

H08 Manual

User's Edition

2nd Edition

Naota Hanasaki



Acknowledgments (First edition for H08 Version 2008)

The H08 water resources model was developed with the help of a great many people. We would like to express our heartfelt gratitude to Professors Taikan Oki and Shinjiro Kanae for the guidance that they have provided over the past 10 years. We would also like to thank Professors Koichi Masuda, Kenichi Motoya, Kenji Tanaka, and Naoki Shirakawa for developing the standard input data and modules for H08, and Tsuyoshi Okazawa, Yuki Koiwa, Toshiyuki Inuzuka, Kazutaka Inaba, and Yadu Pokhrel for testing early versions of H08 and providing extensive feedback.

H08 owes its existence to hydrology, meteorology, agriculture and other disciplines, and to the observation data generated by the efforts of people throughout the world who gathered them. We are accordingly deeply indebted to all those individuals whose achievements contributed to the creation of H08.

This work was supported by the JST CREST project entitled “Long-term Vision for the Sustainable Use of the World's Freshwater Resources” and JSPS KAKENHI Grant Number 23226012. We would also like to thank Mr. Hiroshi Ueno for his help in editing this manual, and Ms. Aoi Takahashi for the cover design.

This is a translation of a manual of the same title from Japanese into English. Translation was funded by the JST-JICA SATREPS project entitled “Integrated Study Project on Hydro-Meteorological Prediction and Adaptation to Climate Change in Thailand” (IMPAC-T).

Acknowledgments (Second edition for H08 Version 2018)

H08 has been fully updated in 2018. The code and manual have been substantially revised accordingly. I would like to express my deep gratitude to my colleagues who supported the development and application of H08. I thank Mr. Yusuke Saito, Mr. Hodaka Kamoshida, Mr. Masashi Fujiwara, Dr. Zhipin Ai, Mr. Menaka Revel, and Hikari Matsuda for checking code and editing the manuals. I would like to thank Mr. Shohei Suzuki, Mr. Yoshiyuki Komazaki, Mr. So Miyahara, Mr. Ryosuke Abe, Mr. Yuya Kanazawa, and Mr. Satoru Ota for developing the data server, which substantially improved the usability of H08.

Version information

August 23, 2010	Source Code and Manual Users Edition Ver. 20100823 released
October 31, 2010	Source Code and Manual Users Edition Ver. 20101031 released (Major update)
January 1, 2012	Source Code and Manual Users Edition Ver. 20120101 released (Major update)
May 1, 2013	Source Code and Manual Users Edition Ver. 20130501 released (Minor update)
Jan 1, 2019	Source Code and Manual Users Edition Second Edition Ver. 20190101 released
Feb 1, 2019	Manual Users Edition SE English proofread
Jul 1, 2019	Ver. 20190701 released (Minor update)

Contents

CHAPTER 1 INTRODUCTION	1
1.1 INTRODUCTION	1
1.2 H08 DOCUMENTATION	1
1.3 DISCLAIMER AND REQUEST TO READERS	1
1.4 VIDEO MANUAL	2
CHAPTER 2 COMPUTER ENVIRONMENT REQUIRED FOR USE OF H08	3
2.1 REQUIRED COMPUTER ENVIRONMENT	3
2.2 REQUIRED FREE SOFTWARE.....	3
CHAPTER 3 INSTALLATION	4
3.1 H08 INSTALLATION	4
3.2 H08 PREFERENCES.....	5
3.3 ENDIAN (FOR ADVANCED USER ONLY).....	5
CHAPTER 4 H08 DIRECTORY STRUCTURE	6
4.1 H08 DIRECTORY STRUCTURE	6
4.2 1ST LEVEL DIRECTORIES	6
4.3 2ND LEVEL DIRECTORIES.....	7
4.4 BIN/ DIRECTORIES.....	7
4.5 PRE/ AND PST/ DIRECTORIES	9
4.6 MAKEFILE.....	9
4.7 3RD LEVEL DIRECTORIES.....	10
CHAPTER 5 H08 INPUT/OUTPUT FILES	11
5.1 DIMENSIONS OF DATA HANDLED BY H08	11
5.2 FILE TYPES HANDLED BY H08.....	11
5.3 EXPRESSING SPACE.....	12
5.4 EXPRESSING TIME.....	14
5.5 H08 FILE FORMAT 2D.....	16
5.6 H08 FILE FORMAT 1D.....	17
CHAPTER 6 CREATING MAP DATA.....	18
6.1 CREATING BASIC SPATIAL INFORMATION.....	18
6.2 CREATING THE MAP DATA REQUIRED TO USE THE LAND MODULE	20
6.3 CREATING THE MAP DATA REQUIRED TO USE THE RIVER MODULE	21
6.4 CREATING THE MAP DATA OF NATIONAL BORDERS AND POPULATION	24
6.5 CREATING THE MAP DATA REQUIRED TO USE THE CROP GROWTH MODULE	24
6.6 CREATING THE MAP DATA REQUIRED TO USE THE RESERVOIR MODULE.....	25
6.7 CREATING THE MAP DATA REQUIRED TO USE THE WATER WITHDRAWAL MODULE.....	27

CHAPTER 7 METEOROLOGICAL DATA	31
7.1 METEOROLOGICAL DATA.....	31
7.2 PREPARING METEOROLOGICAL DATA.....	31
CHAPTER 8 LAND SURFACE PROCESS MODULE	34
8.1 LAND SURFACE PROCESS MODULE MECHANISMS.....	34
8.2 LAND SURFACE PROCESS MODULE CALCULATION PROCESS.....	35
8.3 LAND SURFACE PROCESS MODULE EXECUTION 1: PREPROCESSING.....	36
8.4 LAND SURFACE PROCESS MODULE EXECUTION 2: CALCULATION.....	37
8.5 LAND SURFACE PROCESS MODULE EXECUTION 3: POSTPROCESSING.....	42
CHAPTER 9 RIVER MODULE	44
9.1 RIVER MODULE MECHANISM.....	44
9.2 RIVER MODULE CALCULATION PROCESS.....	44
9.3 RIVER MODULE EXECUTION 1: PREPROCESSING.....	45
9.4 RIVER MODULE EXECUTION 2: CALCULATION.....	45
9.5 RIVER MODULE EXECUTION 3: POSTPROCESSING.....	46
CHAPTER 10 CROP GROWTH MODULE	47
10.1 CROP GROWTH MODULE MECHANISM.....	47
10.2 CROP GROWTH MODULE CALCULATION PROCESS.....	51
10.3 CROP GROWTH MODULE EXECUTION 1: PREPROCESSING.....	52
10.4 CROP GROWTH MODULE EXECUTION 2: CALCULATION.....	53
10.5 CROP GROWTH MODULE EXECUTION 3: POSTPROCESSING.....	54
CHAPTER 11 RESERVOIR OPERATION MODULE	58
11.1 RESERVOIR OPERATION MODULE MECHANISM.....	58
11.2 RESERVOIR OPERATION MODULE CALCULATION PROCESS.....	59
11.3 RESERVOIR OPERATION MODULE EXECUTION 1: PREPROCESSING.....	60
11.4 RESERVOIR OPERATION MODULE EXECUTION 2: CALCULATION.....	61
11.5 RESERVOIR OPERATION MODULE EXECUTION 3: POSTPROCESSING.....	61
CHAPTER 12 ENVIRONMENTAL WATER MODULE	62
12.1 ENVIRONMENTAL WATER MODULE MECHANISM.....	62
12.2 ENVIRONMENTAL WATER MODULE CALCULATION PROCESS.....	62
12.3 ENVIRONMENTAL WATER MODULE EXECUTION METHOD 1: PREPROCESSING.....	63
12.4 ENVIRONMENTAL WATER MODULE EXECUTION METHOD 2: CALCULATION.....	63
12.5 ENVIRONMENTAL WATER MODULE EXECUTION METHOD 3: POSTPROCESSING.....	63
CHAPTER 13 COUPLED MODEL	64
13.1 COUPLED MODEL MECHANISM.....	64
13.2 COUPLED MODEL CALCULATION PROCESS.....	65

13.3 COUPLED MODEL PREPROCESSING.....	66
13.4 COUPLED MODEL CALCULATION.....	66
13.5 COUPLED MODEL POSTPROCESSING.....	69
CHAPTER 14 APPLIED USES	70
14.1 GLOBAL WARMING.....	70
14.2 REGIONAL MODEL.....	72
APPENDIX 1 H08 ANALYSIS TOOLS	73
A1.1 TEXTBOOK.....	73
APPENDIX 2 CODING RULES	74
A2.1 NAMING VARIABLES.....	74
A2.2 TIME LOOPS.....	74
A2.3 COMMENTS.....	74
A2.4 DECLARATION AND INITIALIZATION OF VARIABLES AND ARRAYS.....	75
A2.5 DIMENSION OF ARRAYS.....	75
APPENDIX 3 FILES IN ADM.....	77
A3.1 BACKUP.SH.....	77
A3.2 CLEAN.SH.....	77
A3.3 COUNT.SH.....	77
A3.4 INSTALLER.SH.....	77

Chapter 1

Introduction

This chapter explains what H08 is.

1.1 Introduction

Water is essential to all life forms and indispensable to humans; our survival, daily life, and production activities depend on water. Global water consumption began rising rapidly from the mid-20th century, and is expected to continue to grow, in line with population growth and economic development, at least into the mid-21st century. Global warming is predicted to change rain and snow precipitation patterns, causing snow to melt earlier and promoting aridification in some areas. As a result of these factors, it is expected that an increasing number of regions and people will suffer from water shortages and related problems.

A team of researchers from the National Institute for Environmental Studies (NIES), University of Tokyo, Tokyo Institute of Technology, and other organizations have been working on the development of H08¹, a global water resource model (computer program) for examining such global water resource issues from the present and into the future. This manual explains how to use H08.

1.2 H08 Documentation

This manual is devoted to explaining how to use H08, a large scale computer program, from a user perspective. As explained in more detail in the next chapter, use of H08 requires a UNIX computing environment. We advise those who are not acquainted with UNIX environments to refer to Hanasaki & Nitta (2009).

Readers who wish to learn about H08 in greater detail should refer to Hanasaki et al. (2008a, b). These two papers introduce the background of the development of H08, review preceding research, and explain the theory behind the model, test results, application to global water resource assessment, key issues, and so on. Hanasaki (2006) provides similar information, but should be treated only as background information, as H08 has subsequently undergone fairly extensive improvement.

Features related to virtual water were later improved and expanded, and these changes are covered in Hanasaki et al. (2010). Moreover, features related to groundwater, aqueducts, seawater desalination, and others were expanded, and these changes are covered in Hanasaki et al. (2018). Since the H08 model introduced in this manual now includes these improvements and extensions, Hanasaki et al. (2010) and Hanasaki et al. (2018) can also be referred to.

1.3 Disclaimer and Request to Readers

¹ This model was previously known as H07 for the following reason. The first paper written on this global water resource model was submitted to *Hydrology and Earth System Sciences* in 2007. It was first published very quickly in *Hydrology and Earth System Sciences Discussion* as a discussion paper (one that has not undergone a critical reading). The global water resource model came to be known by the abbreviation H07 as a result of this discussion paper (Hanasaki et al. 2007a, b). The paper was then accepted for publication in *Hydrology and Earth System Sciences* in 2008, becoming Hanasaki (2008a, b), and the model was renamed H08 as a result.

Disclaimer: The developers have developed the source code with the utmost care, but H08 may still contain bugs. The developers have also endeavored to improve H08's performance, but its calculations may differ considerably from reality. In no event shall the developers be liable for any damages suffered by readers as a result of such factors. We ask that users of H08 use the model only at their own risk.

Requests: First, this manual was written by the developers themselves. We have endeavored to make it as easy to understand as possible, but readers who have never used H08 may find some parts difficult to comprehend. In such cases, we ask that you inform us, providing us with as much detail as possible on what you found difficult to understand. Second, H08 harbors limitless scope for improvement and expansion. If you have succeeded in improving or expanding H08 and would like to pass those benefits on to other users, please contact us. After first fully discussing respective rights, we would do our utmost to incorporate such improvements or expansions into subsequent versions of H08.

When citing H08, we request that you use the following citation:

Hanasaki, N., Yoshikawa, S., Pokhrel, Y., and Kanae, S.: A global hydrological simulation to specify the sources of water used by humans, *Hydrology and Earth System Sciences*, 22, 789–817, 2018.

Note that use of H08 for calculation requires the input of great amounts of data. When using such data to conduct research, please be sure to cite each and every source.

1.4 Video Manual

We have prepared an “H08 Video Manual”, which is available at the H08 Homepage (<http://h08.nies.go.jp>) or the URL shown below. [This video manual is for Version 20130501. The video manual for Version 20181001 is under preparation.]

<https://www.youtube.com/channel/UCUSzEjQYcOP-gkXELhrpdBA>

Chapter 2

Computer Environment Required for Use of H08

This chapter explains the computer environment required for the use of H08.

2.1 Required Computer Environment

H08 is composed of Bourne shell scripts and Fortran source code. As such, use of H08 requires a computer environment that meets the following conditions:

1. Supports Bourne shell scripts. Consequently H08 can be installed only on computers running Linux (e.g., Red Hat, CentOS, Ubuntu) or MacOS.
2. Supports Fortran compilers.²

Also, because H08 handles large volumes of input and output data, we recommend using a computer with a hard disk drive with several 100 GB of available space.

2.2 Required Free Software

Installation of the following free software is required to run H08:

0. GMT (The Generic Mapping Tools) (map drawing software).
1. ImageMagick (raster image display and editing software).
2. NetCDF (software for manipulating netCDF format meteorological data).

² Compilers confirmed to date as compatible: Intel Fortran Compiler (commercial), gfortran (free software).

Chapter 3 Installation

This chapter explains how to install H08.

3.1 H08 Installation

The H08 installation procedure is as follows:

0. There is one H08 installation file: `H08_YYYYMMDD.tar.gz`. First, copy the latest tar .gz (or zip) file from the release at <https://github.com/h08model/H08> to the directory where you want to install this program.
Although not recommended, it can also be obtained using git clone.
1. Uncompress the file.

```
% tar zxvf H08_YYYYMMDD.tar.gz
```
2. Move to the directory `H08_YYYYMMDD`.

```
% cd H08_YYYYMMDD
```
3. Execute `install.sh` as follows:

```
% sh ./install.sh
```
4. After a short while, installation will end, with this message appearing:

```
please set adm/Mkinclude  
afterward, continue again
```
5. Create a file named `Mkinclude` in the directory `adm`. A number of sample files (e.g., `Mkinclude.Mac`) are provided in this directory, and you can copy and use them.
6. Edit `Mkinclude`. The main shell variables set in `Mkinclude` are explained below.
Shell variable INC: Specify path to the NetCDF include file³.
Shell variable LIB: Specify path to the NetCDF library file⁴.
Shell variable FC: Specify path to Fortran compiler. For the Intel Fortran Compiler, be sure to include the option `-assume byterecl` as shown below:

```
FC=ifort -assume byterecl5
```
7. Once you have created the `Mkinclude` file in the `adm` directory, execute `install.sh` once again⁶. This will result in the compilation of all the programs.

³ More specifically, you should specify the directory containing `netcdf.inc` and other files. For Linux, this directory is usually `/usr/include` or `/usr/local/include`, and for Mac, `/usr/include` or `/usr/local/include`, or `/sw/include`.

⁴ More specifically, you should specify the directory containing `libnetcdf.a` and other files. For Linux, this directory is usually `/usr/lib` or `/usr/local/lib`, and for Mac, `/usr/lib` or `/usr/local/lib`, or `/sw/lib`.

⁵ When binary data are read, the `recl` (record length) unit can be either the number of bytes or the number of variables, depending on the compiler used. For the Intel Fortran Compiler, it is the number of variables, but the H08 source code has been developed on the assumption that the `recl` unit is bytes. Including this option sets the `recl` unit to bytes.

⁶ For ubuntu users, edit and execute `/${DIRH08}/bin/htdraw.sh` and `htdrawts.sh`, and then `install.sh`.

When you run `install.sh`, warnings will appear. Just ignore them and proceed to the next step.

3.2 H08 Preferences

After installation, configure the computer. First, identify your login shell.

```
% echo $SHELL
```

If `/bin/bash` is displayed as a result, your preferences file is `~/.bashrc`. Carefully read and edit the `adm/sample.bashrc`, and insert it into the end of `~/.bashrc`. Similarly, if `/bin/tcsh` is displayed, your preferences file is `~/.cshrc`. In this case, refer to `adm/sample.cshrc`. The simplest way of configuration is to edit `DIRH08` in your sample file then apply the following command.

```
% cat adm/sample.bashrc >> ~/.bashrc
```

Once configured, the change will take effect from your next login. If you wish to effect the preferences without logging in again, use the following source command.

```
% source ~/.bashrc
```

3.3 Endian (for advanced user only)

Endian is a binary encoding format that exists in two types: big and little. H08 was developed in a Sun Microsystems UltraSPARC CPU environment that uses big-endian, but Intel's CPUs (Itanium 2, Xeon, Core 2 Duo, etc.), which have more users, employ little-endian as their standard. As long as the two types are not used in the same computer, it makes no difference which type is used.

Incidentally, in computers equipped with Intel CPUs, the following lines should be added to the preferences file to write in big-endian binary format:

- If using Intel Fortran Compiler:
 - `setenv F_UFMTENDIAN big` (if login file is `/bin/tcsh`)
 - `export F_UFMTENDIAN=big` (if login file is `/bin/bash`)
- If using gfortran:
 - `setenv GFORTRAN_CONVERT_UNIT big_endian` (if login file is `/bin/tcsh`)
 - `export GFORTRAN_CONVERT_UNIT=big_endian` (if login file is `/bin/bash`)
- If using g95:
 - `setenv G95_ENDIAN big` (if login file is `/bin/tcsh`)
 - `export G95_ENDIAN=big` (if login file is `/bin/bash`)

CHAPTER 4

H08 Directory Structure

This chapter explains H08's directory structure. The content will be very difficult to understand by reading it alone and we recommend that you refer to the source code created in Chapter 3 as you read.

4.1 H08 Directory Structure

H08's directories are basically structured in three layers, called 1st level directories, 2nd level directories, and 3rd level directories, according to the layer. Figure 4-1 shows the structure up to the 2nd level directories.

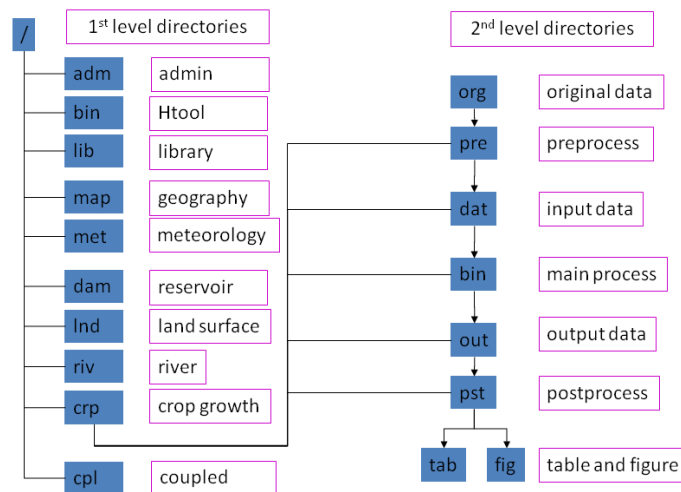


Figure 4-1. Directory structure of H08

4.2 1st Level Directories

Table 4-1 lists the 1st level directories, which represent the overall framework of the elements composing H08.

