’, ‘km2’}	m2
idx_miss	integer	Missing value of grid index	-9999
ara_miss	float	Missing value of grid area	-1e20
wgt_miss	float	Missing value of grid weight	-1e20
xyz_miss	float	Missing value of cartesian coordinates	-1e20
lonlat_miss	float	Missing value of spherical coordinates	-1e20
val_miss	float	Missing value of grid point value	-1e20

[Details]

See *Block mesh_latlon* in 5.1, since most of the variables are common to it.

Block *mesh_polygon*

Table 8-2 Setting variables for block *mesh_polygon*

Variable	Data type	Description	Default value
name	string	Mesh name	mesh
np	integer	The maximum number of grid vertices	
nij	integer	The number of grids	
dir	path	Parent directory of the files below	‘’
f_lon_vertex	file	Input file of longitudes of vertices	‘’, dble, 1, little
f_lat_vertex	file	Input file of latitudes of vertices	‘’, dble, 1, little
f_x_vertex	file	Input file of x coordinates of vertices	‘’, dble, 1, little
f_y_vertex	file	Input file of y coordinates of vertices	‘’, dble, 1, little
f_z_vertex	file	Input file of z coordinates of vertices	‘’, dble, 1, little
coord_unit	string	Unit of coordinates. {‘degree’, ‘radian’} for spherical and {‘m’, ‘km’} for cartesian	‘degree’ or ‘m’
coord_miss	float	Missing value of coordinates	-1e20
f_arctyp	file	Input file of types of grid sides	‘’, int4, 1, little
arc_parallel	boolean	Grid sides having vertices with the same latitude are considered as horizontal	False
fin_grdidx	file	Input file of grid indices	‘’, int4, 1, little
fin_grdara	file	Input file of grid areas	‘’, dble, 1, little
fin_grdwgt	file	Input file of grid weights	‘’, dble, 1, little
fin_grdval	file	Input file of grid point values	‘’, dble, 1, little
in_grid_sz	integer tuple	Size of input grid data	*np, nij*
in_grid_lb	integer tuple	Lower bounds of grid data to use	1, 1
in_grid_ub	integer tuple	Upper bounds of grid data to use	*np, nij*
in_unit_ara	string	Unit of input grid areas. {‘m2’, ‘km2’}	m2
idx_miss	integer	Missing value of grid index	-9999
ara_miss	float	Missing value of grid area	-1e20
wgt_miss	float	Missing value of grid weight	-1e20
val_miss	float	Missing value of grid point values	-1e20

[Details]

See *Block mesh_polygon* in 5.1, since most of the variables are common to it.

Block *raster*

Table 8-3 Setting variables for block *raster*

Variable	Data type	Description	Default value

nx	integer	The number of pixels in the longitudinal direction	
ny	integer	The number of pixels in the latitudinal direction	
west	float	West edge of the mesh	
east	float	East edge of the mesh	
south	float	South edge of the mesh	
north	float	North edge of the mesh	
is_south_to_north	boolean	Raster data is arranged from south to north	True

[Details]

See *Block mesh_raster* in 5.1, since most of the variables are common to it.

Block *output*

Table 8-4 Setting variables for block *output*

Variable	Data type	Description	Default value
ratio_min	float	Lower bound of range of intersection ratio of valid pixels	0.5
ratio_max	float	Upper bound of range of intersection ratio of valid pixels	1.0
include_min	boolean	ratio_min is included in the range	True
include_max	boolean	ratio_max is included in the range	True
ratio_zero_positive	float		0.0
ratio_sum_zero_positive	float		0.0
dir	path		‘’
f_area_sum	file	Output file of summations of intersection area of pixels	‘’, dble, 1, little
f_ratio_sum	file	Output file of summations of intersection ratio of pixels	‘’, dble, 1, little
f_mask	file	Output file of raster mask	‘’, int4, 1, little
f_idx	file	Output file of raster indices	‘’, int4, 1, little
val_miss	float	Missing values	-1e20

Details

[Defining valid pixels]

ratio_min, ratio_max, include_min, include_max

Their values affect the mask data specified by *f_mask*.