Table 4-1. 1st level directories

Directory	Contents
adm	Administrator
bin/	H08 Analysis Tools (binary)
lib/	Library
map/	Geographical data
met/	Meteorological data
lnd/	Land model

riv/	River model
crp/	Crop growth model
dam/	Reservoir operation model
cpl/	Coupled model

4.3 2nd Level Directories

Excluding **adm/**, **bin/**, and **lib/**, each of the remaining 1st level directories (i.e., **map/**, **met/**, **lnd/**, **riv/**, **crp/**, **dam/**, and **cpl/**) contain similar 2nd level directories. Table 4-2 lists these 2nd level directories, which show the flow of data processing. Understanding the roles played by the 2nd level directories is of vital importance when using H08.

The most important 2nd level directories are **dat/**, **bin/**, and **out/**. They contain input data, the main process, and output data, respectively. Files in **dat/** are input data that are processed when the main process in **bin/** is executed, with the resulting output data being written to **out/**. Main process input/output files are all coded according to the rules of a format called H08 File Format, which we explain in the next chapter.

Table 4-2. 2nd level directories

Directory	Contents
org/	Original data: Input data in their original format
pre/	Preprocess: Programs to convert data into H08 File Format
dat/	Input data: Input data in H08 File Format
bin/	Main process: Main programs
out/	Output data: Output data in H08 File Format
pst/	Postprocess: Programs to convert data into figures and tables
fig/	Figures
tab/	Tables

Almost no existing data are written in H08 File Format. Meteorological data, for example, are written and distributed mostly in formats such as NetCDF. In such cases, you put these data into **org/** as original data, and then use the program in **pre/** to pre-process them, converting them into H08 File Format and then storing files in **dat/**.

The output files written to **out/** are in H08 File Format (binary), and cannot be read by humans. You must, accordingly, use the program in **pst/** to post-process them into figures and tables, which will be stored respectively in the **fig/** and **tab/** directories.

4.4 bin/ Directories

The **bin/** directories contain files whose names begin with **main** and end with **.f**. These are the Fortran source code of the main process. After compiling these to create an executable file, you can execute this file by supplying a settings file that shows calculation settings. You should also notice the

presence in the same directory of files beginning with `main` and ending with `.sh`. These are Bourne shell scripts for creating the settings files, and are referred to in this text as executable shells. If these files are edited and executed, they automatically create a settings file and execute the main process. In other words, you execute `main*.sh` to run a simulation, and you can edit the content of `main*.sh` if you want to run a simulation under different conditions.

The main process source code (`main*.f`) is designed to call a great many subroutines and functions. The source code of these subroutines and functions is, as a rule, contained in the same directory (so, in the case of `lsm/bin/main.f`, it would be the source code contained in `lsm/bin/`) or in `lib/` (1st level directory).

H08 has five 2nd level directory `bin/` directories, with each containing a `main.sh` file. Table 4-3 shows the main processes and key subroutines for each `main.sh` file. H08 enables the independent use of land, river, crop growth, and reservoir operation modules through respectively executing `lnd/bin/main.sh`, `riv/bin/main.sh`, `crp/bin/main.sh`, or `dam/bin/main.sh`. A coupled model can be used by executing `cpl/bin/main.sh`. The coupled model main process `main*.f` found in `cpl/bin/` is designed to call the subroutines that make up `main*(.f)` contained in `lnd/bin/`, `riv/bin/`, `crp/bin/`, and `dam/bin/`.

Table 4-3. Modules and Models

Modules/Models	Executable shell	Main program	Key subroutines
Land surface	<code>lnd/bin/main.sh</code>	<code>lnd/bin/main</code>	<code>lnd/bin/calc_leakyb.o</code>
River	<code>riv/bin/main.sh</code>	<code>riv/bin/main</code>	<code>riv/bin/calc_outflw.o</code>
Crop growth	<code>crp/bin/main.sh</code>	<code>crp/bin/main</code>	<code>crp/bin/calc_crpuld.o</code>
Reservoir operation	<code>dam/bin/main.sh</code>	<code>dam/bin/main</code>	<code>dam/bin/calc_resope.o</code>
Coupled model	<code>cpl/bin/main.sh</code>	<code>cpl/bin/main</code>	All of the above

4.5 pre/ and pst/ Directories

The 2nd level directories **pre/** and **pst/** also contain many executable shells. **pre/** contains executable shells, named **pre/prep*.sh**, for converting original data written in formats other than H08 File Format into H08 File Format. **pst/** contains executable shells for processing the results of executing the main process, including **pst/calc*.sh** for analyzing the output files in **out/**, **pst/draw*.sh** for plotting images, and **pst/list*.sh** for creating tables. Most of these executable shells call the working program **prog*.f**.

4.6 Makefile

The 2nd level directory **bin/** contains the main process source code (**main.f**). The main process source code is designed to call multiple subroutines and functions. The source code of these subroutines and functions (**calc*.f**, etc.) may be contained in the same directory or in **lib/**. Makefile contains the method for joining multiple source codes to create one executable file.⁷

To compile the source codes in **bin/**, **pre/**, **pst/**, execute the command:

```
% make all
```

This will cause all of the programs⁸ in the directory to be compiled. Incidentally, the command executed in Chapter 3:

```
% sh install.sh
```

is designed to effect **makeall** in all H08 directories.

Executing the command:

```
% make clean
```

deletes the object files (***.o**) and executable files (**main**, etc.) in the directory. You can use this command to change the compiler or compiler options and recompile the programs.

Settings, libraries, and **include** files common to **Makefile** in multiple directories are written in **adm/Mkinclude**. You should notice that when you read **Makefile**, you are using an **include** statement to read in **adm/Mkinclude**.

⁷ For make basics, see Chapter 1 of *Managing Projects with make*, Second Edition by Andrew Oram and Steve Talbott, pub. O'Reilly Media.

⁸ Actually, the programs defined with the shell variable **OBJS**, are located at the start of **Makefile**.

4.7 3rd Level Directories

3rd level directories appear only in **org/**, **dat/**, **out/**, **fig/**, and **tab/**. The 3rd level directories in **org/** are displayed as the abbreviation of the data provider. For example, data provided by Hanasaki et al. (2006) will be shown as H06. The 3rd level directories in **dat/**, **out/**, **fig/**, and **tab/** are displayed as abbreviations of variables. For example, temperature data in **met/dat/** will be **met/dat/Tair_____**. As with this example, the length of the names of 3rd level directories in **dat/**, **out/**, **fig/**, and **tab/** is fixed at eight characters. Changing the length can result in errors, so exercise care. We explain the different kinds of 3rd level directory names in detail in Chapter 6 and onwards.

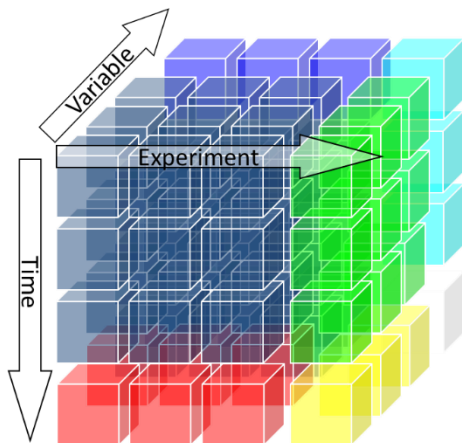
CHAPTER 5

H08 Input/Output Files

This chapter explains H08's input/output files.

5.1 Dimensions of Data Handled by H08

Though it will take us a little out of our way, we would like to consider the dimensions of the data handled by H08 as a departure point for explaining input/output files. How many spatial and temporal dimensions does our world have? Thinking conventionally, the answer would be three spatial dimensions and one temporal dimension, making a total of four, but we would like you to imagine an additional two dimensions. First, when you conduct a simulation using H08, you pose the question, "What if the world was ...?" with respect to the same time and place, and so let us add this separate world as a dimension called, for the sake of convenience, "experiment". Next, quantities such as temperature, precipitation, and various other factors in a certain place and time can all be lumped together and added as yet another dimension that, again for the sake of convenience, we call "variable". Totaling these dimensions, we can see that H08 handles data in six dimensions (Figure 5-1).



- Space: 3D (x,y,z)
- Time: 1D (t)
- Experiment: 1D (e)
(Each experiment expresses different world)
- Variable: 1D (v)
(The world is a superposition of variables)
- Total: 6D (x,y,z,t,e,v)

Figure 5-1. Dimensions of data handled by H08

5.2 File Types Handled by H08

If we suppose, for argument's sake, that today's computers could handle the above six-dimensional array, all H08 input and output data could be contained in one file. However, the size of such a file would likely be anything from several dozen GB to several TB, which is too big to be handled by a conventional computing configuration. Therefore, we divide files into more easily handled sizes for storage and use in H08.

There are basically only two methods for dividing six-dimensional data in an effective way from the perspective of hydrology or water resources. One method is to divide the data into four fixed dimensions of time, experiment, variable, and altitude, and two flexible dimensions of latitude and longitude for

geographically distributed data. The other method is to divide the data into five fixed dimensions — the three spatial dimensions plus experiment and variable — and one flexible dimension of time. The data for the former are best in binary (4-byte real numbers), and for the latter, text. Whichever method is used, the format is very strictly fixed within H08, and from here on, we will refer accordingly to the former as H08 File Format 2D, and to the latter as H08 File Format 1D.

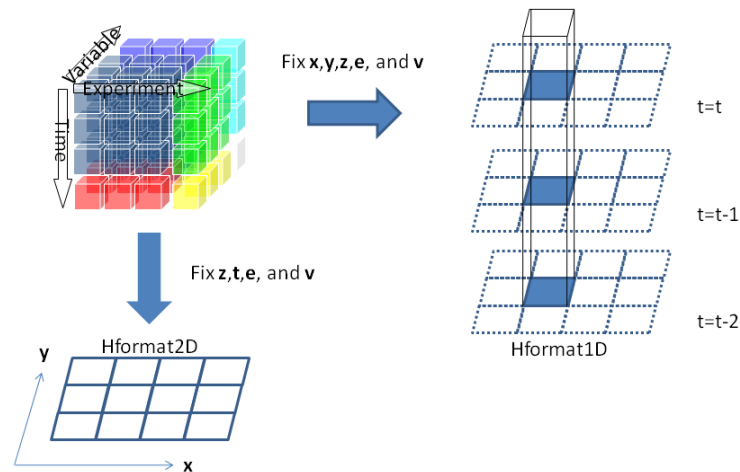


Figure 5-2. H08 File Format 1D and H08 File Format 2D

5.3 Expressing Space

In the previous section, we considered the dimensions of time and space. In this section, we will look at how to display space. Let us consider the case in which the spatial domain is global, and the spatial resolution is 45° latitude \times 45° longitude (Figure 5-3). A global spatial domain can be displayed by setting the maximum and minimum longitude (lonmax and lonmin) to 180° and -180° , and the maximum and minimum latitude (latmax and latmin) to 90° and -90° . Assuming equal intervals, the spatial resolution can be displayed by setting the number of cells at eight in the longitudinal axis (n_x), and four in the latitudinal axis (n_y).

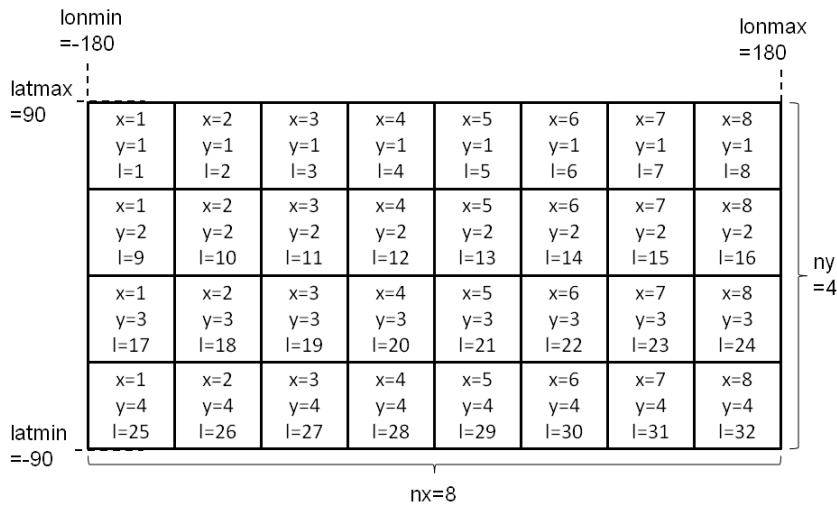


Figure 5-3. Example of a spatial domain

Because expressing space in terms of latitude and longitude is inconvenient for calculation, we have set X and Y coordinates as shown in Figure 5-3. The X coordinates extend from LONMIN to LONMAX in the cells of the grid, with X taking the value of 1 up to NX. The Y coordinates extend from LATMAX to LATMIN in the cells of the grid (note the direction), with Y taking the value of 1 up to NY. The L coordinates are a combination of X and Y coordinates, and can generally be expressed as follows:

$$L = NX \times (Y - 1) + X \quad (5-1)$$

L takes the value of 1 to NL (= NX × NY). Arranged in this way, a grid cell can be expressed either in terms of two coordinates – X and Y – or with a single L coordinate. Conversion tables for converting between these two expression methods can be created. Several methods exist, but H08 uses one conversion table to convert L coordinates into X coordinates and another to convert L coordinates into Y coordinates.

Summing up the above, space in H08 is defined by the nine variables listed in Table 5-1.

Table 5-1. Variables defining the spatial domain

Variable (Shell script)	Variable (Fortran)	Explanation
LONMIN	n0lonmin	Minimum value in longitude
LONMAX	n0lonmax	Maximum value in longitude
LATMIN	n0latmin	Minimum value in latitude
LATMAX	n0latmax	Maximum value in latitude

NX	n0x	Division of longitudinal
NY	n0y	Division of latitudinal
NL	n0l	Total number of grid cells
L2X	c0l2x	L to X conversion table
L2Y	c0l2y	L to Y conversion table

Incidentally, L coordinates may be defined by methods other than equation 5-1. For example, because we are seeking to understand the hydrology and water resources of the world's land areas, sea areas are irrelevant to our calculations. If, as in Figure 5-4, only eight of the 32 grid cells are land, L coordinates may be allocated only to the land cells. For example, the data distributed with the Global Soil Wetness Project (GSWP2; Dirmeyer et al., 2006) were global $1^\circ \times 1^\circ$, but because sea area cells contain only missing values, the data consist not of all cells ($NX \times NY = 360 \times 180 = 64,800$) but of just the land cells ($NL = 15,238$). Thus allocating L coordinates only to land cells can reduce the amount of data by about one third. However, in such cases, the relationship between the L, X, and Y coordinates is not as simple as in equation 5-1, making the L2X and L2Y files important.

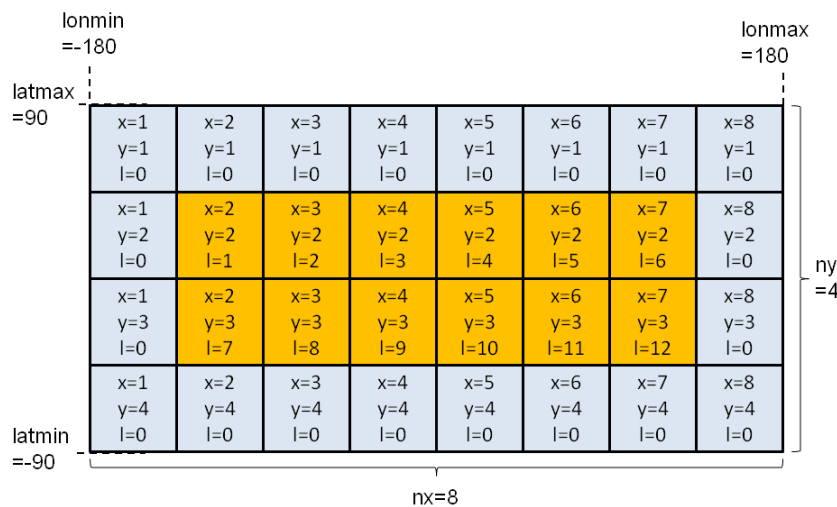


Figure 5-4. Example of a spatial domain allocating L to land only

5.4 Expressing Time

We will now consider how to express time. H08 uses the Gregorian calendar, that is, the standard calendar including leap years. It expresses time as year, month, day or year, month, day, hour. For example, August 23, 2010 is expressed as:

20100823

And 15:00 on August 23, 2010 as:

2010082315

H08 does not at present have the ability to write time intervals of less than one hour to a file.

The times shown for data can be for instantaneous values or average values. An instantaneous value is the quantity at a particular instant in time, while an average value is the average quantity over a specific period. H08 uses instantaneous values to express state variables (soil moisture, river channel storage), and average values to express flux variables (runoff, river flow). For average values, the time shown is the last time in the period employed to derive the average (Figure 5-5). Accordingly, the conservation law is expressed as:

$$S_t - S_{t-1} = F_t \times \Delta t$$

where S is state variables, F is flux variables, and Δt is the time interval.

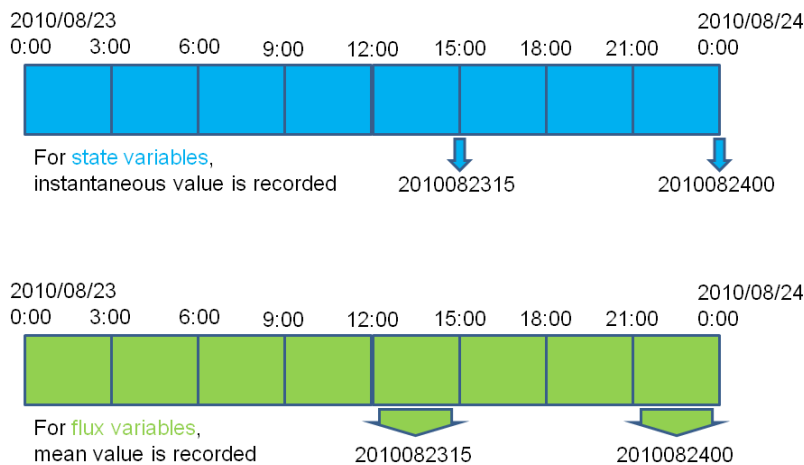


Figure 5-5. Time and state/flux variables

Next, we will consider annual/monthly/daily average values. If, for example, we have calculated in 3-hourly intervals for the year from January 1, 2010, to January 1, 2011, then the annual/monthly/daily means for state and flux variables for 2010 are defined as in Table 5-2.

Table 5-2. Annual/Monthly/Daily means for state and flux variables for 2010

Means	Expression	State variables	Flux variables
Annual mean	20100000	Quantity at 2011/1/1 0:00	Average from 2010/1/1 0:00 to 2011/1/1 0:00

Monthly mean	20100100	Quantity at 2010/2/1 0:00	Average from 2010/1/1 0:00 to 2010/2/1 0:00
Daily mean	20100101	Quantity at 2010/1/2 0:00	Average from 2010/1/1 0:00 to 2010/1/2 0:00
3-hourly mean	2010010100	Quantity at 2010/1/1 0:00	Average from 2009/12/31 21:00 to 2010/1/1 0:00

5.5 H08 File Format 2D

This section will explain the H08 File Format 2D file format, file names, and file extensions.

H08 File Format 2D files are binary files using 4-byte real numbers. They have no header information, and consist only of rows of real numbers in the same number as the cells. The real numbers are arranged in the same order as the L coordinates.