Intersection ratio r_i of the ith pixel is calculated by

$$r_i = \frac{1}{A_i}\sum_{k=1}^{n} a_{i,k}\,,$$

where A_i is the area of the pixel, $a_{i,k}$ is an intersection area with the kth grid and n is the number of grids.

Only the pixels that have intersection ratio in a certain range are considered valid. For example, if the source mesh has missing grids, part of pixels will intersect with missing grids or not intersect with valid grids at all. Or, if the source mesh does not cover the whole area of the raster mesh, part of pixels will have intersection area smaller than their own area. Those pixels with small intersection area are considered missing pixels in general, and those with no intersection are definitely missing.

ratio_min, *ratio_max*, *include_min* and *include_max* control the range of valid pixels' intersection ratio. If *include_min* is True, the pixels with intersection ratio exactly same to *include_min* is considered valid, and vice versa. It is the same for *include_max*. In general, *ratio_max* is 1.0 and *include_max* is True.

Examples:
ratio_min=0.3, ratio_max=1.0, include_min=True, include_max=True → range is [0.3, 1.0]
ratio_min=0.3, ratio_max=1.0, include_min=False, include_max=True → range is (0.3, 1.0]

ratio_zero_positive
Intersection of a polygon and a pixel less than this value is ignored if its ratio to the pixel area is less than the value of *ratio_zero_positive*. It is usually used to exclude tiny intersections that sometimes appears due to calculation errors.

ratio_sum_zero_positive
Its value affects the index data specified by *f_idx*.
Missing index is applied to the pixels whose total intersection ratio is less than this value.

[Output files]

f_mask
1 for valid and 0 for invalid pixels.
Valid pixels are defined by *ratio_min*, *ratio_max*, *include_min* and *include_max*.

f_area_sum, f_ratio_sum
Summation of intersection area for each pixel and its ratio to the pixel area.

f_idx
Index of the source mesh that has the largest intersection with the pixel.

Block *options*

Table 8-5 Setting variables for block *options*

Variable	Data type	Description	Default value
old_files	string	Options for dealing with old files. {'stop', 'remove', 'overwrite'}	stop
dir_intermediates	path	Output directory for intermediate data	Directory of report file
remove_intermediates	string	Remove intermediate data at the end of the execution	True
earth_shape	string	Shape of the Earth. {'sphere', 'ellips'}	sphere
earth_r	float	Radius of the Earth in meter	371000.7900
earth_e2	float	Square of eccentricity of the Earth	

[Details]

See *Block options* in 5.1.

8.2 Executing the program

Rasterizing grid data is done by executing a program file rasterize.exe in bin/std, giving a configuration file as a command line argument as follows:

```
$ cd $(SPRINGDIR)
$ ./bin/std/rasterize.exe set/rasterize/COCO-1deg_1min_frc-0.5.conf
$ ls out/rasterize/COCO-1deg/1min/lndfrc-0.5/
idx.bin  lndara.bin  lndfrc.bin  lndmsk.bin  report.txt
```

9 [Extension] Making remapping tables for new boundary data of ILS version-MATSIRO