File names are either 20 or 22 characters in length, with the characters representing the project name (four characters), experiment name (four characters), year (four characters), month (two characters), day (two characters), hour (two characters), and file extension (four characters). As mentioned earlier, the time, experiment, variable, and altitude dimensions are fixed in H08 File Format 2D. Specifying the year, month, day, and hour fixes the time, while specifying the project name and experiment name fixes the experiment dimension. Specifying the relevant 3rd level directory name (8 characters) fixes the variable dimension. H08 File Format 2D file names, accordingly, express fixed dimensions, and the file name length is strictly fixed. Failing to observe this rule can result in errors, and the program will crash, so you should exercise care.

File extensions are used to signify the spatial domain or resolution, and can be decided at your discretion as long as they consist of four characters. The developers use those shown in the table below.

Table 5-3. Examples of file extensions

File extensions	Spatial domain	Spatial resolution
.one	Global	1.0°×1.0°
.h1f	Global	0.5°×0.5°
.g15	Global	5'×5'
.cp5	The Chao Phraya River	5'×5'

To sum up the above, the name of a file containing, for example, the global temperature data at a 1°×1° spatial resolution on January 1, 2001, at 12:00 (UTC) for a project named CRU and experiment named 2.0 would be as follows:

Tair____/CRU_2.0_2001010112.one

5.6 H08 File Format 1D

This section will explain the H08 File Format 1D file format, file names, and extensions.

H08 File Format 1D files are text files and, in the case of per-day data, the 1st column is the year, 2nd the month, 3rd the day, and 4th the data. Columns are separated by a tab or one or more spaces.

File names are 20 characters in length, with the characters representing the project name (four characters), experiment name (four characters), location ID (eight characters), and file extension (four characters). As mentioned earlier, the spatial, experiment, and variable dimensions are fixed in H08 File Format 1D. Specifying the location ID fixes the spatial dimensions, while specifying the project name and experiment name fixes the experiment dimension. Specifying the relevant 3rd level directory name (eight characters) fixes the variable dimension. Unlike H08 File Format 2D files, the only file extension is `.txt`.⁹

To sum up the above, the name of a file containing, for example, a time series of air temperature data for a point located at L coordinate 12,345 for a project named CRU and experiment named 2.0 would be as follows:

```
Tair____/CRU_2.0_00012345.txt
```

⁹ Because time resolution information cannot be gleaned from the file name with this method, we are considering changing the extension to `.yr` for yearly data, and `.mo` for monthly data.

CHAPTER 6

Creating Map Data

H08 input data consist of map data and meteorological data. This chapter explains how to create map data.

6.1 Creating Basic Spatial Information

Of the basic spatial information shown in Table 5-1 in chapter 5, we explain here how to create the two files (**L2X** and **L2Y**) required to convert L coordinates to X and Y coordinates. We will also create a file that displays the area of each cell that is derived solely from basic spatial information. The specific steps for accomplishing these tasks are provided below.

Table 6-1. Basic spatial information to be created

File name	Note	Directory	Unit
L2X, L2Y	L to X, L to Y converter	<code>map/dat/12x_12y/</code>	-
GRDARA	Grid cell area	<code>map/dat/grd_ara/</code>	m ²

0. Change directory to `map/pre/`.
1. Check the settings of `prep_basmap.sh`.
2. Execute `prep_basmap.sh` as follows:

```
% sh prep_basmap.sh
```

As a result, the files `12x.h1f.txt` and `12y.h1f.txt`, for converting L coordinates into X and Y coordinates, respectively, will be outputted to the directory `map/dat/12x_12y/`. Another file containing area information for each cell (`grdara.h1f`) will be outputted to the directory `map/dat/grd_ara/`.

3. Similarly, change parameters and execute `prep_basmap.sh` for `.one` and `.g15` (Table 5.3; see `adm/sample.bashrc` for geographical parameters).

• Column 1

• Checking Output Data 1: Analyzing Map Data

As you learned in the last chapter, the file created through the above processes, `map/dat/grd_ara_/grdara.hlf`, is written in H08 File Format 2D. Let's take a look at this file. First check the file size to make sure that `grdara.hlf` was outputted correctly. In cases covering the whole world at a resolution of $0.5^\circ \times 0.5^\circ$, $NL = 720 \times 360 = 259,200$. Because recording one real number requires 4 bytes, the file size should be $259,200 \times 4 = 1,036,800$ bytes. Use the `ls -l` command to check that this is the actual size.

Next, take a look inside `grdara.hlf`. Files written in H08 File Format can be easily analyzed and plotted using H08 Analysis Tools. See Saito and Hanasaki (2012) for details. Using H08 Analysis Tools as is requires a great many arguments here, so to save space below, we use the shortened input for H08 Analysis Tools provided in Appendix A of Saito and Hanasaki (2012) (NB: it works when you have completed the H08 environment setting shown in Chapter 3.2). If you want to know the sum of all cells, or in other words, the surface area of the Earth, write:

```
% cd ${DIRH08}/map/pre/
% sumhlf ../../map/dat/grd_ara_/grdara.hlf
```

You should obtain 5.09×10^{14} as the result. Because this area is expressed in units of m^2 , the calculated surface area of the earth is about 5.09×10^8 km^2 . If you want to know the maximum (or minimum) area of each cell, write:

```
% maxhlf ../../map/dat/grd_ara_/grdara.hlf
% minhlf ../../map/dat/grd_ara_/grdara.hlf
```

You should obtain 3.08×10^3 km^2 and 13.58 km^2 . If you want to draw a world map, you can easily draw a map like the one shown in Figure 6-1 by writing:

```
% gmt makecpt -T0/4000000000/1000000000 -Z > temp.cpt
% hlf2eps ../../map/dat/grd_ara_/grdara.hlf temp.cpt temp.eps
% htconv temp.eps temp.tif rot
% display temp.tif
```

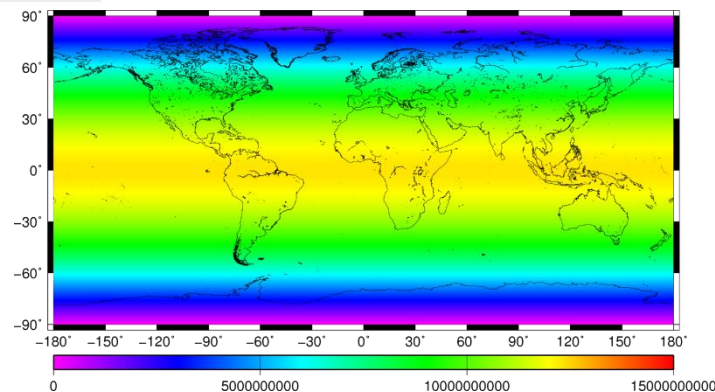


Figure 6-1. A world map of `grdara.hlf` [m^2]

6.2 Creating the Map Data Required to Use the Land Module

The map data shown in Table 6-2 are required to use the land surface process module. Use the process detailed below to create these data.

Table 6-2. Data required to use the land module

File name	Note	Directory	Unit
LNDMSK	Land sea mask	<code>map/dat/lnd_msk_</code>	-
LNDARA	Land area	<code>map/dat/lnd_ara_</code>	m ²
ALBEDO	Albedo	<code>map/dat/Albedo</code>	-
SLPCLS	Slope class	<code>map/dat/slp_cls_</code>	-
GEOCLS	Geological class	<code>map/dat/geo_cls_</code>	-
PRMMSK	Permafrost mask	<code>map/dat/prm_msk_</code>	-
SOITYP	Soil type	<code>map/dat/soi_typ_</code>	-

0. Download `Data_Global_2018.yyyymmdd.tar.gz` from “Manuals” in the H08 website. All input data required to execute this manual is included in this file. Decompress `Data_Global_2018.yyyymmdd.tar.gz` and put all data folders in the `Data_Global_2018.yyyymmdd` directory (e.g. `WFDEI/`, `AQUASTAT/`, etc) into `$(DIRH08)/map/org`. If you do not have `map/org`, then make a new directory. Now you will have `map/org/WFDEI`.
 1. Change directory to `map/pre/`.
 2. Check the settings and execute `prep_lnd_WFDEI.sh`. `WFDEI` in the file name signifies that land and sea distribution conforms to `WFDEI` data.
 3. Prepare global monthly mean albedo data. Check that there is `GSWP2_Albedo/` in `map/org`. These data were prepared using the global monthly albedo data for 1986–1995, which were distributed for the Global Soil Wetness Project Phase 2 (`GSWP2`).
 4. Check the settings and execute `prep_lnd_GSWP2_Albedo.sh`.
 5. Prepare global slope data. Check that there is `FAO2009_Slope` in `map/org`. These data were prepared using the slope data in the Harmonized World Soil Database v1.2 by the Food Agriculture Organization (FAO).
 6. Check the settings and execute `prep_lnd_FAO2009_Slope.sh`.
 7. Prepare global geology data. Check that there is `OneGeology/` in `map/org`. These data were prepared using the World CGMW 1:50M Geological Units Onshore by One Geology.
 8. Check the settings and execute `prep_lnd_OneGeology.sh`.
 9. Prepare global permafrost data. Check that there is `NSIDC/` in `map/org`. These data were prepared using the permafrost data by the National Snow Ice Data Center.
 10. Check the settings and execute `prep_lnd_NSIDC_mercator.sh`.
 11. Prepare soil type data. Check that there is `GSWP3_SoilType/` in `map/org`. These data were prepared using the Global Soil Wetness Project Phase 3.

12. Check the settings and execute `prep_lnd_GSWP3_Soiltype.sh`. The legend of soil type ID will be added in `map/dat/soi_typ_`.

As a result of doing the above, a file that divides the spatial domain into land and sea (`lndmsk.WFDEI.hlf`) is created in the directory `map/dat/lnd_msk_`. Figure 6-2 shows the result of plotting this file with `htdraw` (short form is `hlf2eps`). Land domain cells contain 1, and sea domain cells contain 0. You can see from this map that Antarctica is treated as sea.

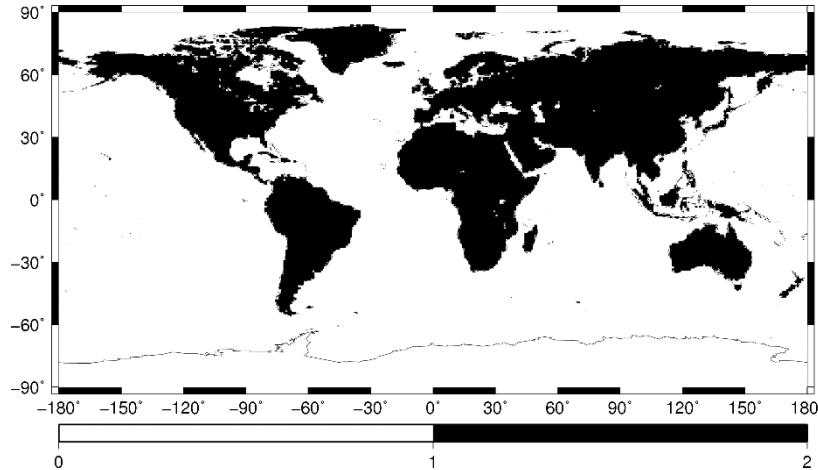


Figure 6-2. A map of the land-sea mask. A value of 1 indicates land, and 0 indicates sea.

A file containing land area (`lndara.WFDEI.hlf`) is also created in the directory `map/dat/lnd_ara_`. Figure 6-3 shows the result of plotting this file. Note that sea domain cells contain the value assigned to missing data (10^{20}).

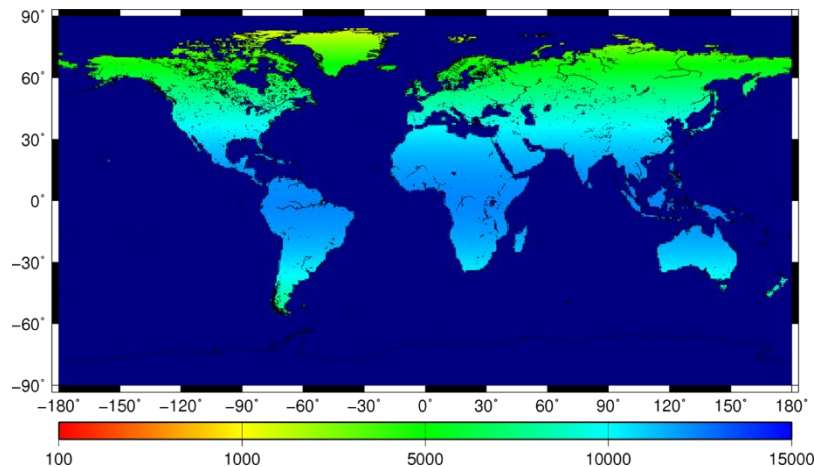


Figure 6-3. A map of land area [km²]. Note that sea domain cells contain the value assigned to missing data (10^{20}).

6.3 Creating the Map Data Required to Use the River Module

Tables 6-3 and 6-4 show the items required for the river module. Use the process detailed below to create these data.

Table 6-3. Map data required to use the river module

File name	Note	Directory	Unit
FLWDIR	Flow direction	<code>map/dat/flw_dir/</code>	-
RIVSEQ	River sequence	<code>map/out/riv_seq/</code>	-
RIVNXL	The next downstream cell's L coordinate	<code>map/out/riv_nxl/</code>	-
RIVNXD	Distance to the next downstream cell	<code>map/out/riv_nxd/</code>	m

Table 6-4. Useful map data in the river module

File name	Note	Directory	Unit
RIVARA	Catchment area	<code>map/out/riv_ara/</code>	m ²
RIVNUM	River ID	<code>map/out/riv_num/</code>	-

0. Change directory to `map/pre/`.
1. Edit and execute `prep_riv_WFDEI.sh`.

This will result in the creation of a flow direction file (`flwdir.WFDEI.hlf`) in the directory `map/dat/flw_dir/`. As shown in Figure 6-4, this file shows flow direction for each cell with the numbers 0 to 9. Values from 1 to 8 show flow direction in 45° segments from north to northwest. The numbers 0 and 9 are special, with 0 signifying that a cell is sea, and 9 signifying river mouth.

Next,

2. Change directory to `map/bin/`.
3. Check the settings and execute `main_riv.sh`.

This results, initially, in the creation of a river sequence file (`rivseq.WFDEI.hlf`) in the directory `map/out/riv_seq/`. In this file, 1 signifies headwaters, 2 is immediately downstream from headwaters, 3 is downstream from this downstream section, and so forth, with the number rising by 1 for successive downstream cells. Next, a file containing the L coordinates for downstream cells (`rivnxl.WFDEI.hlf`) is created in directory `map/out/riv_nxl/`. This file shows the L coordinate of the next downstream cell. Finally, a file giving distance to the next downstream cell file (`rivnxd.WFDEI.hlf`) is created in the directory `map/out/riv_nxd/`. This file provides direct distances from the center of a cell to the center of the next downstream cell.

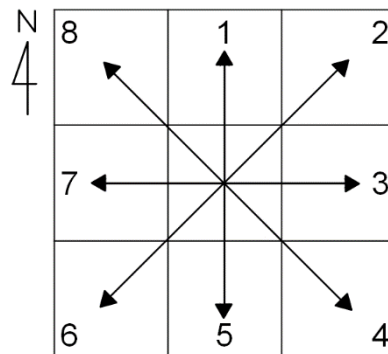


Figure 6-4. Flow direction for each cell with the numbers 0 to 9. Numbers 1 to 8 show flow direction in 45° segments from north to northwest. The numbers 0 and 9 are special numbers, with 0 signifying that a cell is sea, and 9 signifying river mouth.

Though they are not essential for using the river module, two additional files shown in Table 6-4 that are useful for analysis are created. The catchment area file (**rivara.WFDEI.hlf**) created in the directory **map/out/riv_ara_** shows the catchment area for the river above each cell location. In the case of the headwater cell, the area is the area of that cell. The river ID file created in the directory **map/out/riv_num_** allocates a unique ID to each river in order of catchment area size. Thus, in a global map, the Amazon's ID is 1, the Congo is 2, and the Mississippi is 3.

Table 6-5. Major basins in the world

ID	Name	ID	Name
1	Amazon	11	Niger
2	Congo	12	Mackenzie
3	Mississippi	13	Ganges-Brahmaputra
4	Nile	14	Lake Chad (the Chari River)
5	Parana	15	Volga
6	Yenisei	16	Zambezi
7	Ob	17	Nelson
8	Lena	18	Lake Eyre
9	Yangtze	19	Murray Darling
10	Amur	20	St Lawrence

6.4 Creating the Map Data of National Borders and Population

Information such as national borders and population is indispensable in assessment of water resources. The files shown in Table 6-6 are required for basic geographic analysis. Use the process detailed below to create these files.

Table 6-6. Basic geographic data

File name	Note	Directory	Unit
NATMSK	Nation mask	<code>map/dat/nat_msk_</code>	-
NATCOD	Nation code	<code>map/dat/nat_cod_</code>	-
POPTOT	Total population	<code>map/dat/pop_tot_</code>	-

1. To prepare national border and population data, check that there is `C05/` in `map/org`. These data were prepared using the Gridded Population of the World (GPW) version 3 and provided by the Center for International Earth Science Information Network (CIESIN).
2. Check the settings and execute `map/pre/prep_map_C05_nat.sh`.
3. The nation mask is written out in `map/dat/nat_msk_` and the list of nations is written out in `map/dat/nat_cod_`.
4. Check the settings and execute `map/pre/prep_map_C05_pop.sh`.
5. The total population is outputted to `map/dat/pop_tot_`.

6.5 Creating the Map Data Required to Use the Crop Growth Module

The five files shown in Table 6-7 are required to use the crop growth module. Use the process detailed below to create these files.

Table 6-7. Map data required to use the crop growth module

File name	Note	Directory	Unit
IRGARA	Irrigated area	<code>map/dat/irg_ara_</code>	m ²
CRPINT	Cropping intensity	<code>map/dat/crp_int_</code>	-
IRGEFF	Irrigation efficiency	<code>map/dat/irg_eff_</code>	-
CRPARA	Cropland area	<code>map/dat/crp_ara_</code>	m ²
HVSARA	Harvested area according to crop type	<code>map/dat/hvs_ara_</code>	m ²

0. To prepare cropping intensity, cropland area, crop type, and irrigated area, check that there is `DS02/`, `R08/`, `M08/`, `S05/` in `map/org`. These files were prepared using Döll and Siebert (2002), Ramankutty et al. (2008), Monfreda et al. (2008), and Siebert et al. (2005).

1. Change directory to `map/pre/`.
2. Check the settings and execute `prep_crp_R08M08S05.sh`.

This results in the outputting of Ramankutty et al. (2008) cropland area, Monfreda et al. (2008) harvested area, and Siebert et al. (2005) irrigated area files in the directories `map/dat/crp_ara_`, `map/dat/pas_ara_`, `map/dat/hvs_ara_`, and `map/dat/irg_ara_`, respectively.

3. Check the settings and execute `prep_crp_DS02.sh`.

This results in the output of Döll and Siebert (2002) irrigated area, cropping intensity, and irrigation efficiency files in the directories `map/dat/irg_ara_`, `map/dat/crp_int_`, and `map/dat/irg_eff_`, respectively.

4. Change directory to `map/bin`.
5. Check the settings and execute `calc_crptyp.sh`.

This results in the output of the crop types of the largest and the second largest harvested area into `map/out/crp_typ1` and `map/out/crp_typ2`, respectively.

6. Check the settings and execute `calc_crpfrc.sh`.

This results in the output of the fractions of (1) double crop irrigated cropland, (2) single crop irrigated cropland, (3) rainfed cropland, and (4) other land use into `map/out/irg_frcd`, `map/out/irg_frcs`, `map/out/rfd_frc_`, and `map/out/non_frc_`, respectively.