9.1 Executing the script

Here shows how to prepare a directory and setting file and run the script.

```
### Make a directory for your project and prepare the script
$ cd $(SPRINGDIR)/run/ILS_bnd
$ ls
sample_cpl_ILS-AOGCM/  sample_FLOW_v396/
$ ls *
sample_cpl_ILS-AOGCM:
conf.json  ‘run_all.py

sample_FLOW_v396:
conf.json  ‘run_all.py
$ cp -r sample_FLOW_v396 glb30min  ### “glb30min” is just an example
$ cd glb30min

### Edit the setting file
$ vim conf.json

### Run the script with Python 3.x
### It will take a long time. You can run the script in parallel to reduce execution
time. See descriptions below for more detail.
$ python run_all.py

### Check the outputs
$ ls
conf.json  log/  out/  ‘run_all.py  set/  tmp/

### Main products are in the directory `out`
$ ls out/*
out/mesh:
‘CMF/  ‘MATSIRO/

out/remapping_table:
‘GLCNMO/  ‘GTOPO30/  ‘HWSD/  ‘ISLSCP1/  ‘JRA55/  ‘MODIS/
$ ls out/mesh/CMF
river/  noriv/
$ ls out/mesh/CMF/river
grididx.bin  rstidx.bin
$ ls out/mesh/MATSIRO/river
grdbndidx.bin  grdidx.bin  rstidx.bin
$ ls out/remapping_table/GLCNMO
mapping_table_area_E000N10_noriv.bin
```

```
mapping_table_area_E000N10_river.bin
mapping_table_area_E000N20_noriv.bin
…
mapping_table_area_W180S80_river.bin
mapping_table_coef_E000N10_noriv.bin
…
mapping_table_coef_W180S80_river.bin
mapping_table_idx_E000N10_noriv.bin
…
mapping_table_idx_W180S80_river.bin
report_E000N10_noriv.txt
…
report_W180S80_river.txt
$ ls out/remapping_table/ISLSCP1

mapping_table_area_global_noriv.bin
mapping_table_area_global_river.bin
mapping_table_coef_global_noriv.bin
mapping_table_coef_global_river.bin
mapping_table_idx_global_noriv.bin
mapping_table_idx_global_river.bin
report_global_noriv.txt
report_global_river.txt
```

Executing the second step takes a substantially long time since it uses big data. To reduce the executional time, use options to specify the data, land type and tile name as explained below.

9.2 Steps

The process of creating the product consists of following two steps.

1 Making mesh data of CMF and MATSIRO

2 Making remapping tables

1 Making mesh data of CMF and MATSIRO

Mesh data of CMF and MATSIRO are made from catmxy data and nextxy data. catmxy data is a unit catchment index data represented by x-index and y-index. nextxy is a river network data. These are included in CMF dataset or generated by FLOW program.

2 Making remapping tables

Remapping tables are made from data made in step 1 and mesh information of each dataset.

ILS uses six datasets listed below and SPRING makes remapping tables for each.

Table 9-1 Input data for boundary data of MATSIRO

Name	Data	Mesh type	Tiling (x,y)
GLCNMO	Land cover	LatLon	36x18

GTOPO30	Topography	LatLon	36x18
HWSD	Soil type	LatLon	36x18
ISLSCP1	Radiation?	LatLon	Not tiled
JRA55	Air temperature	LatLon	36x18
MODIS	LAI	Polygon (sinusoidal projection)	36x18 (sinusoidal projection)

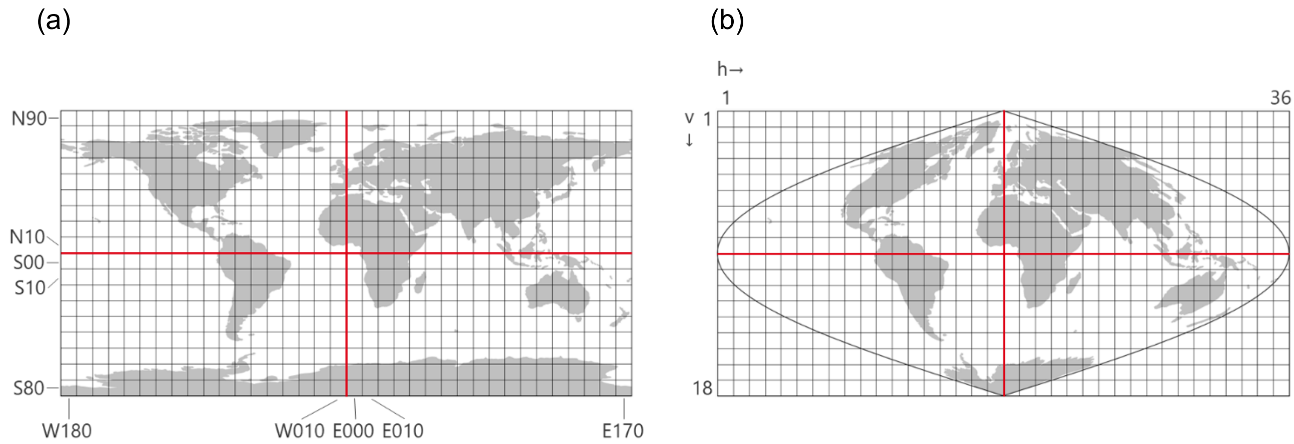

Figure 9-1 Tiling of input data. (a) LatLon mesh and (b) MODIS (sinusoidal) mesh. Red lines indicate 0 degree. Both maps are divided into 36 pieces in longitudinal direction and 18 pieces in latitudinal direction. In LatLon mesh, tiles are named using latitudes of northern grid lines and longitudes of western grid lines, e.g., E130N40. In MODIS mesh, tiles are named by their horizonal index and vertical index, e.g., h18v02.

9.3 Options of the script

The script can take options:

```
$ python run_all.py -h
usage: run_all.py [-h] [-s STEP] [-x] [-d DATA] [-l LAND]
                  [-t TILE] [-f CONFIG]

optional arguments:
  -h, --help            show this help message and exit
  -s STEP, --step STEP  step number (<= 2)
  -x                    not update data
  -d DATA, --data DATA  data name
  -l LAND, --land LAND  land type
  -t TILE, --tile TILE  tile name
  -f CONFIG, --config CONFIG
                        config file
```

- **Executing a single step**

The command, *python run_all.py*, executes all steps. To execute a single step, use the *-s* / *--step* option. These options can take a step number from 1 to 2 as an argument.

- **Specifying the configuration file**

  conf.json is used in default. To specify a configuration file, use the *-f* / *-file* option.

- **Specifying data, land type and tile name at step 2**

Some of these data are tiled and remapping table is made for each. This process is not parallelized in this script, so execute the script in parallel, utilizing the option as described in this section, to shorten the time to last this step.

In the default setting file *conf.json*, “GLCNMO”, “GTOPO30”, “ISLSCP1”, “JRA55” and “MODIS” are defined as input data and “river” and “noriv” as land type. To make a remapping table for data “GLCNMO”, land type “river” and tile “W180N90”, execute

```
$ python run_all.py -s 2 -d GLCNMO -l river -t W180N90
```

Remapping tables for all tiles are made if tile name was not specified as follows:

```
$ python run_all.py -s 2 -d GLCNMO -l river
```

This is similar to the options for data name and land type.

One easy way to parallelize the step is to divide it by data as follows:

```
for data in GLCNMO GTOPO30 ISLSCP1 JRA55 MODIS; do
  $ python run_all.py -s 2 -d $data >log/${data}.out 2>&1 &
done
```

or by data and land type as follows:

```
for data in GLCNMO GTOPO30 ISLSCP1 JRA55 MODIS; do
  for land in river noriv; do
    $ python run_all.py -s 2 -d $data -l $land >log/${data}_${land}.out 2>&1 &
  done
done
```

9.4 Setting variables

Modify values in the configuration file depending on the mesh or data you use.

- (Top)

dir_top

The absolute path of the top directory of SPRING.

- **meshes > CMF**

The mesh of CMF.

- **inputData**

Meshes of input data.

- **remappingTables_common**

Common settings for remapping tables.

10 [Extension] Making remapping tables for default ILS

In ILS, data are exchanged between components using remapping tables. ILS has three components in default: I/O, MATSIRO and CaMa-Flood. I/O is a component to input/output data. The models, MATSIRO and CMF, basically handle the input/output data through the I/O component. Different remapping tables are necessary for different forms of input/output data. MATSIRO has three kinds of input data: boundary data, forcing data given by NetCDF files and those given by plain binary data. Default ILS uses eight remapping tables:

IO_bnd (boundary data) to MATSIRO
IO_metnc (NetCDF forcing data) to MATSIRO
IO_met (plain binary forcing data) to MATSIRO
MATSIRO to IO_matrow (MATSIRO model grid, i.e., catchment-based grid form)
MATSIRO to IO_latlon (latlon grid form)
CMF to IO_cmfrow (CMF model grid)
MATSIRO to CMF
CMF to MATSIRO