6.6 Creating the Map Data Required to Use the Reservoir Module

The five files shown in Table 6-8 are required to use the reservoir module. The six files in Table 6-9 are also useful for understanding the features of the data. Use the process detailed below to create these files.

Table 6-8. Map data required to use the reservoir module

File name	Note	Directory	Unit
<code>DAMCAP</code>	Reservoir capacity	<code>map/dat/dam_cap_</code>	kg
<code>DAMID_</code>	Reservoir ID	<code>map/dat/dam_id_</code>	-
<code>DAMP RP</code>	Primary purpose of reservoir	<code>map/dat/dam_prp_</code>	-
<code>DAMS RF</code>	Surface area of reservoir	<code>map/dat/dam_srf_</code>	m ²
<code>DAMALC</code>	Water demand allocation for each reservoir	<code>map/out/dam_alc_</code>	-

Table 6-9. Useful map data for reservoirs

File name	Note	Directory	Unit
DAMNUM	Number of reservoirs in a grid cell	map/dat/dam_num_	-
DAMYR_	Reservoir construction year	map/dat/dam_yr_	-
DAMD2D	Reservoir governing area 1 (dam to dam)	map/out/dam_d2d_	-
DAMD2S	Reservoir governing area 2 (dam to sea)	map/out/dam_d2s_	-
DAMUP_	Number of reservoirs in upper stream	map/out/dam_up_	-
DAMUPC	Capacity of reservoirs in upper stream	map/out/dam_upc_	kg

0. Carry out the processes through Chapter 9, and finish the river flow calculations.

1. Check that there is **Grand/** in **map/org**.
2. Change directory to **map/pre/**.
3. Check the settings and execute **prep_dam_GRanD.sh**.

This results in the conversion of reservoir positional data (text data containing reservoir ID, capacity, purpose, surface area, and other information) to H08 File Format 2D format in the directory **map/dat**.

4. Change directory to **map/bin/**.
5. Check the settings and execute **main_dam.sh**.

DAMDBG (the ID of dam to debug) must be a valid value .

Here we need to specify which year's discharge is to be used in dam operation. You can set the mean monthly discharge of an experimental period. You can calculate the mean discharge using **riv/pst/calc_mean.sh**.

This results in the creation of several reservoir information items. Initially, a file containing water demand allocation information for each reservoir (***.hlf**) is created in the directory **map/out/dam_alc_**. Put another way, this file provides information on which upstream reservoir a certain cell's water demand has been allocated to. Next, reservoir governing files (***.hlf**) are created in the directories **map/out/dam_d2d_** and **map/out/dam_d2s_**. The first file shows the cells whose water demand is catered for by specific reservoirs, while the second file shows the cells down to the river mouth. Finally, a file showing the total number of reservoirs upstream of specified locations is created in **map/out/dam_up_**, and a file showing total capacity of those reservoirs is created in **map/out/dam_upc_**.

6.7 Creating the Map Data Required to Use the Water Withdrawal Module

The twenty files shown in Table 6-10 are required to consider industrial and domestic water withdrawal. Use the process detailed below to create these files.

Table 6-10. Map data required to use the water withdrawal module

File name	Note	Directory	Unit
DEMAGR	Agricultural water demand	<code>Ind/out/DemAgr_</code>	kg s ⁻¹
DEMIND	Industrial water demand	<code>map/dat/dem_ind_</code>	kg s ⁻¹
DEMDOM	Domestic water demand	<code>map/dat/dem_dom_</code>	kg s ⁻¹
WITAGR	Agricultural water withdrawal	<code>map/dat/wit_agr_</code>	kg s ⁻¹
WITIND	Industrial water withdrawal	<code>map/dat/wit_ind_</code>	kg s ⁻¹
WITDOM	Domestic water withdrawal	<code>map/dat/wit_dom_</code>	kg s ⁻¹
FRCGWAGR	GW fraction for agricultural water	<code>map/dat/frc_gwa_</code>	-
FRCGWIND	GW fraction for industrial water	<code>map/dat/frc_gwi_</code>	-
FRCGWDOM	GW fraction for domestic water	<code>map/dat/frc_gwd_</code>	-
AEI	Area equipped for irrigation	<code>map/dat/aei_____</code>	m ²
AEIG	Area equipped for groundwater irrigation	<code>map/dat/aeig_____</code>	m ²
AEIS	Area equipped for surface water irrigation	<code>map/dat/aeis_____</code>	m ²
SUPAGRGT	Total groundwater abstraction for agricultural sector	<code>map/dat/SupAgrGT</code>	kg s ⁻¹
SUPINDGT	Total groundwater abstraction for industrial sector	<code>map/dat/SupIndGT</code>	kg s ⁻¹
SUPDOMGT	Total groundwater abstraction for domestic sector	<code>map/dat/SupDomGT</code>	kg s ⁻¹
FRCGWA	Fraction of groundwater to total water demand in the agricultural sector	<code>map/dat/frc_gwa_</code>	-
FRCGWI	Fraction of groundwater to total water demand in the industrial sector	<code>map/dat/frc_gwi_</code>	-
FRCGWD	Fraction of groundwater to total water demand in the domestic sector	<code>map/dat/frc_gwd_</code>	-
CANORG	Origin (Source) of aqueducts	<code>map/dat/can_org_</code>	
CANDES	Destination (Recipient) of aqueducts	<code>map/dat/can_des_</code>	

0. To prepare water use data, check that there is `AQUASTAT/` in `map/org`. This file was prepared using the FAO AQUASTAT database.

1. Change directory to `map/pre/`.
2. Edit and execute `prep_map_AQUASTAT.sh`. This results in creating industrial and domestic water demand and withdrawal.
3. To prepare irrigated area equipped with groundwater irrigation, check that there is `GMIA5/` in `map/org`. This file was prepared using the Global Map of Irrigated Area version 5 by Siebert et al. (2010).
4. Check the settings and execute `prep_map_GMIA5.sh`. This results in creating water source-specific irrigated area information.
5. Check the settings and execute `prep_map_GMIA5_aux.sh`. This results in creating auxiliary information on water source-specific irrigated area.
6. To prepare groundwater use data, check that there is `IGRAC/` in `map/org`. This file was prepared using the country-wise groundwater use database by the International Groundwater Resources Assessment Centre.
7. Check the settings and execute `prep_map_IGRAC.sh`. This results in creating sector-specific groundwater abstraction information.
8. Check the settings and execute `prep_map_frcgw_doll.sh`. This results in creating the fraction of groundwater to total water demand. The method was inspired by Döll et al. (2012).
9. To prepare global aqueduct network data, check that there is `K14/` in `map/org`. This file was prepared using the country-wise groundwater use database by Kitamura et al. (2014).
10. Check the settings and execute `prep_map_lcan.sh`. This results in creating global information on aqueducts, generating a global “implicit” aqueduct network. Here implicit means that these aqueducts are not identified by literature but inferred from geological characteristics. See Hanasaki et al. (2018) for further details.
11. Check the settings and execute `prep_map_K14.sh`. This results in generating a global map of “explicit” aqueducts. Here explicit means that these aqueducts are identified by literature collected by Kitamura et al. (2014). At the same time, a map of aqueducts is created that integrates explicit and implicit aqueducts.
12. To prepare GDP data, check that there is `IIASA_SSP/` in `map/org`.
13. Check the settings and execute `prep_map_IIASA_SSP.sh`. This results in generating GDP data of Shared Socioeconomic Pathways.
14. To prepare coast line data, check that there is `MISC_Maps/` in `map/org`.
15. Check the settings and execute `prep_map_cstlin.sh`. This results in generating the global coast line data, which is essential for seawater desalination modeling.
16. Before moving on to the next step, finish Chapter 8 and Chapter 10 (until 10.3) since simulated evapotranspiration is needed to proceed.
17. Check the settings This results in generating the area utilizing seawater desalination (AUSD; Hanasaki et al. 2016).

•Column 2

● **Editing shell scripts**

From H08_20111130, each shell script is subdivided into four parts:

1. **Header**
2. **Settings A** (all users can/should edit)
3. **Settings B** (only experts can/should edit)
4. **Job**

About **Header**: you don't need to edit it at all. It is a good idea to put in a memorandum if you have considerably changed the script.

Settings A includes **Basic Settings**, **Geographical Settings**, and others. If you are new at H08, and you wish to run H08 with some different settings, edit here.

Settings B includes **Output**, **Output Directory**, **Macro**, and others. If you modify the code or scripts, you might need to change these parts for consistency.

About **Job**: only experts should modify this. All of the lines here are important for H08 simulation.

CHAPTER 7

Meteorological Data

H08 input data consist of map data and meteorological data. In this chapter, we will explain the meteorological data.

7.1 Meteorological Data

Table 7-1 shows the variables (eight meteorological variables) required for H08 calculations. The variable abbreviations, names, units, and signs (direction of positive values) used here conform to the ALMA conventions Version 3¹⁰. A number of bodies of meteorological data covering the whole world and including all of these variables have been developed. In this manual, we use the data developed by the WATCH Forcing Data methodology applied to ERA-Interim reanalysis data (WFDEI; Weedon et al. 2014) as a reference.

Table 7-1. Meteorological variables

Variable Name	Note	Directory	Unit
PSurf	Surface pressure	<code>met/dat/PSurf</code>	Pa
Rainf	Rainfall rate	<code>met/dat/Rainf</code>	kg m ⁻² s ⁻¹
Snowf	Snowfall rate	<code>met/dat/Snowf</code>	kg m ⁻² s ⁻¹
Wind	Wind speed	<code>met/dat/Wind</code>	m s ⁻¹
LWdown	Longwave downward radiation	<code>met/dat/LWdown</code>	W m ⁻²
Qair	Specific humidity	<code>met/dat/Qair</code>	kg kg ⁻¹
SWdown	Shortwave downward radiation	<code>met/dat/SWdown</code>	W m ⁻²
Tair	Air temperature	<code>met/dat/Tair</code>	K

7.2 Preparing Meteorological Data

In this section, we explain how to prepare WFDEI meteorological data for use with H08. There are two ways to prepare the data. One is to convert the NetCDF files distributed by the WFDEI website into H08 File Format (Option 1). The other way is to download H08-formatted binary files directly from A-PLAT Pro (<https://ccca-scenario.nies.go.jp>; Option 2). Both ways are shown below.

1. Change directory to `met/pre/`.
2. If you take Option 1, download WFDEI meteorological data files from the official WFDEI website¹¹, and save them in `met/org/WFDEI/daily`. Then edit and execute `prep_WFDEI.sh`.¹²

¹⁰ http://web.lmd.jussieu.fr/~polcher/ALMA/convention_3.html

¹¹ <https://doi.org/10.1002/2014WR015638>

¹² For half degree global simulation, use `prep_WFD.sh`.

If you take Option 2, visit A-PLAT Pro (<https://ccca-scenario.nies.go.jp>) and download WFDEI data files in Historical data, using options listed in Table 7-2. Then edit and execute `prep_wfdei_DataServer.sh`.

3. Check the settings and execute `prep_mean.sh`. This results in generating mean meteorological variables between YEARMIN and YEARMAX.
4. Check the settings and execute `prep_Prcp.sh`. This results in generating total precipitation (Prcp) by summing up rainfall (Rainf) and snowfall (Snowf).
5. Change directory to `met/pst`.
6. Edit and execute `calc_koppen.sh`. This results in outputting the Köppen-Geiger Climatic Zones into `met/out/Koppen__`.

This process results in the outputting of meteorological data to directory `met/dat`. Refer to Table 7-1 for subdirectories.

Table 7-2. Download options

Options	Settings
Dataset	WATCH-Forcing-Data-ERA-Interim/wfde (standard)
Temporal resolution	Daily
Variables	rlds, huss, prrn, prsn, tas, ps, rsds, sfcWind
Year	1979-1979
Spatial domain	Original
Select mask	Don't select
Option (mean calculation)	Don't calculate
File format	NetCDF
Compress the files	Yes (gzip compression)

• Column 3

● Checking Output Data 2: Analyzing Time Series Data

Following the procedure explained in Chapter 7 Section 2 should have resulted in the creation of temperature data in `met/dat/Tair_____`. Take a look at these data, which are daily. If you want to obtain Tokyo data (E139.75, N35.75), enter:

```
% cd met/dat/Tair_____
% punchhlf lonlat ./wfde_____hlfDY 1979 1979 139.75 35.75
```

This results in the display of daily time series data. Comparing WFDEI input data with observation data will help you to understand their features. Now convert these data into daily data by first converting the H08 File Format 2D daily data into monthly data, and then using the above procedure to display the data for just one location. The procedure is as follows:

```
% mon2yearhlf ./wfde_____hlfDY 1979 1979 ./wfde_____hlfMO
% punchhlf lonlat ./wfde_____hlfMO 1979 1979 139.75 35.75
```

When you use a large-scale model for research, you need to examine input and output data very closely. You can use all sorts of software to do so, but we recommend that you thoroughly familiarize yourself with H08 Analysis Tools by consulting Saito and Hanasaki (2012), since it is a dedicated H08 analysis tool designed to carry out the most important analyses with maximum efficiency. Once you are familiar with H08 Analysis Tools, you can adapt the source code as occasion demands to conduct more detailed analyses.

CHAPTER 8

Land Surface Process Module

This chapter explains the mechanisms and operation of the land surface process module, which is the most important of H08's modules.

8.1 Land Surface Process Module Mechanisms

H08's land surface process module is based on the standard bucket model described in Robock et al. (1995), but differs from it, mainly, in the following three points. First, the force-restore method (Bhumralkar, 1975) has been incorporated for surface temperature. Second, subsurface runoff has been incorporated. Third, groundwater recharge formulated by Döll and Fiedler (2008) has been incorporated. Those wishing to know detailed formulae and theoretical background should refer to Hanasaki et al. (2008a; 2018).

Figure 8-1 is a schematic representation of H08's land surface process module. For the sake of convenience, we will explain water balance (on the left of the diagram) and heat balance (right) separately. First, from the water balance perspective, the land surface process module is a single-layer bucket model. However, unlike the original bucket model (Manabe, 1969), subsurface runoff occurs continually according to soil moisture. This type of bucket model is also known as a leaky bucket model. Adopting the formulation by Döll and Fiedler (2009), a part of the total runoff turns into groundwater recharge. Recharge is stored in a one layer groundwater reservoir, which generates baseflow. Water flow from groundwater to surface water (capillary rise) is ignored in this model. Groundwater can be abstracted to a point exceeding the volume stored in the groundwater reservoir, which is called groundwater overexploitation. There are three state variables on water balance: soil moisture (SoilMoist), snow water equivalent (SWE), and groundwater (GW). Inputting precipitation (Prcp) causes evaporation (Evap), total runoff (Q_{tot}), and other fluxes to be calculated based on land surface water balance and heat balance, and soil moisture, snow water equivalent, and groundwater are updated as a result.

Next, from the heat balance perspective, the land surface process module serves as a model for understanding heat balance in the shallow surface layer. There are two state variables on heat balance: average surface temperature (AvgSurfT) and soil temperature (SoilTemp). Inputting six meteorological variables (see Table 7.1), namely, downward shortwave radiation (SWdown), downward longwave radiation (LWdown), air temperature (T_{air}), specific humidity (Q_{air}) or relative humidity (RH), wind speed (Wind), and surface pressure (PSurf) causes the calculation of the only surface temperature at which inputs and outputs at the surface are balanced, as well as the calculation of sensible heat flux (Q_h), latent heat flux (Q_{le}), ground heat flux (Q_g), upward shortwave radiation (SWup), upward longwave radiation (LWup), and other fluxes; surface temperature and soil temperature are updated as a result.

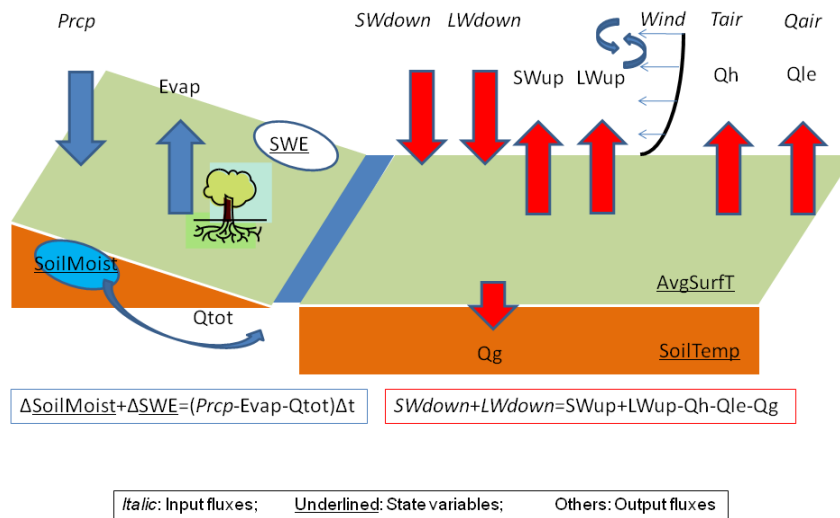


Figure 8-1. Schematic diagram of the land surface process module

8.2 Land Surface Process Module Calculation Process

The land surface process module (`lnd/bin/main.f`) calculation process is as follows:

1. Read in the file name of the namelist that defines calculation settings.
2. Read the namelist.
3. Read the map data.
4. Initialize the four state variables.
5. Start the time loop (spinup mode).
 - 5.1. In the calculation period, set just the calculation start year to repeat.¹³
 - 5.2. Read the eight meteorological data items.
 - 5.3. Call the `calc_leakyb` subroutine.
 - 5.4. Output the state variables.
 - 5.5. Output the fluxes.
 - 5.6. Save values of state variables
6. Stop the loop.
7. Ascertain whether you have satisfied the spinup termination conditions. Spinup is considered to be finished if the state variables in at least $\text{SPNRAT} \times \text{NL}$ of all cells (NL) change no more than the percentage of SPNERR compared with the previous spinup. If this condition is not satisfied, go back to No. 5.¹⁴
8. Start the time loop (calculation mode).

¹³ Which means that, if the calculation period is 1986–1995, you repeat just 1986.

¹⁴ Note that, depending on the combination of SPNRAT and SPNERR, spinup may continue endlessly.

- 8.1. Set the calculation period to from start year to end year.
- 8.2. Read the eight meteorological data items.
- 8.3. Call the `calc_leakyb` subroutine.
- 8.4. Output the state variables.
- 8.5. Output the fluxes.
- 8.6. Save the values of the state variables.
9. Stop the loop.
10. Output the locations and number of water balance anomalies, heat balance anomalies, and unresolvables.

8.3 Land Surface Process Module Execution 1: Preprocessing

Parameters and initial values for state variables are required to execute the land surface process module. The following explains how to create a file of spatially uniform provisional initial values and parameters.

0. Change directory to `lnd/pre/`.
1. Edit `prep.sh`. Set values for the parameters in Table 8-1 and the initial values for state variables in Table 8-2. For example, to set a soil moisture initial value file, set a specific number for the shell variable VAL related to SOILMOISTINI. Here, the same values have been applied globally, but you can edit values at your discretion.
2. Execute `prep.sh`. This results in the outputting of parameters in `lnd/dat`, and initial values for state variables in `lnd/ini`.
3. Edit and execute `prep_gamtau.sh`. This results in generating the files for the runoff shape parameter γ (`lnd/dat/gamma__`) and the time constant τ (`lnd/dat/tau__`), which reflect the Köppen-Geiger climatic zones (see Hanasaki et al. 2008a).
4. Edit and execute `prep_gwr.sh`. This results in generating the groundwater recharge parameters `fr`, `ft`, `fa`, `fg`, `fp`, and `rgmax` (`lnd/dat/gwr__`; see Hanasaki et al. 2018).