10.1 Executing the script

Here shows how to prepare a directory and setting file and run the script.

```
### Make a directory for your project and prepare the script
$ cd $(SPRINGDIR)/run/rt_ILS_default
$ ls *
sample_CMF_v407_catmxy:
conf.json  ‘run_all.py

sample_CMF_v422_catmxy:
conf.json  ‘run_all.py

sample_FLOW_v396_catmxy:
conf.json  ‘run_all.py

sample_FLOW_v396_idxmap
conf.json  ‘run_all.py

$ cp -r sample_CMF_v422_catmxy my_project  ### “my_project” is just an example
$ cd my_project

### Edit the setting file
$ vim conf.json

### Run the script with Python 3.x
$ python run_all.py
```

```
### Check the outputs
$ ls
conf.json  log/  out/  ‘run_all.py  set/  tmp/

### Main products are in the directory `out`
$ ls out/*
out/mesh:
‘CMF/  MATSIRO/

out/remapping_table:
‘CMF_to_IO_cmfrow/  ‘IO_bnd_to_MAT/    ‘MAT_to_CMF/
‘CMF_to_IO_latlon/  ‘IO_metnc_to_MAT/  ‘MAT_to_IO_latlon/
‘CMF_to_MAT/        ‘IO_met_to_MAT/    ‘MAT_to_IO_matrow/
```

10.2 Steps

The process of creating the product consists of following four steps:

1. Making mesh data of CMF and MATSIRO
2. Making grid data of CMF and MATSIRO
3. Making remapping tables
4. Merging remapping tables

1 Making mesh data of CMF and MATSIRO

Mesh data of CMF and MATSIRO are made from catmxy data and nextxy data. catmxy data is a unit catchment index data represented by x-index and y-index. nextxy is a river network data. These are included in CMF dataset or generated by FLOW program.

2 Making grid data of CMF and MATSIRO

Grid areas are calculated, that are used in the subsequent step.

3 Making remapping tables

Part of remapping tables made in this step are used as they are:

- MATSIRO to CMF
- CMF to IO_cmfrow
- CMF to IO_latlon

Others are merged in the next step.

4 Merging remapping tables

MATSIRO has two land types, river and noriv. Remapping tables are made for each land type in the previous step and merged here. Following remapping tables are merged:

- IO_bnd to MATSIRO
- IO_metnc to MATSIRO
- IO_met to MATSIRO
- MATSIRO to IO_matrow
- MATSIRO to IO_latlon

10.3 Options of the script

The script can take options:

```
$ python run_all.py -h
usage: run_all.py [-h] [-s STEP] [-x] [-f CONFIG]

optional arguments:
  -h, --help            show this help message and exit
  -s STEP, --step STEP  step number (<= 4)
  -x                    not update data
  -f CONFIG, --config CONFIG
                        config file
```

- **Executing a single step**

The command, *python run_all.py*, executes all steps. To execute a single step, use the *-s* / *--step* option. These options can take a step number from 1 to 4 as an argument.

- **Specifying the configuration file**

conf.json is used in default. To specify a configuration file, use the *-f* / *-file* option.

10.4 Setting variables

Modify values in the configuration file depending on the mesh or data you use.

- (Top)

dir_top

The absolute path of the top directory of SPRING.

landTypes

A list of land types.

If you use lands with no river, specify *["river, noriv"]*. If not, specify *["river"]*.

- **meshes > CMF**

nx_raster, ny_raster

The number of pixels in longitudinal and latitudinal direction, respectively.

nx_grid, ny_grid
The number of grids in longitudinal and latitudinal direction, respectively.

west, east, south, north
Boundaries of the mesh.

dir
The parent directory of the files. If it is relative path, the base directory is *dir_top*.

fin_catmxy, catmxy_index
Input file of catmxy data and its special values, respectively.

fin_nextxy, nextxy_index
Input file of nextxy data and its special values, respectively.

fin_rstidx_river, fin_grdidx_river
Input files of raster index and grid index of the model, respectively.
Specify them when you use index data made from catmxy data.

- **meshes > MATSIRO_{landType}**

fin_rstidx, fin_grdidx, fin_grdbndidx
Input files of raster index, grid index of the model and grid index of boundary data, respectively.
Specify them when you use index data made from catmxy data.

- **meshes > IO_metnc**

The I/O mesh for forcing data input as NetCDF files.

- **meshes > IO_met**

The I/O mesh for forcing data input as plain binary files.

- **meshes > IO_latlon**

The I/O mesh for output data remapped to latlon mesh.

Other meshes in meshes such as *MATSIRO_bnd_simple_{landType}* and *IO_MATSIRO_row_{landType}* are automatically defined based on the CMF mesh.

For the case that catmxy data is used, specify:
meshes > *CMF*
> *fin_catmxy, catmxy_index, fin_nextxy, nextxy_index*
and do not specify:
meshes > *CMF*
> *fin_rstidx_river, fin_grdidx_river*
meshes > *MATSIRO_{landType}*
> *fin_rstidx*, *fin_grdidx, fin_grdbndidx*
See the configuration file in sample_CMF_v407_catmxy/, sample_CMF_v422_catmxy/ or sample_FLOW_v396_catmxy/ under $(SPRING)/run/rt_ILS_default/.

For the case that idxmap is used, specify:
meshes > *CMF*
> *fin_rstidx_river, fin_grdidx_river*
meshes > *MATSIRO_{landType}*
> *fin_rstidx*, *fin_grdidx, fin_grdbndidx*
and do not specify:
meshes > *CMF*
> *fin_catmxy, catmxy_index, fin_nextxy, nextxy_index*
See the configuration file in sample_FLOW_v396_idxmap/ under $(SPRING)/run/rt_ILS_default/.

10.5 Correspondence with ILS

Correspondence of the names of remapping tables between SPRING and ILS are shown in Table 10-1. *MAT_to_IO_latlon* or *CMF_to_IO_latlon* of SPRING are not used in ILS in default. Names of remapping tables in ILS point to the directories of the remapping tables specified in $(ILS)/runs/coupling.conf.

Table 10-1 Correspondence of the names of remapping tables between SPRING and ILS

SPRING	ILS
IO_bnd_to_MAT	bnd2mat
IO_metnc_to_MAT	metnc2mat
IO_met_to_MAT	met2mat
MAT_to_CMF	mat2cama
MAT_to_IO_matrow	mat2io
MAT_to_IO_latlon	
CMF_to_MAT	cama2mat
CMF_to_IO_cmfrow	cama2io
CMF_to_IO_latlon	

11 [Extension] Defining grid data and making remapping tables for ILS-AOGCM coupling

11.1 Executing the script

Here shows how to prepare a directory and setting file and run the script.

```
### Make a directory for your project and prepare the script
$ cd $(SPRINGDIR)/run/cpl_ILS-AOGCM
$ ls
sample/
$ ls sample
conf.json  ‘run_all.py
$ cp -r sample my_project  ### “my_project” is just an example
$ cd my_project

### Edit the setting file
$ vim conf.json

### Run the script with Python 3.x
$ python run_all.py

### Check the outputs
$ ls
conf.json  log/  out/  ‘run_all.py  set/  tmp/

### Main products are in the directory `out`
$ ls out/*
out/mesh:
‘AGCM/  LSM/  ‘OGCM_ocean/  RM/  RM_CMF/

out/remapping_table:
‘AGCM_to_LSM/        ‘LSM_noriv_to_OGCM_via_AGCM/  ‘LSM_to_RM/
‘IO_bnd_to_LSM/      ‘LSM_to_AGCM/                 ‘RM_to_IO_latlon/
‘IO_metnc_to_LSM/    ‘LSM_to_IO_latlon/            ‘RM_to_IO_row/
‘IO_met_to_LSM/      ‘LSM_to_IO_row/               ‘RM_to_LSM/
‘LSM_noriv_to_AGCM/  ‘LSM_to_OGCM_via_AGCM/        ‘RM_to_OGCM_via_AGCM/
```