Table 8-1. List of model parameters

Variable	Note	Unit
SOILDEPTH	Soil depth	m
FIELD CAP	Field capacity	-
WILT	Wilting point	-
CG	Effective heat capacity of soil	J K ⁻¹ m ⁻²
CD	Bulk transfer coefficient	-
GAMMA	Parameter gamma	-
TAU	Parameter tau	Day
FR	Relief-related factor	-
FT	Soil texture-related factor	-
FA	Hydrogeology-related factor	-
FP	Permafrost/glacier-related factor	-
FG	Aggregated factor (FR*F*FA*FP)	-
RGMAX	Maximum groundwater recharge	kg m ⁻² s ⁻¹

Table 8-2. List of files on state variables¹⁵

File	Note	Unit
SOILMOIST	Soil moisture	kg m ⁻²
SOILTEMP	Soil temperature	K
SWE	Snow water equivalent	kg m ⁻²
AVGSURFT	Average surface temperature	K

8.4 Land Surface Process Module Execution 2: Calculation

After completing the instructions in the previous section, you should have all of the files needed to execute the land surface process module. Execute the land surface process module as follows:

0. Change directory to `ind/bin/`.
1. Edit `main.sh`.¹⁶ Set all shell variables shown in Tables 8-3 through 8-7. Be sure to set QTOT, POTEVAP, and EVAP to be outputted at a daily interval. QTOT will be used in Chapter 9 and POTEVAP and EVAP will be used in Chapter 10.

¹⁵ A variable name prefixed by DIR signifies a directory, and a variable name ending in INI signifies an initial value file. Variables with neither are output files.

¹⁶ When calculating the entire globe with a spatial resolution of 0.5, special attention should be paid to (1) changing PRJ from GSW2 to WFDE, (2) setting LDBG to a number less than L (259200 in this case), (3) changing the input data from GSWP2B1b_ to WFD_____, (4) changing the last two characters of BALBEDO from MO (monthly, year is variable) to MM (mean monthly, year is fixed at 0000).

2. Execute `main.sh`.¹⁷ This results in the background execution of the main process (`main.f`). A log file is created in `./log` and, during a simulation, a log is written to this file. As such, you can use such commands as:

```
% tail -f $LOGFILE
```

to check the simulation process.

Simulation results are outputted to `lnd/out`. 3rd level directories corresponding to module output elements and conforming to the ALMA conventions Version 3 are created directly below `lnd/out`.

¹⁷ H08 is designed to run with any spatial domain and spatial resolution but, at the present time, the main process is the one exception. If you look at the top of `main.f`, you should be able to see that a fixed number is given to the variable `n0l` that signifies array size. If you change the spatial domain or resolution and execute H08, you need to modify this part and recompile.

Table 8-3. List of variables on calculation condition

Variable	Note	Unit
PRJ	Project: Must be four characters. Used in output file name.	-
RUN	Run: Must be four characters. Used in output file name.	-
YEARMIN	Minimum year	-
YEARMAX	Maximum year	-
SECINT	Interval in seconds	sec
LDBG	Debugging point in the L coordinate. LDBG must be smaller than L (the size of array).	-
SPNFLG	Flag of spinup: 1 = the condition spinup has been completed, 0 = not completed. If this variable is set at 0, the model is executed in spinup mode, followed by normal calculation mode. If this is set at 1, the model is executed in normal calculation mode immediately after the state variables are initialized.	-
SPNERR	Spinup error tolerance: Spinup is considered to be finished if state variables in at least SPNRAT×NL of all cells (NL) change no more than the percentage of SPNERR compared to the previous spinup.	-
SPNRAT	Spinup ratio (See above)	-
ENGBALC	Energy balance error tolerance	W
WATBALC	Water balance error tolerance	mm day ⁻¹
CNTC	Maximum calculation iteration	-
PROG	Program	-

Table 8-4. List of files on input meteorological variables and albedo

File	Note	Unit
WIND	Wind speed	m s^{-1}
RAINF	Rainfall rate	$\text{kg m}^{-2}\text{s}^{-1}$
SNOWF	Snowfall rate	$\text{kg m}^{-2}\text{s}^{-1}$
TAIR	Air temperature	K
QAIR	Specific humidity. Set either QAIR or RH. Do not set both.	kg kg^{-1}
PSURF	Surface pressure	Pa
SWDOWN	Downward shortwave radiation	W m^{-2}
LWDOWN	Downward longwave radiation	W m^{-2}
RH	Relative humidity. Set either QAIR or RH. Do not set both.	

Table 8-5. List of files for simple climate change experiment

File	Note	Unit
TCOR	Air temperature difference correction	K
PCOR	Precipitation ratio correction	-
LCOR	Longwave ratio correction	W m^{-2}
TAIROUT	Corrected air temperature	K
RAINFOUT	Corrected rainfall rate	kg kg^{-1}
SNOWFOUT	Corrected snowfall rate	Pa
LWDOWNOUT	Corrected downward longwave radiation	W m^{-2}

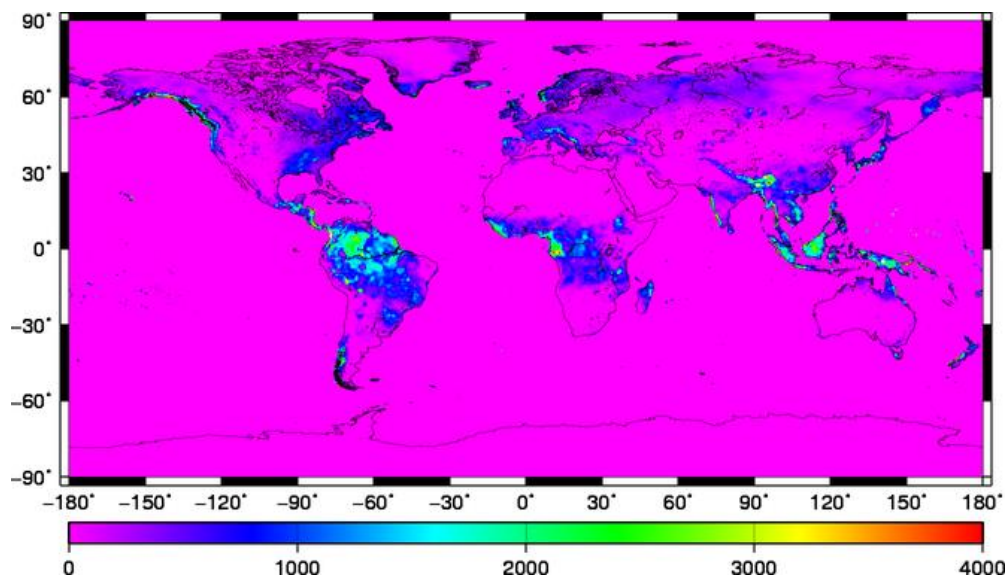
Table 8-6. List of files on input map variables

File	Note	Unit
LNDMSK	Land mask	-
BALBEDO	Base snow-free albedo	

Table 8-7. List of output files

File	Note	Unit
SWNET	Net shortwave radiation	$W m^{-2}$
LWNET	Net longwave radiation	$W m^{-2}$
QH	Sensible heat flux	$W m^{-2}$
QLE	Latent heat flux	$W m^{-2}$
QG	Ground heat flux	$W m^{-2}$
QF	Energy of fusion	$W m^{-2}$
QV	Energy of sublimation	$W m^{-2}$
EVAP	Evapotranspiration	$kg m^{-2} s^{-1}$
POTEVAP	Potential evapotranspiration	$kg m^{-2} s^{-1}$
QS	Surface runoff	$kg m^{-2} s^{-1}$
QSB	Subsurface runoff	$kg m^{-2} s^{-1}$
QTOT	Total runoff	$kg m^{-2} s^{-1}$
SOILMOIST	Soil moisture	$kg m^{-2}$
SOILTEMP	Soil temperature	K
SWE	Snow water equivalent	$kg m^{-2}$
AVGSURFT	Average surface temperature	K
SUBSNOW	Snow sublimation	$kg m^{-2} s^{-1}$
SALBEDO	Snow albedo	-

This calculation produces a great many results. As an example, Figure 8-2 shows annual runoff for 1979.

Figure 8-2. Mean annual runoff in 1979 [mm]¹⁸

¹⁸ Created using the following procedure:

8.5 Land Surface Process Module Execution 3: Postprocessing

Be sure to check water balance as follows after calculation has finished to confirm that calculations have been completed correctly.

0. Change directory to `cpl/pst/`.
1. Edit and execute `list_watbal.sh`.¹⁹

This results in the outputting of water balance to `cpl/tab/wat_bal_`. The BAL part is water balance. Check that this value is small enough. A value of 9999.99 indicates a missing value, showing that the relevant file does not exist. `list_watbal.sh` is a program for checking water balance in yearly units. Water balance calculation for multiple years is not supported.

The output of `list_watbal.sh` displays water balance results calculated using two methods. The first 100 to 200 lines show the global (or basin) totals of all terms appearing in the water balance equation, followed by the global (or basin) water balance errors calculated by substituting these totals into the water balance equation. On the other hand, the last 40 lines or so of the output (starting with “+++Water imbalance of land+++”) display information about the grid with the largest water balance error. Additionally, BALLND, BALSU, and BALGW show the global (or basin) water balance error calculated by solving the water balance equation for each grid and summing the results.

At a global resolution of 0.5 degrees (approximately 70,000 effective grid cells), no significant differences are observed between the two methods. However, at a global resolution of 5 minutes (approximately 2 million effective grid cells), the differences become significant, with the former method showing larger errors. When simulating at high resolutions, it is recommended to adopt the water balance error from the latter method.

3. Check the settings and execute `draw_all.sh`.

This results in the display of a time series graph showing water balance items for one location. This script can be useful for the detailed analysis of calculation results for cells in which calculation errors have occurred.

```
% cd lnd/out/Qtot
% gmt makecpt -T0/4000/1000 -Z > temp.cpt
% mulhlf GSW2LR_19860000.hlf 86400 temp.hlf
% mulhlf temp.hlf 365 temp.hlf
% hlf2eps temp.hlf temp.cpt temp.eps
% htconv temp.eps temp.png rot
```

¹⁹ If analyzing for the year 1979, set `YEAR=1979`, `MON=00`, `YEARINI=1978`, `MONINI=12`, `YEAREND=1979`, and `MONEND=12`. See Figure 5-5 for why it is set like this.

• Column 4

● Spinup

Initial values for soil moisture and other state variables are required for starting the land surface process module simulation. If the state variables are unknown at the start of the calculation, a method must be used to estimate them. We will take a look at two methods here.

One method makes use of the fact that it takes only about 2–3 years for the state variables in H08's land surface process module to reach a steady state. You can pick appropriate initial values to start the simulation confident in the knowledge that a completely steady state will be achieved in 5 years. You can make those 5 years the spinup period and exclude them from the data to be analyzed, starting analysis with the 6th year. The disadvantage of this method is that it shortens the period analyzed.

A second method involves starting the calculations with appropriate initial values and then repeatedly providing the model with the meteorological data for the first year of the simulation to bring it to a steady state. More specifically, the module is deemed to have reached a steady state once the difference between the previous year's state variables fall below a certain level, and then those values are set as the initial values and the simulation is restarted. With the standard settings of the land surface process module, the condition for deciding that soil moisture has reached a steady state is when the difference between the previous year's value is less than 5% in at least 95% of the cells.

The ideal method is to combine these two methods. The period used for analysis was provided by the data developer (Weedon et al. 2014). The spinup method adopted was to use the 1979 data to run repeated calculations and generate initial values, after which calculations were run up to 2013, and only the last 30 years were used for analysis.

CHAPTER 9

River Module

This chapter explains the river module.

9.1 River Module Mechanism

The H08 river module is based on the model and source code provided by Oki et al. (1999). Those wishing to know detailed formulae and theoretical background should refer to Oki et al. (1999).

Figure 9-1 is a schematic representation of the river module. Rivers are virtual elements depicted as straight lines and lacking any cross-sectional component. The single state variable is river storage (RivSto). Inflow (RivInf) from the upstream river and total runoff ($Q_{tot} \times A$) from the same cell flow into the river. River flow (RivOut) is calculated on the assumption that water flows at a constant speed over the distance from the calculated cell to the next downstream cell.

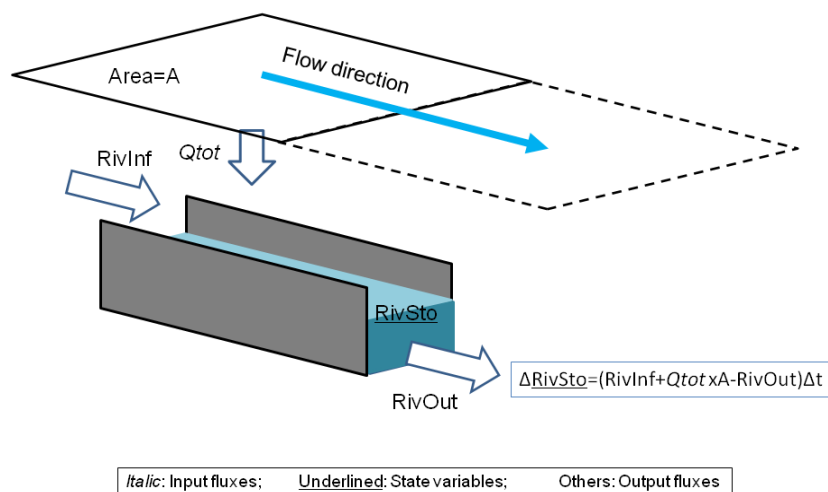


Figure 9-1. Schematic figure of the river module

9.2 River Module Calculation Process

The river module calculation process is as follows:

1. Read in the file name of the namelist that defines calculation settings.
2. Read the namelist.
3. Read the map data.
4. Initialize the state variable (only river storage).
5. Start the loop. (See Chapter 8 for spinup information.)
 - 5.1. Read the flux (only total runoff).
 - 5.2. Call the `calc_outflow` subroutine.

- 5.3. Write the state variable (only river storage).
- 5.4. Write the flux (only river flow).

9.3 River Module Execution 1: Preprocessing

0. Change directory to `riv/pre/`.
1. Check the settings and execute `prep.sh`. Set values for the parameters in Table 9-1 and the initial value for the state variable in Table 9-2.
2. This results in outputting of two parameter files, namely, for meander ratio and flow velocity in `riv/dat`, and an initial value file for river storage in `riv/ini`.

Table 9-1. List of model parameters

Variable	Note	Unit
FLWVEL	Flow velocity	m s ⁻¹
MEDRAT	Meandering ratio	-

Table 9-2. List of files on state variables

File	Note	Unit
<code>RIVSTO</code>	River storage	kg

9.4 River Module Execution 2: Calculation

0. Change directory to `riv/bin/`.
1. Edit and execute `main.sh`²⁰. Set the shell variables shown in Tables 9-3 through 9-5.
2. A data file of the downstream side flow for each cell's rivers is outputted to directory `riv/out/riv_out_`. The flow unit is [kg s⁻¹]. A river storage data file is outputted to directory `riv/out/riv_sto_`, with [kg] as the river storage unit.

Table 9-3. List of hydrological variables

File	Note	Unit
<code>QTOT</code>	Total runoff	kg m ⁻² s ⁻¹

²⁰ The method for performing calculations under the FTCS scheme is as follows;

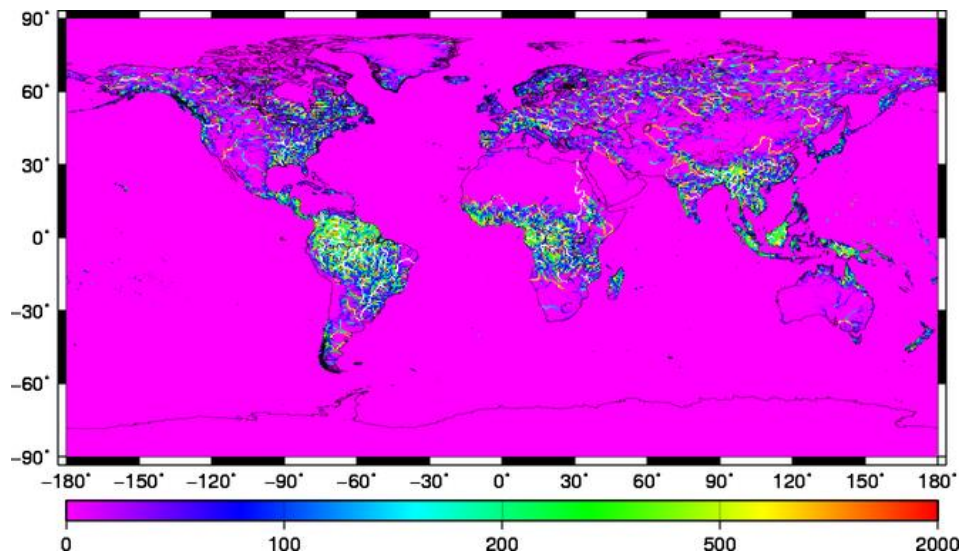
1. Confirm that there are four files, `main_ftcs.f`, `main_ftcs.sh`, `calc_outflow_ftcs.f` and `calc_humact_ftcs.f` in `riv/bin` and that they have all been compiled.
2. Open `main_ftcs.sh` and edit the basic settings. At this time, make sure that SECINT (seconds interval) is set to a sufficiently short value.
(`main_ftcs.f` contains the process to refine the time resolution of the river model until it meets the CFL condition.)
3. Execute `main_ftcs.sh`.
4. Proceed to 9.5 and check the results in the same way as when `main.sh` was executed.

Table 9-4. List of input map data

File	Note	Unit
RIVSEQ	River sequence: The origin of the river channel is 1, and 1 is added for every grid downstream. 0 is assigned to sea grids.	-
RIVNXL	The L coordinate of the next (downstream) grid cell	-
RIVNXD	Distance to the next (downstream) grid cell	m
LNDARA	Area of land cell	m ²

Table 9-5. List of output files

File	Note	Unit
RIVOUT	River discharge	kg s ⁻¹

Figure 9-2. Annual river discharge in 1979 [m³ s⁻¹]

9.5 River Module Execution 3: Postprocessing

0. Change directory to `cp1/pst/`.
1. Edit and execute `list_watbal.sh`.

This results in the creation of a table of water balance according to river basin. Check that the water balance is closed (i.e., the inputs and outputs are balanced).

2. Return back to Section 6.6 and execute `main_dam.sh`.

CHAPTER 10

Crop Growth Module

This chapter explains the crop growth module.

10.1 Crop Growth Module Mechanism

We adapted the crop growth algorithm appearing in SWIM (Krysanova et al., 2000) to create the H08 crop growth module. SWIM is a river basin hydrology model that contains one model for simulating hydrological cycles and another model for simulating crop growth. In general, crop growth models are used to estimate or predict crop yields, but since H08 is designed mainly to simulate daily irrigation water demand, it is used to estimate global cropping calendars. This is a fairly specialized usage that requires care.

The most important concept in the crop growth module is accumulated temperature, which is the sum of daily average temperatures (T_{air}) from the planting date. If, for example, the crop was planted on April 1, and the average temperature was 15°C for that day, 18°C for April 2, and 12°C for April 3, accumulated temperature would be 15°C for April 1, 33°C for April 2, and 45°C for April 3. Once this accumulated temperature reaches a certain mark (e.g., 1,500°C), the crop is mature and ready for harvest. Each crop is given its own accumulated temperature up to maturity. The period from planting date to harvesting date is known as the cropping period. We will now look at a specific example. The red line in Figure 10-1 is the daily average temperature for an imaginary location. The green line shows the length of the cropping period in days (i.e., days needed until the accumulated temperature reaches 1,500°C) for a crop planted each day from January 1 to December 31. The cropping period becomes shorter as the planting date approaches the summer period, and longer as it approaches the winter.