11.2 Steps

The process of creating the product consists of following eleven steps:

1 Making grid data of AGCM and OGCM

2 Rasterizing OGCM

3 Running FLOW to construct a river network
4 Making index data of CMF
5 Making grid data of CMF
6 Making standard remapping tables (1)
7 Defining the MATSIRO mesh
8 Making grid data of MATSIRO
9 Making standard remapping tables (2)
10 Making special remapping tables
11 Merging remapping tables

11.3 Options of the script

The script can take options:

- **Executing a single step**

The command, *python run_all.py*, executes all steps. To execute a single step, use the *-s* / *--step* option. These options can take a step number from 1 to 11 as an argument.

- **Specifying the configuration file**

conf.json is used in default. To specify a configuration file, use the *-f* / *--file* option.

11.4 Setting variables

Modify values in the configuration file depending on the mesh or data you use.

- (Top)

dir_top

The absolute path of the top directory of SPRING.

- **meshes**

Meshes of OGCM, AGCM, LSM (MATSIRO), river model (CMF) and I/O components are specified here. Mesh type is specified by “type” for each mesh. Setting variables for each mesh type are listed in Table 11-1 to Table 11-3.

- **meshes > OGCM_{landType}**

The OGCM mesh for different land types, *“ocean”* and *“land”*.

In addition to the normal setting variables listed in Table 11-1 to Table 11-3, a list of land types, *[“ocean”, “land”]* are given as *landTypes*.

- **meshes > AGCM**

The AGCM mesh.

Mesh type can be specified by “type”, but currently only “latlon” is valid because the current version of the program to construct a river network can handle only latlon mesh for AGCM mesh.

- **meshes > RM**

The CMF (River Model; RM) mesh.

- **meshes > IO_metnc**

The input mesh for forcing data input as NetCDF files.

- **meshes > IO_met**

The input mesh for forcing data input as plain binary files.

- **meshes > IO_latlon**

The output latlon mesh for LSM and RM. Some of output data of LSM are remapped to this mesh.

- **meshes > IO_LSM_bnd_{landType}**

The input mesh for boundary data of LSM for each land type.

Automatically defined based on the CMF mesh.

- **meshes > IO_LSM_row_{landType}**

The output mesh to output the grid point values of MATSIRO as they are.

Automatically defined based on the CMF mesh.

- **meshes > IO_RM_row_{landType}**

The output mesh to output the grid point values of CMF as they are.

Automatically defined based on the CMF mesh.

Table 11-1 Setting variables for latlon meshes

Variable	Data type	Description	Default value
name	string	Mesh name	‘source’ or ‘target’
nx	integer	The number of grids in longitudinal direction	
ny	integer	The number of grids in latitudinal direction	
west	float	West edge of the mesh	
east	float	East edge of the mesh	
south	float	South edge of the mesh	
north	float	North edge of the mesh	
is_south_to_north	boolean	Grid data is arranged from south to north	True
dir	path	Parent directory of the files	‘’

f_lon_bound	file	Input file of longitudes of latitude-direction grid lines	‘’, dble, 1, little
f_lat_bound	file	Input file of latitudes of longitude-direction grid lines	‘’, dble, 1, little
coord_unit	string	Unit of coordinates in *f_lon_bound* and *f_lat_bound*. {‘degree’, ‘radian’}	degree
idx_bgn	integer	Initial number of grid index automatically set	1
fin_grdidx	file	Input file of grid indices	‘’, int4, 1, little
idx_miss	integer	Missing value of grid index	-9999

Table 11-2 Setting variables for polygon meshes

Variable	Data type	Description	Default value
name	string	Mesh name	‘source’ or ‘target’
np	integer	The maximum number of grid vertices	
nij	integer	The number of grids	
dir	path	Parent directory of the files	‘’
f_lon_vertex	file	Input file of longitudes of vertices	‘’, dble, 1, little
f_lat_vertex	file	Input file of latitudes of vertices	‘’, dble, 1, little
f_x_vertex	file	Input file of x coordinates of vertices	‘’, dble, 1, little
f_y_vertex	file	Input file of y coordinates of vertices	‘’, dble, 1, little
f_z_vertex	file	Input file of z coordinates of vertices	‘’, dble, 1, little
coord_unit	string	Unit of coordinates. {‘degree’, ‘radian’} for spherical and {‘m’, ‘km’} for cartesian	‘degree’ for spherical and ‘m’ for cartesian
coord_miss	float	Missing value of coordinates	-1e20
f_arctyp	file	Input file of types of grid sides	‘’, int4, 1, little
arc_parallel	boolean	Grid sides having vertices with the same latitude are considered as horizontal	False
idx_bgn	integer	Initial number of grid index automatically set	1
fin_grdidx	file	Input file of grid indices	‘’, int4, 1, little
idx_miss	integer	Missing value of grid index	-9999

Table 11-3 Setting variables for raster meshes

Variable	Data type	Description	Default value
name	string	Mesh name	‘source’ or ‘target’
nx_raster	integer	The number of pixels in the longitudinal direction	
ny_raster	integer	The number of pixels in the latitudinal direction	
west	float	West edge of the mesh	
east	float	East edge of the mesh	
south	float	South edge of the mesh	
north	float	North edge of the mesh	
is_south_to_north	boolean	Raster data is arranged from south to north	True
dir	path	Parent directory of the files	‘’
fin_rstidx	file	Input file of raster indices	‘’, int4, 1, little
nx_grid	integer	The number of grids in the longitudinal direction	
ny_grid	integer	The number of grids in the latitudinal direction	
idx_condition	string	Condition that raster indices and grid indices must be fulfilled. {‘match’, ‘grid_in_raster’, ‘raster_in_grid’, ‘none’}	match
idx_miss	integer	Missing value of grid index	-9999

11.5 Products

- Mesh data

Mesh data of LSM and RM, namely raster data and grid data of grid indices.

out/mesh/LSM/{landType}/raster/index.bin
Raster data of grid indices of LSM. {landType} is ‘river’, ‘noriv’, ‘noriv_real’ or ‘noriv_virt’.

out/mesh/LSM/{landType}/raster/index_bnd.bin
Raster data of grid indices of LSM for making boundary data using the script of ILS. {landType} is ‘river’, ‘noriv’, ‘noriv_real’ or ‘noriv_virt’.

out/mesh/LSM/{landType}/grid/index.bin
Grid data of grid indices.

out/mesh/LSM/{landType}/grid/index_bnd.bin
Grid data of grid indices for making boundary data using the script of ILS.

out/mesh/RM/{landType}/raster/index.bin
Raster data of grid indices of RM. {landType} is ‘river’, ‘river_end’, ‘noriv’ or ‘ocean’.

out/mesh/RM/river/raster/basin.bin
Raster data of basin ids for the land type ‘river’.

out/mesh/RM/{landType}/grid/index.bin
Grid data of grid indices of RM.

- Grid data

Grid data such as grid area or representative coordinates of grids are calculated and output to out/mesh.

- CMF river network data

out/mesh/RM_CMF/grid and out/mesh/RM_CMF/raster
CMF river network data. Some of them are used for running CMF or making topography data for CMF using the script in CMF.
Note that the version of the program for constructing river network (FLOW) used here is different from the one included in the latest version of CMF.

catmxy.bin is recorded in 4-byte integer and little endian. Correspondence between values and land types is listed in Table 11-4.

Table 11-4 Land types and values

Land type	Description	Value
river	Catchments	> 1

noriv_coastal	No-river land in coastal regions	0
noriv_inland	No-river land in inland regions	-1
ocean	Ocean	-9999

If you make a new boundary data for MATSIRO in ILS, you will use the script in run/ILS_bnd. In this case, see run/ILS_bnd/sample_cpl_ILS-AOGCM/conf.json since the setting of CMF in this file is consistent to the one made in run/cpl_ILS-AOGCM/sample.

12 Contact

If you have found errors or faced troubles, please contact the developer.

Akira Takeshima
akira.takeshima.f90@gmail.com