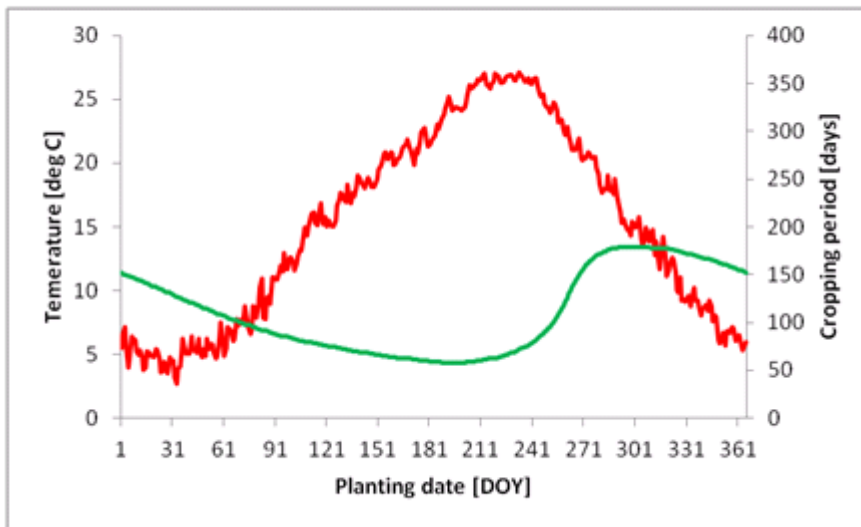


Figure 10-1. Daily average temperature [°C] (red) and cropping period [days] (green) for an imaginary location

In general, plants do not grow if the temperature is too low, so taking this into consideration, a base temperature is given to the crop concerned, and this is often first subtracted from each daily average temperature before summation. The accumulated temperature in this case is known as the effective accumulated temperature. SWIM also uses effective accumulated temperature. The green line in Figure 10-2 shows a cropping period with a base temperature of 10°C. Here it is assumed that the crop fails if a day in the cropping period falls below 10°C.

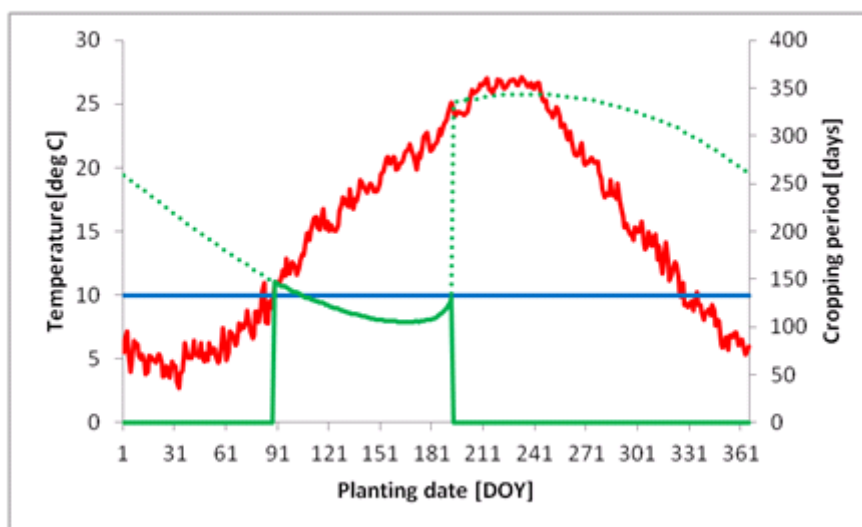


Figure 10-2. Daily average temperature [°C] (red) and cropping period [days] (green) with effective accumulative temperature

Next, we will explain the basics of calculating yield. Total biomass (BT) for the cropping period is expressed as follows. The symbols used in the equations below follow the SWIM manual (Krysanova et al., 2000).

$$BT = \sum \Delta B \quad (10-1)$$

where ΔB represents daily biomass production, and is expressed as follows:

$$\Delta B = BE \times PAR \times REGF \quad (10-2)$$

where BE is photosynthetic efficiency, a parameter that is unique to each plant species. For example C4 plants have a higher photosynthetic efficiency than C3 plants. PAR is photosynthetically active radiation, and is calculated by converting downward shortwave radiation (SWdown). $REGF$ stands for regulation factors. The SWIM model has four regulation factors: temperature (high and low), water, nitrogen, and phosphorus. If none of these impede growth, $REGF$ is 1, and photosynthesis occurs according to PAR . If there is some kind of impediment, $REGF$ falls below 1, and biomass production will also be impeded. In general, the longer the cropping period is, the higher PAR becomes, and biomass production with it.

Yield (YLD) is expressed as follows:

$$YLD = BAG \times HVSTI \times \frac{WSF}{WSF + \exp(6.117 - 0.086WSF)} \quad (10-3)$$

Here, BAG is aboveground biomass, $HVSTI$ is a parameter known as harvest index and represents the highest percentage of BAG that the harvested product accounts for. The 3rd term shows the water stress on the crop in the latter half of the cropping period.²¹ WSF is the water stress factor, and is expressed as follows:

$$WSF = \frac{SWU}{SWP} \times 100 \quad (10-4)$$

Here, SWU is actual evapotranspiration in the latter half of the cropping period²², and SWP is potential evapotranspiration in the same period. Fruits grow in the latter half of the cropping period, but if there is water stress during this period, yield will be reduced in line with the WSF value.

We will now explain the specific calculation method that H08 uses for estimating cropping calendars (planting date and harvesting date). First, the crop growth model calculates harvesting date and yield for a crop planted on every day from January 1 to December 31. If the temperature during the period drops below the temperature threshold for death from cold, crops will perish and yield will be zero. The planting date that results in the greatest yield over the year is made the planting date for that crop at that location, and the yield for that date is the potential yield. Figure 10-3 shows an example of the

²¹ Crop growth is divided into vegetative growth, represented by the growth of roots, leaves, stems and other vegetative components, and reproductive growth, which is the growth of flowers and fruits. The latter half of the cropping period is when reproductive growth occurs. In SWIM, the latter half of the cropping period starts when the effective accumulated temperature exceeds half of the accumulated temperature when maturity is reached. You should note, however, that in actual plants the switch from vegetative to reproductive growth is not decided just by accumulated temperature.

²² SWIM differentiates between evaporation and transpiration, and this is expressed as actual transpiration in the SWIM manual.

relationship between cropping date and crop yield. In this case, the crop can be harvested if the planting date ranges between 80 and 190 [DOY], and that of 95 [DOY] produces the maximum yield. A global cropping calendar is created for the 19 crops (see Table 10-1) for which global harvest area distribution is given in Leff et al. (2004).

If calculated as above, yield will be dictated by the weather each day, and it does not change smoothly. Moreover, temperature is low on many days in early spring, and if it drops too low, plants may die from cold. To exclude such possibilities, meteorological data that provide the average over several years are used for calculations rather than data for a specific year. (Normally GSWP2-B1 1986–1995 10-year averages are used.) Also, taking the moving average of the yields for the planting date plus yields for the 10 days before and 10 days after (for a total of 21 days) when calculating the cropping period smooths the fluctuations over a year. We use this to decide the planting date and harvesting date that will provide the greatest yield, and make this the cropping period.

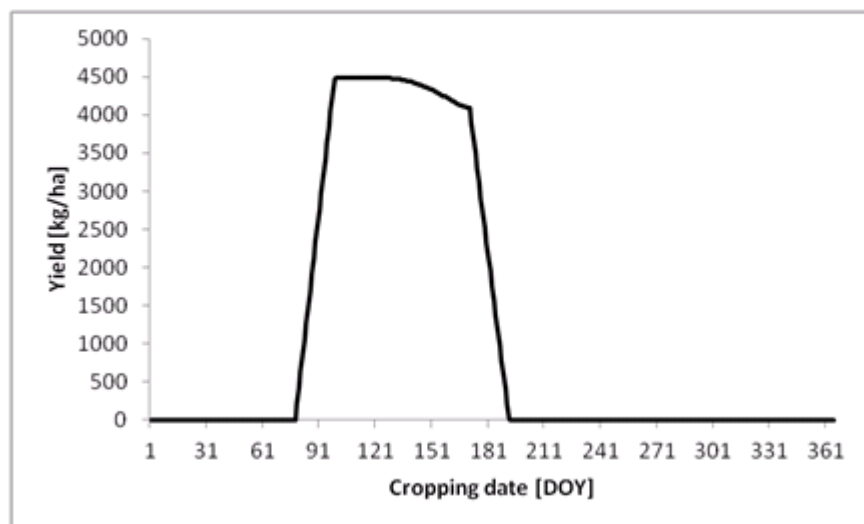


Figure 10-3. Cropping date and yield

In warm regions, double cropping or two different crops are grown in a year. (We will refer to the second crops in both systems hereafter as “second crops” to differentiate between the first and second crops.) Cropping calendars for second crops can be created in the same way as for first crops by using every day from January 1 to December 31 as a planting date, and calculating resulting yield for each day. However second crops must necessarily be planted and harvested outside the cropping period of first crops. Moreover, a space of at least 15 days is required between harvesting the first crop and planting the second. If this condition is not met, yield will be given as zero on the assumption that the calendars overlap.

We created cropping calendars for first crops of 19 crop types. Theoretically, counting second crops too, $19 \times 19 = 361$ combinations are possible, but actually running so many simulations would require enormous computing resources, making such a proposition unfeasible. As such, under the standard

configuration, we assume that the crop type in each cell with the 2nd highest planting area in the harvest area percentage data of Leff et al. (2004) is the second crop, and estimate the cropping period for these crops. You should note, however, that this is a fairly crude assumption.

Table 10-1. List of crop IDs. Abbreviations ending with “g” indicate that crop parameters are NOT available, and are substituted by using a generic crop parameter.

Crop ID	Abbreviation	Note
1	bar_	Barley
2	casg	Cassava (generic crop parameter)
3	cot_	Cotton
4	grn_	Ground nut (peanut)
5	mai_	Maize (corn for grain)
6	milg	Millet (generic crop parameter)
7	oilg	Oil palm (generic crop parameter)
8	othg	Others (generic crop parameter)
9	pot_	Potatoes
10	pulg	Pulses (generic crop parameter)
11	rap_	Rape
12	ric_	Rice
13	rye_	Rye
14	sor_	Sorghum
15	soy_	Soybean
16	sub_	Sugar beet
17	suc_	Sugarcane
18	sun_	Sunflower
19	whe_	Wheat

10.2 Crop Growth Module Calculation Process

The crop growth module calculation process is as follows:

1. Read in the file name for the namelist defining the calculation settings.
2. Read the namelist.
3. Read the parameter file.
4. Read the map data.
5. Create conversion tables for the L coordinates of all cells and just land cells.
6. Initialize the state variables.
7. Read the input hydrological and meteorological data (temperature, potential evapotranspiration, actual evapotranspiration, downward shortwave radiation) for the whole period.

8. Start the planting date loop.
 - 8.1. Set the cropping flag to 1 when the temperature exceeds the planting date temperature.
 - 8.2. Start the cropping days loop.
 - 8.2.1. If cropping days overlap, set the cropping flag to 0.
 - 8.2.2. Set the input hydrological and meteorological data.
 - 8.2.3. Call the `calc_crpyld` subroutine.
 - 8.2.4. Copy outputted results to an array.
 - 8.3. Initialize the array.
9. Copy outputted results (yield) to an array.
10. Calculate the moving average for yield, and output the planting dates, harvesting dates, and cropping days that provide maximum yield.
11. Output results to a file.

2.3 Crop Growth Module Execution 1: Preprocessing

3. Move `SWIM` from `map/org/` to `crp/org/`.


```
% cd $DIRH08
% mkdir crp/org
% mv map/org/SWIM crp/org/
```
4. Change directory to `crp/pre/`.
5. Check the settings and execute `prep.sh`. This results in calculation of the mean SWdown, Tair, PotEvap, and Evap.
6. Return back to Section 6.7 and execute `prep_map_despot.sh`.

Here, note that the crop growth module also uses the evapotranspiration and potential evapotranspiration quantities that are outputted results of the land surface process module.

Table 10-2. List of files on state variables

File	Note	Unit
BT	Total biomass	kg ha ⁻¹
RSD	Residual (un-decomposed biomass in soil)	kg ha ⁻¹
OUTB	Biomass	kg ha ⁻¹
HUNA	Effective accumulated temperature	degC
SWU	Evaporation in latter half of cropping period	mm
SWP	Potential evaporation in latter half of cropping period	mm
REGFW	Days with water stress	day
REGFL	Days with low temperature stress	day
REGFH	Days with high temperature stress	day
REGFN	Days with nitrogen stress	day
REGFP	Days with phosphorus stress	day

10.4 Crop Growth Module Execution 2: Calculation

0. Change directory to `crp/bin/`.
1. Edit and execute `main.sh`. Set the shell variables of Tables 10-3 through 10-8. Set `JOBS="1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19"`. These numbers represent crop types. Doing this enables the calculation of cropping calendars for all 19 crop types.

This results in the outputting of calculation results for each crop to `crp/out`. The directory beginning with `crp_` is cropping days, that beginning with `hvs_` is harvesting date, that beginning with `yld_` is yield [kg ha⁻¹], that beginning with `reg_` is the ID of the dominant regulating factor (1: high temperature stress; 2: low temperature stress; 3: water stress; 4: nitrogen stress; 5: phosphorus stress), that beginning with `plt_` is planting date, and that beginning with `mav_` is the 21-day moving average yield [kg ha⁻¹]. The yield [kg ha⁻¹] for barley planted on January 1, for example, can be found in the `../out/mav_bar_/WFDE__C_00000101.hlf` file.

2. Change directory to `crp/pst/`.
3. Check the settings and execute `calc_crpcal.sh`. The MARGIN variable shows the minimum number of days required between the first and second crops.

This results in the outputting of the start and end dates of the occupation periods of first crops to `crp/out/ocu_ini_` and `crp/out/ocu_end_` respectively. The start date is the date MARGIN days before planting date of the first crop, and end date is the date MARGIN days after harvesting date of

the first crop. The first crop planting date, harvesting date, cropping days, yield, and dominant regulating factor are also outputted respectively to `crp/out/plt_1st_`, `crp/out/hvs_1st_`, `crp/out/crp_1st_`, `crp/out/yld_1st_`, and `crp/out/reg_1st_`. These are for the crops that count as first crops in the calculation results for the 19 crops.

4. Change directory to `crp/bin/`.
5. Edit and execute `main.sh`. Set `JOBS=2nd`. Note that `ocu_ini_` and `ocu_end_` are set inside the `JOBS` loop.

This results in creating the second crop planting date, harvesting date, cropping days, yield, and total biomass being outputted respectively to `crp/out/plt_2nd_`, `crp/out/hvs_2nd_`, `crp/out/crp_2nd_`, `crp/out/yld_2nd_`, and `crp/out/bt___2nd_`.

10.5 Crop Growth Module Execution 3: Postprocessing

0. Change directory to `crp/pst/`.
1. Check the settings and execute `draw_crp_yld_map.sh`.
2. Cropping calendars and maps showing the global distribution of yield will be outputted to the `crp/fig/` directory.

Table 10-3. List of variables on simulation settings

Variable	Note	Unit
PRJ	Project name (four characters)	-
RUN	Run name (four characters)	-
SUF	Suffix of H08 File Format 2D (four characters)	-
MAP	Type of map (e.g., <code>.WFDEI</code> or <code>flwdir.WFDEI.hlf</code>)	-
YEAR	Year to simulate: 0000 for the average of the period	-
SECINT	Calculation interval	sec
NL	Total number of cells (e.g., 259,200 [720×360] for 0.5°×0.5° global domain)	-
LDBG	L coordinate to output in debugging mode	-
RAMDBG	Crop ID to output in debugging mode	-
JOBS	Crop ID: from 1 to 19 (integer) for first crop; "2nd" for second crop calculation	-

Table 10-4. List of files on model parameters (standard parameters of SWIM)

File	Note	Unit
RAM2SWIM	Conversion tables for the crop IDs of Leff et al. (2004) and SWIM	-
RAM2NAME	Conversion tables for the crop IDs of Leff et al. (2004) and abbreviations shown in Table 10-1.	-
SWIM2RAM	Conversion tables for the crop IDs of SWIM and Leff et al. (2004)	-
CRPPAR	Parameter file of SWIM	-

Table 10-5. List of model parameters (original parameters of H08)²³

Variables	Note	Unit
DAYMAV	Days to calculate the moving average of the yields for a planting date: 10 indicates the average yields for the 10 days before and 10 days after the planting date (for a total of 21 days)	day
INTCRPDAYMAX	Maximum cropping days	day
REGFMIN	REGF is the primary regulating factor among high temperature, low temperature, water, nitrogen, and phosphorus stresses; REGFMIN sets the threshold of REGF (e.g., when you set REGFMIN=0.7, REGF is outputted when one of stresses falls below 0.7)	-
TDORM	Temperature threshold of dormancy	degC
TFRZ	Temperature threshold of death from cold	degC
HUNMAX	Maximum daily accumulation of temperature	degC
IHUNMAT	<u>No longer used. Set to "1.0".</u>	-
TSAW	Minimum temperature for planting	degC
THVS	Minimum temperature for harvesting	degC
OPTTS	Option of temperature stress ("yes" to enable the process)	yes/no
OPTWS	Option of water stress ("yes" to enable the process)	yes/no
OPTNS	Option of nitrogen stress ("yes" to enable the process)	yes/no
OPTPS	Option of phosphorus stress ("yes" to enable the process)	yes/no
OPTFRZ	Option of death due to cold ("yes" to enable the process)	yes/no

²³ Because parameters in this table are not found in the original SWIM, they should be examined especially carefully.

Table 10-6. List of files on input meteorological variables

File	Note	Unit
TAIR	Air temperature	K
SWDOWN	Shortwave downward flux	Wm ⁻²
POTEVAP	Potential evapotranspiration (output of land surface process module)	kgm ⁻² s ⁻¹
EVAP	Evapotranspiration (output of land surface process module)	kgm ⁻² s ⁻¹

Table 10-7. List of files on maps

File	Note	Unit
LNDMSK	Land mask	-
CRPTYP	Crop type	-
DOYOCUINI	Initial date of the period 1 st crop occupies cropland	DOY
DOYOCUEND	Final date of the period 1 st crop occupies cropland	DOY

Table 10-8. List of output files

File	Note	Unit
YLDMAV	Yield (moving average)	kg ha ⁻¹
YLDMAX	Maximum yield	kg ha ⁻¹
PLTDOYMAX	Planting date producing the maximum yield	DOY
HVSDOYMAX	Harvesting date producing the maximum yield	DOY
FILECRPDAYMAX	Cropping days producing the maximum yield	day
REGFD	Dominant regulating factor	-
CWSG	Crop water supply from green water ²⁴	kg m ⁻²
CWSB	Crop water supply from blue water ²⁵	kg m ⁻²

²⁴ Water consumed during cropping period in rainfed cropland.

²⁵ Irrigation water consumed during cropping period in irrigated cropland.

CHAPTER 11

Reservoir Operation Module

This chapter explains the reservoir operation module. Reservoir level, storage, release, and other variables are controlled according to operation rules. In Japan, for example, reservoir operations, including periods of operation, are governed by very strict rules, and reservoirs are operated according to those rules. If it were possible to obtain the operation rules for all of the world's reservoirs, H08 would be able to use those rules to simulate reservoir operations but, in general, these rules are not disclosed, and obtaining them from developing countries is particularly difficult. Moreover, if the storage capacity of a reservoir is larger than the annual inflow, storage will be carried over to the following year and, in such cases, operation rules are often decided each year according to circumstances. For the above reasons, operation rules need to be estimated for each of the world's reservoirs, and it is the reservoir operation module that does this.

11.1 Reservoir Operation Module Mechanism

The reservoir operation module mechanism and basis is explained in detail in Hanasaki et al. (2006), and so the following is only an overview.

The reservoir operation module is a model for estimating reservoir operation rules. Each reservoir has its own actual rules, and estimating such rules is not easy. However, because the amount of accessible information on reservoirs is very limited at present, the reservoir operation module estimates only the simplest conceptual operation rules. These rules are based on two assumptions. The first is that reservoirs are operated in such a way as to reduce the size and frequency of yearly and seasonal fluctuations in flow. The second is that if there are seasonal fluctuations in downstream water demand, reservoirs will be operated in such a way as to address those fluctuations.

The reservoir operation module handles only reservoirs with a total storage capacity of at least 1 km³, labeling those whose main purpose is irrigation as "irrigation reservoirs", and others as "non-irrigation reservoirs".

The reservoir operation rules estimated by H08 can be divided roughly into those designed to ease yearly fluctuations and those designed to ease seasonal fluctuations (by releasing water in line with seasonal fluctuations in downstream demand). The former rules take the following single form. Annual total release $R[\text{m}^3]$, when river flow is $I_{mean}[\text{m}^3]$, is expressed as

$$R \approx k_{rls} \cdot I_{mean} . \quad (11-1)$$

Here, k_{rls} is a release coefficient, where

$$k_{rls} = \frac{S_{first}}{0.85C} . \quad (11-2)$$

S_{first} is storage at the starting point of the reservoir operation year, C is total storage capacity of the reservoir, and 0.85 is an empirically derived coefficient.

Reservoir operation year can be explained as follows. First, the seasonal fluctuation in inflow to the reservoir in an average year is used to divide the year into periods when inflow exceeds the annual average, and periods when it is below the annual average. By using mathematical processing to eliminate fluctuations, the year can be divided into a water storage period when inflow exceeds the annual average, and a water release period when inflow falls below the annual average. The reservoir operation year starts at the start of the water release period, and ends at the end of the water storage period. Accordingly, S_{first} is the end point of the water storage period, or in other words, storage at the time of year that it is at its maximum.

Next, we will look at the operation to ease seasonal fluctuations (by discharging water in line with seasonal fluctuations in downstream demand). As mentioned above, reservoirs are divided, according to purpose, into irrigation and non-irrigation reservoirs. The provisional value for non-irrigation reservoir release (r' [m³/s]) is expressed as follows:

$$r' = k_{rls} \cdot i_{mean} \quad (11-3)$$

Here, i_{mean} is the average annual flow [m³/s]. The provisional value for irrigation reservoir release (r' [m³/s]) is expressed as follows:

$$r' = \begin{cases} \frac{i_{mean}}{2} + \frac{i_{mean}}{2} \times \frac{\sum \{k_{alc} \times (d_{agr} + d_{ind} + d_{dom})\}}{d_{mean}} & \left(\frac{i_{mean}}{2} \leq d_{mean} \right) \\ i_{mean} + \sum_{area} \{k_{alc} \times (d_{agr} + d_{ind} + d_{dom})\} - d_{mean} & \left(d_{mean} < \frac{i_{mean}}{2} \right) \end{cases} \quad (11-4)$$

Here, d_{agr} , d_{ind} , and d_{dom} are, respectively, agricultural, industrial and domestic household water demand; d_{mean} is long-term mean total water demand and k_{alc} shows dependence of water demand on reservoirs.

Take the example of two tributaries upstream from a certain point, each with a reservoir; k_{alc} shows which reservoir water demand depends on and to what extent or, put another way, which reservoir water demand is allocated to, and the extent of that allocation. Σ area is the sum of the calculated cells downstream from the reservoirs.

Finally, release (r [m³/s]) is expressed as follows:

$$r = \begin{cases} k_{rls} \times r' & (c \geq 0.5) \\ \left(\frac{c}{0.5} \right)^2 k_{rls} \times r' + \left(1 - \frac{c}{0.5} \right)^2 i & (c < 0.5) \end{cases} \quad (11-5)$$

Here, i is inflow [m³/s], and $c (=C/I_{mean})$ is the comparison of storage capacity with annual flow. If c is smaller than 0.5, release will depend on inflow. If storage capacity is conspicuously smaller than annual flow (that is, $c \approx 0$), release cannot be changed through reservoir operation, and release is accordingly the same as inflow.

11.2 Reservoir Operation Module Calculation Process

The reservoir operation module calculation process is as follows:

1. Read in the file name for the namelist defining calculation settings.

2. Read the namelist.
3. Read the input data.
4. Initialize the state variables.
5. Start the time loop.
 - 5.1. Set the inflow data.
 - 5.2. Set the water demand data.
 - 5.3. Call the `calc_resope` subroutine.
 - 5.4. Copy the outputted results to an array.
6. Output the results to a file.

11.3 Reservoir Operation Module Execution 1: Preprocessing

0. Change directory to `riv/pst/`.
1. Check the settings and execute `calc_mean.sh`.
2. Check the settings and execute `calc_flddro.sh`.

This results in dividing a year into storage and release periods using the river flow simulation. Based on this result, `riv/out/fld_dro_` is created.

You MUST skip the following procedures and proceed Chapter 12 unless you wish to develop or modify the reservoir operation module.

The following part introduces how to run the reservoir operation module in a stand-alone mode. This is a special simulation only for intensive model development and validation²⁶. The module is usually used together with the land surface hydrology and river module as shown in Chapter 13.

3. Check that there is `H06-GRanD/` in `map/org`. Move it to `dam/org/`.
4. Change directory to `dam/pre/`.
5. Execute `prep_obsope.sh`. This results in the conversion of reservoir operation records contained in `dam/org/H06_GRanD` into H08 File Format.
6. Execute `list_obsope.sh`. This results in the creation of the dam information file `dam/dat/obs_1st_/obsdat.txt` that is referenced by `dam/bin/main.sh`.
7. Execute `prep_damdem.sh`. This results in the recording of reservoir demand in `dam_dem_`. This operation should be conducted only after calculation of agricultural water demand.
8. If you execute `draw_obsope.sh`, you can plot the observed water storage and discharge rate.

²⁶ To be specific, if you wish to corroborate Hanasaki et al. (2006) by repeating a similar simulation.

11.4 Reservoir Operation Module Execution 2: Calculation

Here we explain a calculation method that uses solely the reservoir operation module. We explain in Chapter 13 how to run a river flow simulation linked with the land surface hydrology river module.

0. Change directory to `dam/bin/`.
1. Check the settings and execute `main.sh`.

11.5 Reservoir Operation Module Execution 3: Postprocessing

Because the results are outputted as a text file, they can be easily analyzed using MS Excel, etc. The following is a method for creating and checking a basic map.

0. Change directory to `dam/pst/`.
1. Edit and execute `draw_result.sh`.

CHAPTER 12

Environmental Water Module

This chapter explains the environmental water module.

12.1 Environmental Water Module Mechanism

Environmental water is the flow required to maintain the natural environment of rivers. H08 uses the Shirakawa (2005) model²⁷ to estimate global environmental water. This model uses gridded data for global river runoff height (a quantity derived by dividing river flow by catchment area) to extract the months (out of 12 months) with the highest and lowest monthly runoff height. It then divides land areas into the four categories listed in Table 12-1 based on monthly runoff height maximums (q_{\max}) and minimums (q_{\min}). This sets the environmental flow (q_{env}) for each month according to monthly river runoff height (q).

Table 12-1. Climatic category

Climatic category	Monthly river discharge	Condition	Environmental flow requirement
Dry	$q_{\min} < 1$ [mm month ⁻¹] and $q_{\max} < 10$ [mm month ⁻¹]	$0 \leq q < 1$	$q_{\text{env}} = 0$
		$1 \leq q$	$q_{\text{env}} = 0.1q$
Wet	10 [mm month ⁻¹] $\leq q_{\min}$ and 100 [mm month ⁻¹] $\leq q_{\max}$		$q_{\text{env}} = 0.4q$
Stable	1 [mm month ⁻¹] $\leq q_{\min}$ and $q_{\max} < 100$ [mm month ⁻¹]		$q_{\text{env}} = 0.1q$
Variable	Other than above	$0 \leq q < 1$	$q_{\text{env}} = 0$
		$1 \leq q < 10$	$q_{\text{env}} = 0.1q$
		$10 \leq q$	$q_{\text{env}} = 0.4q$

12.2 Environmental Water Module Calculation Process

The environmental water module (`riv/pst/prog_envout.f`) calculation process is as follows:

0. Read in the arguments.
1. Read the catchment area file.
2. Read the monthly river flow file [kg s⁻¹], and use the catchment area data to convert the unit to [mm month⁻¹].
3. Calculate monthly river flow maximums and minimums.
4. Divide global land areas into the four categories according to Shirakawa (2005).

²⁷ The model has been partially simplified. For example, the flooding disturbance in the Shirakawa (2005) model was excluded because it proved difficult to handle in the global model.

5. Calculate environmental water [kg s^{-1}] according to category.

12.3 Environmental Water Module Execution Method 1: Preprocessing

A prerequisite for this process is that the river module river flow simulation is run in monthly units.

12.4 Environmental Water Module Execution Method 2: Calculation

Environmental water is calculated as follows:

0. Change directory to `riv/pst/`.
1. Check the settings and execute `calc_envout.sh`.
2. Environmental flow is outputted to `riv/out/env_out_`.

12.5 Environmental Water Module Execution Method 3: Postprocessing

0. Change directory to `cp1/pst/`.
1. Edit and execute `list_watbal.sh`.
2. Environmental water totals will be displayed in the `cp1/tab/wat_bal_` tables.

CHAPTER 13

Coupled Model

We have so far explained how to use the land surface process module, river module, crop growth module, and reservoir operation module separately, and will now explain how the modules can be coupled and used as one.

13.1 Coupled Model Mechanism

Looking at the main files of the land surface process, river, crop growth, and reservoir operation modules, you can see that they are composed to conduct the following process: (1) Read in input data and settings; (2) call subroutines related to calculation; and (3) write out output data. In the coupled model, subroutines of the four modules are called from the single main file to enable the modules to be manipulated as a single model.

Apart from coupling the modules, the coupled model has a number of other features. The first is that a number of human activity elements, such as irrigation of cropland and withdrawal of water from rivers, have been added. The second is that single land area cells can be divided into subcells of a chosen number and size. Using this feature enables the handling of differing land uses or vegetation types that exist in one cell.

First, we will explain irrigation. In the bucket model used by the land surface process module we assume that evaporation efficiency β [that is, the ratio of actual evapotranspiration (E) to potential evapotranspiration (Epot)] and the soil moisture index²⁸ θ are related as follows:

$$\begin{cases} \beta = \theta/0.75 & \theta < 0.75 \\ \beta = 1 & 0.75 \leq \theta \end{cases} \quad (13-1)$$

If the soil moisture index is held at 0.75, actual and potential evapotranspiration become equal, and crops avoid water stress. As such, in H08, we define irrigation as the supply of water other than precipitation to maintain the soil moisture index at 0.75 during the cropping period. Similarly, irrigation water demand is the amount of water required to maintain the soil moisture index of irrigated cropland at 0.75 during the cropping period. Irrigation water demand defined in this way is known as net irrigation water demand, and previous research has calculated that this demand stands globally at 1,000 to 1,500 km³ yr⁻¹. Since global irrigated croplands cover a total area of about 2.50×10^6 km², irrigation water demand per unit area is, on average, 400 to 600 mm yr⁻¹.

We will now explain division into subcells. H08 allows a single calculation cell to be divided into a chosen number of subcells. These subcells can have their own state or flux variables. However, each subcell will be given the same input meteorological data; dividing cells into subcells like this is also known as "mosaicing". At least two subcells are required to conduct the above irrigation calculation: an irrigated cropland subcell that shows the percentage of the whole cell area occupied by irrigated land, and another subcell representing other land use. In this situation, the irrigated cropland subcells will have

²⁸ This is the value for normalized soil moisture when wilting point is given the value 0 and cropland water capacity is given the value 1.

higher soil moisture than non-irrigated subcells, and evaporation will also rise. In the standard H08, a grid cell is divided into four: irrigated cropland on which second crops are grown, irrigated cropland used only for a single crop each year, rainfed cropland used only for a single crop, and other land use. Calculation is carried out separately for these four types of subcells.

13.2 Coupled Model Calculation Process

The coupled model calculation process is as follows:

1. Read in the file names of each namelist defining calculation settings.
2. Read each namelist (four: land, river, crop growth, human activity).
3. Read the map parameter files (four: land, river, crop growth, human activity).
4. Initialize the state variables (land, river, and human activity).
5. Start the time loop.
 - 5.1. Give the variables for spinup.
 - 5.2. Read input meteorological data.
 - 5.3. Read environmental water data.
 - 5.4. Set cropping and irrigation flags (call the `calc_flgcrp` subroutine).
 - 5.5. Irrigate (call the `calc_irgapp` subroutine). [subcell calculation occurs]
 - 5.6. Calculate reservoir water demand (call the `calc_damdem` subroutine).
 - 5.7. Calculate reservoir release (call the `calc_resope` subroutine).
 - 5.8. Calculate land surface process (call the `calc_leakyb` subroutine), and output results to a file. [subcell calculation occurs]
 - 5.9. When calculating global warming (see Chapter 14), output meteorological data to a file.
 - 5.10. Convert downward shortwave radiation into daily data.
 - 5.11. Calculate river process including human activity (call the `calc_humact` subroutine).
 - 5.12. Output river, reservoir operation, medium-sized reservoir, and water withdrawal data to files.
 - 5.13. Calculate crop growth (call the `calc_crpyld` subroutine).
 - 5.13.1. Divide subcells into first crop and second crop subcells according to land use, and set cropping/irrigation flags and crop types.
 - 5.13.2. Provide each subcell with meteorological and state variable data.
 - 5.13.3. Call the `calc_crpyld` subroutine.
 - 5.13.4. If harvesting date has been reached, record harvest data (yield and harvesting date) in an array, and then reset the state variables to zero.
 - 5.13.5. Record the state variables of each subcell in an array.
 - 5.13.6. Output the harvest data to a file on December 31.
 - 5.14. Initialize the array of flags for forced termination of cultivation (`i2flgculkiller`).

6. Judge whether spinup termination conditions have been met.
7. Output locations and number of times that water balance or heat balance anomalies or irresolvable issues occurred during calculation.

13.3 Coupled Model Preprocessing

0. Change directory to `cpl/pre/`.
1. Edit and execute `prep.sh`. This results in making several initial and parameter files in the directories `dam/ini` and `dam/dat`.

Table 13-1. List of files on state variables

File	Note	Unit
<code>DAMSTO</code>	Storage of large reservoirs	kg
<code>PNDSTO</code>	Storage of medium-size reservoirs	kg

13.4 Coupled Model Calculation

Using the coupled model enables all sorts of numerical experiments. We will introduce just two here. One is the `N_C` experiment, which assumes virtual inexhaustible water sources (non-local and non-renewable blue water) that completely meet all agricultural water demand [See Hanasaki et al. (2008a; 2010) for detail]. The `N_C` experiment can provide you with an idea of global potential agricultural water demand. The second experiment is the `LECD` experiment, which involves enabling all of H08's functions, and assumes that all agricultural, industrial, and domestic household water demands are met by rivers alone. The `LECD` experiment enables global calculation at a resolution of $1^{\circ} \times 1^{\circ}$ of the ability to obtain the desired quantity of water at the desired time from rivers.

0. Change directory to `cpl/bin/`.
1. Edit `main.sh` and execute the `N_C` experiment²⁹. The items requiring setting are listed in Tables 13-2 through 13-6. The features of the `N_C` experiment settings are as follows:
 - Set `RUN` to `N_C`.
 - Disable the reservoir operation module: Set all the items in `Input for reservoirs (Edit here)` to `NO`.
 - Enable non-local and non-renewable blue water (virtual inexhaustible water sources): Set `OPTNNB` to `yes`.

²⁹ The method for performing calculations under the FTCS scheme is as follows;

1. Confirm that there are two files, `main_ftcs.f` and `main_ftcs.sh`, in `cpl/bin` and that they have all been compiled.
2. Open `main_ftcs.sh` and edit the basic settings according to the `N_C` experiment. At this time, make sure that `SECINT2` (seconds interval for river) is set to a sufficiently short value.
3. Execute `main_ftcs.sh`, and after confirming that the calculation was successfully performed, check the basic settings in `cpl/pst/calc_mean.sh` and execute it.
4. Run the `LECD` experiment in the same way, replacing `main.sh` with `main_ftcs.sh`.

2. Change directory to `cpl/pst`, and execute `calc_mean.sh`. Then change directory again to `cpl/bin`.
3. Make sure that `map/bin/main_dam.sh` (Chapter 6) has been executed.
4. Edit `main.sh`, and execute the `LECD` experiment. The features of the `LECD` experiment settings are as follows:
 - Set `RUN` to `LECD`.
 - Enable the reservoir operation model. Set the correct path to the variables `DAMID_`, `DAMP RP`, `DAMCAP`, `DAMMON1ST`, `DAMALC`, `DAMRIVOUTFIX`, and `DAMDEMAGRFX`, `MSRCAP`, `MSRAFC`. Note that `DAMSRE` should be kept as `NO`. Look carefully at the shell script: everything is already described. You only need to add eight numbers.
 - Disable non-local and non-renewable blue water (virtual inexhaustible water sources): Set `OPTNNB` to `No`. The result of this is, even with irrigated cropland, crops will be subject to water stress if water cannot be withdrawn from rivers.

Table 13-2. Variables on subcell

Variable name	Note	Unit
NOFMOS	Number of mosaics	-
ARAFRC	Areal fraction	-
OPTLN DUSE	Land use option (See Table 13-3)	-

Table 13-3. List of land use options for subcell

Option	Note
<code>sci</code>	Single crop irrigated
<code>dci</code>	Double crop irrigated
<code>scr</code>	Single crop rain fed
<code>non</code>	Non-agricultural land

Table 13-4. List of variables on model parameters

Variable name	Note	Unit
DAYADVIRG	Days applying advanced irrigation	day
FCTPAD	Soil moisture target for paddy irrigation	-
FCTNONPAD	Soil moisture target for other than paddy	-
KNORM	k_{norm} : α of Hanasaki et al. (2006)	-
OPTKRLS	Enable k_{rls} of Hanasaki et al. (2006)	yes/no
OPTDAMRLS	Reservoir release model (Basically always set to "H06")	H06/M98/nokrls/nodem
OPTDAMWBC	Reservoir water balance calculation	yes/no

	(Basically always set to "NO")	
OPTNNB	Enable non-local and non-renewable blue water of Hanasaki et al. (2010)	yes/no
OPTHVSDOY	Fix harvesting date by file or calculate	free

Table 13-5. List of files on input map data

Files	Note	Unit
PLTDOY#	Planting day of subcell #	DOY
HVSDOY#	Harvesting day of subcell #	DOY
CRPTYP#	Crop type of subcell #	-
ARAFRC#	Areal fraction of subcell #	-
DEMIND	Industrial water demand (consumption)	kg s ⁻¹
DEMDOM	Domestic water demand (consumption)	kg s ⁻¹
ENVFLW	Environmental flow requirement	kg s ⁻¹
DAMID	Reservoir ID	-
DAMMON1ST	The 1 st month of the operating year	-
DAMCAP	Capacity of large reservoirs	kg
DAMSRF	Surface area of large reservoirs	m ²
DAMALC	Allocation coefficient (k_{alc}) of large reservoirs	-
DAMRIVOUTFIX	Mean annual river flow	kg s ⁻¹
DAMDEMAGRFIX	Mean annual agricultural water demand	kg s ⁻¹
PNDCAP	Medium size reservoir capacity	kg

Table 13-6. List of output files

File	Note	Unit
DAMINF	Inflow to large reservoirs	kg s ⁻¹
DAMOUT	Outflow from large reservoirs	kg s ⁻¹
DAMDEM	Water demand for large reservoir	kg s ⁻¹
PNDOUT	Outflow from medium reservoir	kg s ⁻¹
DEMAGR	Agricultural water requirement	kg s ⁻¹
SUPAGR	Total agricultural water withdrawal	kg s ⁻¹
SUPIND	Industrial water withdrawal	kg s ⁻¹
SUPDOM	Domestic water withdrawal	kg s ⁻¹
SUPAGRRIV	Agricultural water withdrawal from river	kg s ⁻¹
SUPINDRIV	Industrial water withdrawal from river	kg s ⁻¹
SUPDOMRIV	Domestic water withdrawal from river	kg s ⁻¹
SUPAGRPNPND	Agricultural water withdrawal from medium reservoir	kg s ⁻¹
SUPINDPNPND	Industrial water withdrawal from medium reservoir	kg s ⁻¹
SUPDOMPNPND	Domestic water withdrawal from medium reservoir	kg s ⁻¹

SUPAGRNNB	Agricultural water withdrawal from NNBW	kg s ⁻¹
SUPINDNNB	Industrial water withdrawal from NNBW	kg s ⁻¹
SUPDOMNNB	Domestic water withdrawal from NNBW	kg s ⁻¹

13.5 Coupled Model Postprocessing

Before moving to the analysis of coupled model results, check the water balance.

0. Change directory to `cpl/pst/`.
1. Edit and execute `list_watbal.sh`.³⁰

³⁰ In the case of global calculation, river water inputs and outputs may sometimes fail to balance by 2–3 km³ (0.005–0.0075%). We have not yet pinpointed and resolved the cause.

CHAPTER 14

Applied Uses

This chapter introduces a few applied uses of H08.

14.1 Global Warming

Public interest in global warming is currently (2018) at an extremely high level, and H08 was designed to be able to be applied for assessing the influence of global warming on global water resources.

But what exactly does the assessment of global warming using a model mean? To us at the present point in time, it mainly means examining the results of the model's output with respect to changes in meteorological data. This goes for H08 too. If users prepare the eight items of meteorological data listed in Table 7-1 that show a world in which global warming is taking place, along with any necessary map data, and supply H08 with these data, the impact of global warming can be assessed. In general, meteorological data that show a world in which global warming is taking place is called a climate scenario.

A great many methods for creating climate scenarios have been published, and are still being upgraded each year. A decade ago, modelers developed simple climate scenarios by themselves, but this is no longer the case; specialists gathered for several large projects and developed detailed scenarios applying the latest techniques. The following sections introduce the method of how to conduct climate change simulation using the Inter Sectoral Impact Model Intercomparison Project Fast Track (ISIMIP-FT; Hempel et al. 2013) for past and future periods. In many cases past and future periods cover 20 years and longer but, for the sake of brevity, below we take the year 1979 for the past and 2079 for the future.

1. Visit the global meteorological data server³¹ and download ISIMIP-FT data. The download condition is as listed in Table 14-1.
2. Change directory to `met/pre/`, and edit `prep_ISIMIPFT.sh`. Set `YEARMIN=1979` and `YEARMAX=1979`. `PID1` and `PID2` must be set as shown in the personal download page. Execution of this shell script results in generating meteorological variables in `met/dat`, for example, `mesc851_2079mdd.hlf`. Note that `mesc` is short for `MIROC ES` `ESM CHEM` and `851_` for `RCP8.5 Ensemble 1`.
3. Again edit and execute `prep_ISIMIPFT.sh` by setting `YEARMIN=2079` and `YEARMAX=2079`. This results in generating meteorological variables in `met/dat` for the future period.
4. Edit and execute `prep_Rainf.sh`. This creates rainfall files from precipitation (`pr`) and snowfall (`prsn`) data, which are provided by the data developers. Set `PRJ=mesc`, `RUN=hs1_`, and `YEARMIN=YEARMAX=1979`. This results in creating rainfall files for 1979.
5. Similarly, edit and execute `prep_Rainf.sh` for `YEARMIN=YEARMAX=2079`.

³¹ <http://h08.nies.go.jp> (user: cmip5; password: CMIP5)

6. Change directory to `lnd/bin/` and edit the `Meteorological input` section of `main.sh`. In a standard simulation, humidity is expressed in specific humidity, therefore, `QAIR` will contain file names, and `RH` will contain `NO`. In a climate change simulation, humidity is expressed in relative humidity, therefore, `QAIR` will contain `NO`, and `RH` will contain file names.
7. Run the simulation for the past (1979). Edit the `Basic Settings`, `Meteorological input`, and `Climate change output` sections of `main.sh`. Set `PRJ=mesc`, `RUN=LR__`, `PRJMET=mesc`, `RUNMET=hs1_`, and `YEARMIN=YEARMAX=1979`. Execute `main.sh`.
8. Run the simulation for the future (2079). Edit the `Basic Settings`, `Meteorological input`, and `Climate change output` sections of `main.sh`. Set `PRJ=mesc`, `RUN=LR__`, `PRJMET=mesc`, `RUNMET=851_`, and `YEARMIN=YEARMAX=2079`. Execute `main.sh`.
9. You can run `riv/bin/main.sh`, `crp/bin/main.sh`, and `cp1/bin/main.sh` similarly.
10. Change directory to `cp1/pst/`, and edit and execute `list_watbal.sh`.

Table. 14-1 Data to retrieve

Options	Settings
Dataset	ISIMIP-FT (Hempel et al. 2013)
Model	MIROC-ESM-CHEM/mesc (standard)
Scenarios	Historical and RCP8.5 (85)
Year:	1979-1979 for historical, 2079-2079 for RCP8.5
Ensemble run	----
Variable	rlds (longwave downwelling radiation), ps (surface air pressure), hurs (near surface relative humidity), rsds (shortwave downwelling radiation), prsn (snowfall rate), tas (surface air temperature), sfcWind (wind speed), pr (precipitation flux)
Temporal resolution	Daily
Spatial domain	Original
Select mask	Don't select
Mean calculation	Don't calculate
File format	NetCDF
Compress	Yes (gzip compression)

14.2 Regional Model

H08 was originally developed as a global model, but it can also be used as a domain model for chosen domains. The basic method for doing so is as follows:

1. Decide the region to be researched, and decide the spatial parameters shown in Table 5-1.
2. Carry out the processes from Chapter 6 on, editing the **Geographical settings** section of shell scripts carefully as you go.

H08 source code and shell scripts are not dependent on calculation domain or spatial resolution. In other words, they will work with whatever changes you make in the settings in Table 5-1. However, the one exception is **main.f**, whose array size (NL in Table 5-1) is written into the source code. As a result, when you change the resolution, you need to edit **main.f**, changing

```
parameter (n01=259200)
```

and then recompiling.

Appendix 1

H08 Analysis Tools

H08 outputs large amounts of binary and text data. H08 Analysis Tools is a package of programs for efficiently processing these data. It enables easy plotting and calculation of H08 File Format outputted files. Binary data can, of course, be analyzed and plotted with software such as GrADS, and text data with Microsoft Excel, etc.

A1.1 Textbook

See “H08 Analyzer Manual” (Saito and Hanasaki, 2012) for detail.

Appendix 2

Coding Rules

H08 is composed of a large volume of Fortran source code that is coded according to certain rules. We will explain the basics here.

A2.1 Naming Variables

Variable names have the following rules. The first character denotes the type of variable: **i**, **r**, **c**, **n**, **p**, and **s**, respectively, stand for integer, real, and character variables, and number, parameter, and string constants. The next character in the variable name is an integer denoting the dimension of the array, namely **0**, **1**, **2**, or **3**, respectively, for a variable, 1-dimensional array, 2-dimensional array, and 3-dimensional array. The remainder is made up of abbreviations of the variables concerned. Although there are exceptions, as a rule, variables are expressed with three characters. For example, **irg** denotes irrigation, and **ara**, area. To sum up, **rlirgara** would be a real number 1-dimensional array for irrigated area.

A2.2 Time Loops

In H08, only the main process can push the clock forward. Normally, the main process contains the following time loops:

```
do i0year=i0yearmin,i0yearmax
  do i0mon=i0monmin,i0monmax
    do i0day=1,i0daymax
      do i0sec=i0secint,n0secday,i0secint

        end do
      end do
    end do
  end do
```

The loops advance year, month, day, and second, respectively. Year, month, and day advance in single units, while seconds advance in intervals of **i0secint** up to **n0secday**, which is 1 day's worth of seconds (=86400 sec). For example, **i0sec=10800 [sec]=3 [hour]** means that seconds will advance in 3-hour intervals.

A2.3 Comments

Comments start with the letter **c**. Though it is not that well-known, comments can also start with **d**. In such cases, depending on the compile options, lines starting with **d** can be either comments or code. The code used exclusively for debugging starts with a **d** in the first column.

A2.4 Declaration and Initialization of Variables and Arrays

The variables and arrays are declared and initialized as shown in Table A2-1. Initialization is particularly important in programming. If you find any variables that are not initialized, please let us know.

Table A2-1. Declaration and initialization of variables and arrays

Type	Explanation	Initialization
parameter (array)	Parameters to set the size of arrays	NA
parameter (physical)	Physical parameters	NA
parameter (default)	Commonly used parameters across the entire H08 code	NA
index (array)	Index of arrays	NA
index (time)	Index or loops	NA
temporary	Temporal variables and arrays	Initialized at the beginning of each source code
function	Functions	Initialized at the beginning of each source code of functions
in (set)	Input data that are given from arguments and namelist	Initialized by arguments and namelist
in (map)	Input data that are given from map files	Initialized by files
in (flux)	Input data that are given from data files	Initialized by files
state variable	State variables	Initialized by initial files
out	Output data to files	Initialized at the beginning of each source code
local	Local variables and arrays	Initialized at the beginning of each source code
namelist	Namelist	Initialized with namelist

A2.5 Dimension of arrays

In most cases, one-dimensional arrays have the dimension of space, typically the size of `(n01)`. Exceptionally, the crop parameter arrays that appear in `cp1/bin/main.f` and `crp/bin/main.f` have the dimension of crop types (e.g., `rlicnum(n0ram)`).

In many cases, two-dimensional arrays have the dimensions of space and temporal

resolution, typically the size of `(n0l,0:n0t)`. Here, `n0t=0, 1, 2, and 3` indicate instantaneous value, daily mean, monthly mean, and annual mean respectively. The most important exceptions that should be kept in mind are as follows:

- crop variables in `cp1/bin/main.f` and `crp/bin/main.f`. They have dimensions of space and mosaic (e.g., `r2huna(n0l,0:n0m)`).
- crop parameters in `cp1/bin/main.f` and `crp/bin/main.f`. They have dimensions of variable ID and crop types (e.g., `r2crppar(24,n0swim)`).
- target soil moisture in `cp1/bin/main.f`. It has dimensions of space and crop intensity (e.g., `r2target(n0l,n0c)`).
- meteorological variables in `crp/bin/main.f`. They have dimensions of space and day of year. (`r2tair, r2sdown, r2potevap, r2evap, r2hvsoy, r2crpday, r2yld, r2cwd, r2cws, r2regfd(0:n0llnd,n0doy)`).

In most cases, three-dimensional arrays have the dimensions of space, time, and mosaic, typically the size of `(n0l,0:n0t,0:n0m)`.

Appendix 3

Files in adm

This section introduces the files found in `adm/bin/` that were not mentioned in the main text.

A3.1 backup.sh

This is a shell script for backing up source code (`.f` or `.F` extension) and shell scripts (`.sh` extension) in the directories of H08. Because it takes very little time, it is useful for everyday backups. Files will be backed up in the destination specified in `DIROUT`.

A3.2 clean.sh

This is a shell script for searching all of the `out/` directories in the directories of H08 and deleting files that meet the specified conditions. It is useful, for example, for batch deletion of files created during debugging.

A3.3 count.sh

This is a shell script for counting the lines of source code (`.f` or `.F` extension) and shell scripts (`.sh` extension) in H08 directories. It is useful for answering "just curious" questions such as "How many lines does the whole H08 source code contain?"

A3.4 installer.sh

This is a shell script for creating H08 installer (`install.sh`). Run this script after executing `backup.sh`. This adds an installer to your backup directory. It is useful for providing other people with H08 source code.

References

- Bhumralkar, C. (1975), Numerical experiments on the computation of ground surface temperature in an atmospheric general circulation model, *J. Appl. Meteorol.*, 14, 1246-1258.
- Dirmeyer, P. A., X. A. Gao, M. Zhao, Z. C. Guo, T. K. Oki, and N. Hanasaki (2006), GSWP-2 - Multimodel analysis and implications for our perception of the land surface, *B. Am. Meteorol. Soc.*, 87, 1381-1397.
- Döll, P., and S. Siebert (2002), Global modeling of irrigation water requirements, *Water Resour. Res.*, 38, 1037, doi:10.1029/2001WR000355.
- Döll, P., and K. Fiedler (2008), Global-scale modeling of groundwater recharge, *Hydrol. Earth Syst. Sci.*, 12(3), 863-885, doi: doi:10.5194/hess-12-863-2008.
- Döll, P., H. Hoffmann-Dobrev, F. T. Portmann, S. Siebert, A. Eicker, M. Rodell, G. Strassberg, and B. R. Scanlon (2012), Impact of water withdrawals from groundwater and surface water on continental water storage variations, *J. Geodyn.*, 59-60(0), 143-156, doi: 10.1016/j.jog.2011.05.001.
- Hanasaki, N. (2006), Development of a global water cycle model considering anthropogenic activities and estimation of temporal variation in global water resources, 136 pp, Doctoral Dissertation, The University of Tokyo.
- Hanasaki, N., S. Kanae, and T. Oki (2006), A reservoir operation scheme for global river routing models, *J. Hydrol.*, 327, 22-41.
- Hanasaki, N., S. Kanae, T. Oki, K. Masuda, K. Motoya, and K. Tanaka (2007a), An integrated model for assessment of global water resources. Part 1: Input meteorological forcing and natural hydrological cycle modules, *Hydrol. Earth Syst. Sci. Discuss.*, 4, 3535-3582.
- Hanasaki, N., S. Kanae, T. Oki, and N. Shirakawa (2007b), An integrated model for assessment of global water resources. Part 2: Anthropogenic activities modules and assessments, *Hydrol. Earth Syst. Sci. Discuss.*, 4, 3583-3626.
- Hanasaki, N., S. Kanae, T. Oki, K. Masuda, K. Motoya, N. Shirakawa, Y. Shen, and K. Tanaka (2008a), An integrated model for the assessment of global water resources - Part 1: Model description and input meteorological forcing, *Hydrol. Earth Syst. Sci.*, 12, 1007-1025.
- Hanasaki, N., S. Kanae, T. Oki, K. Masuda, K. Motoya, N. Shirakawa, Y. Shen, and K. Tanaka (2008b), An integrated model for the assessment of global water resources - Part 2: Applications and assessments, *Hydrol. Earth Syst. Sci.*, 12, 1027-1037.
- Hanasaki, N., T. Nitta (2009) UNIX/FORTRAN/Bourne Shell Script Self-study Text, 78 pp, National Institute for Environmental Studies, Tsukuba, Japan.
- Hanasaki, N., T. Inuzuka, S. Kanae, and T. Oki (2010), An estimation of global virtual water flow and sources of water withdrawal for major crops and livestock products using a global hydrological model, *J. Hydrol.*, 384, 232-244.
- Hanasaki, N., S. Yoshikawa, K. Kakinuma, and S. Kanae (2016), A seawater desalination scheme for global hydrological models, *Hydrol. Earth Syst. Sci.*, 20(10), 4143-4157, doi: 10.5194/hess-20-4143-2016.
- Hanasaki, N., S. Yoshikawa, Y. Pokhrel, and S. Kanae (2018), A global hydrological simulation to specify the sources of water used by humans, *Hydrol. Earth Syst. Sci.*, 22(1), 789-817.

- Hempel, S., K. Frieler, L. Warszawski, J. Schewe, and F. Piontek (2013), A trend-preserving bias correction – the ISI-MIP approach, *Earth Syst. Dynam.*, 4(2), 219-236, doi: 10.5194/esd-4-219-2013.
- Kitamura, S., S. Yoshikawa, and S. Kanae (2014), Global water resource model taking into account irrigation water withdrawal from nonlocal water resources, *J. JSCE (B1)*, 70(4), I_253-I_258.
- Krysanova, V., F. Wechsung, J. Arnold, R. Srinivasan, and J. Williams (2000), SWIM (Soil and Water Integrated Model) User Manual, Potsdam Institute for Climate Impact Research.
- Leff, B., N. Ramankutty, and J. A. Foley (2004), Geographic distribution of major crops across the world, *Global Biogeochem. Cy.*, 18, Gb1009, doi:10.1029/2003GB002108.
- Manabe, S. (1969), Climate and the ocean circulation 1. The atmospheric circulation and the hydrology of the earth's surface, *Mon. Weather Rev.*, 97-11, 739-774.
- Monfreda, C., N. Ramankutty, and J. A. Foley (2008), Farming the Planet. Part 2: The Geographic Distribution of Crop Areas and Yields in the Year 2000, *Glob. Biogeochem. Cycles*, GB1022, doi:10.1029/2007GB002947.
- Ramankutty, N., A. Evan, C. Monfreda, and J. A. Foley (2008), Farming the Planet. Part 1: The Geographic Distribution of Global Agricultural Lands in the Year 2000, *Glob. Biogeochem. Cycles*, 22, doi: GB1003, doi:10.1029/2007GB002952.
- Saito, Y. and N. Hanasaki (2012), H08 manual analyzer's edition, 81 pp, National Institute for Environmental Studies, Tsukuba, Japan.
- Oki, T., T. Nishimura, and P. Dirmeyer (1999), Assessment of annual runoff from land surface models using Total Runoff Integrating Pathways (TRIP), *J. Meteorol. Soc. Jpn.*, 77, 235-255.
- Robock, A., K. Y. Vinnikov, C. A. Schlosser, N. A. Speranskaya, and Y. K. Xue (1995), Use of Midlatitude Soil-Moisture and Meteorological Observations to Validate Soil-Moisture Simulations with Biosphere and Bucket Models, *J. Climate*, 8, 15-35.
- Siebert, S., P. Döll, J. Hoogeveen, J. M. Faures, K. Frenken, and S. Feick (2005), Development and validation of the global map of irrigation areas, *Hydrol. Earth Syst. Sc.*, 9, 535-547.
- Shirakawa, N. (2005), Global estimation of environmental flow requirement based on river runoff seasonality, *Annual Journal of Hydraulic Engineering*, 49, 391-396, in Japanese with English abstract.
- Weedon, G. P., G. Balsamo, N. Bellouin, S. Gomes, M. J. Best, and P. Viterbo (2014), The WFDEI meteorological forcing data set: WATCH Forcing Data methodology applied to ERA-Interim reanalysis data, *Water Resour. Res.*, 50, 7505-7514, 10.1002/2014WR015638.