H08 Manual
User's Edition

Naota Hanasaki
Takahiro Yamamoto

Acknowledgments

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 space of 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 the data used to test it too owes its existence to the efforts of people throughout the world who gathered it. We are accordingly deeply indebted to all of those on whose achievements H08 was built.

This work was supported by 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 JST-JICA SATREPS project entitled “Integrated Study Project on Hydro-Meteorological Prediction and Adaptation to Climate Change in Thailand” (IMPAC-T).

Version information

See http://h08.nies.go.jp/ for 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 31, 2011	Source Code and Manual Users Edition Ver. 20110131 released (Minor update)
January 1, 2012	Source Code and Manual Users Edition Ver. 20120101 released (Major update)
August 1, 2012	Source Code and Manual Users Edition Ver. 20120801 released (Minor update)
May 1, 2013	Source Code and Manual Users Edition Ver. 20130501 released (Minor update)

Contents

Chapter 1
Introduction

This chapter explains what exactly H08 is.

1.1 Introduction

Water is essential to all life forms. Water is indispensible to us humans too – to our survival, daily life, and production activities. Global water consumption started rising rapidly from the mid-20th century, and is expected to continue to grow in line with population growth and economic development up to the mid-21st century at least. Global warming is also predicted to change rain and snow precipitation patterns, cause snow to thaw earlier, and promote 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 has been working on the development of H08,[1] a global water resource model (computer program) for examining such global water resource issues from the present 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 2 papers introduce the background to H08 development, review preceding research, and explain the theory behind the model, test results, application to global water resource assessment, key issues, and so forth. Hanasaki (2006) provides similar information, but should be treated only as background information since 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). Since the H08 model introduced in this

[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.

manual now includes these improvements and extensions, Hanasaki et al. (2010) can also be referred to.

1.3 Disclaimer and Request to Readers

Disclaimer: The developers have developed the source code with 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.

Request: Firstly, this manual was written by the developers themselves. We have endeavored to make is 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 as much detail as possible on what you found difficult to understand. H08, moreover, harbors limitless scope for improvement and expansion, and so 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., Kanae, S., Oki, T., Masuda, K., Motoya, K., Shirakawa, N., Shen, Y. and Tanaka, K., 2008. An integrated model for the assessment of global water resources - Part 1: Model description and input meteorological forcing. Hydrol. Earth Syst. Sci., 12(4): 1007-1025

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

Chapter 2
Computer Environment Required for Use of H08

This chapter explains the kind of 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 UNIX, Linux, or MacOS.[2]
2. Supports Fortran compilers.[3]

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

2.2 Required Free Software

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

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

[2] Systems confirmed to date as compatible include: Sun Microsystems Sun Fire V440 (UltraSPARC IIIi 1.593 GHz, Solaris 9), SGI Altix 450 (Intel Itanium 2, Red Hat Linux), Dell PowerEdge 6850 (Intel Xeon, Red Hat Linux), Apple Macbook Pro 13 inch (Intel Core 2 Duo, MacOS 10.5)

[3] Compilers confirmed to date as compatible: Intel Fortran Compiler and Sun Fortran Compiler (both commercial), gfortran and g95 (both free software)

Chapter 3 Installation
Installation

This chapter explains how to install H08.

3.1 H08 Installation

H08 installation procedure is as follows:

1. There is one H08 installation files: H08 yyyymmdd.tar.gz. First copy this file to the directory in which you want to install the program.
2. Uncompress the file.
   % gunzip H08_yyyymmdd.tar.gz
   % tar xf H08_yyyymmdd.tar
3. Move to the directory H08 yyyymmdd
4. Execute install.sh as follows:
   % sh ./install.sh
5. After a short while, installation will end with this message appearing:
   please set adm/Mkinclde
   afterward, continue again
6. 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.
7. Edit Mkinclude. The main shell variables set in Mkinclude are explained below.
   Shell variable INC: Specify path to the NetCDF include file[4].
   Shell variable LIB: Specify path to the NetCDF library file[5].
   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 byterecl[6]
8. Once you have created the Mkinclude file in the adm directory, execute install.sh once again[7]. This will result in the compilation of all the programs .

3.2 H08 Preferences

[4] 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.

[5] 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.

[6] When binary data is 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 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.

[7] For ubuntu users, edit and execute ${DIRH08}/bin/htdraw.sh and htdrawts.sh, and then install.sh.

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

```
% echo $SHELL
```

If /bin/bash is displayed as a result, your preferences file is ~/.bashrc. Read and edit the adm/sample.bashrc comment carefully, 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. 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

Endian is a binary encoding format that exists in 2 types─big and little. H08 was developed in a Sun Microsystems UltraSPARC CPU environment that uses big-endian, but Intel's CPUs (Itanium2, 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 just by reading it, 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 3 layers, so we call them 1st level directories, 2nd level directories, and 3rd level directories according to layer. Figure 4-1 shows the structure as far as the 2nd level directories.

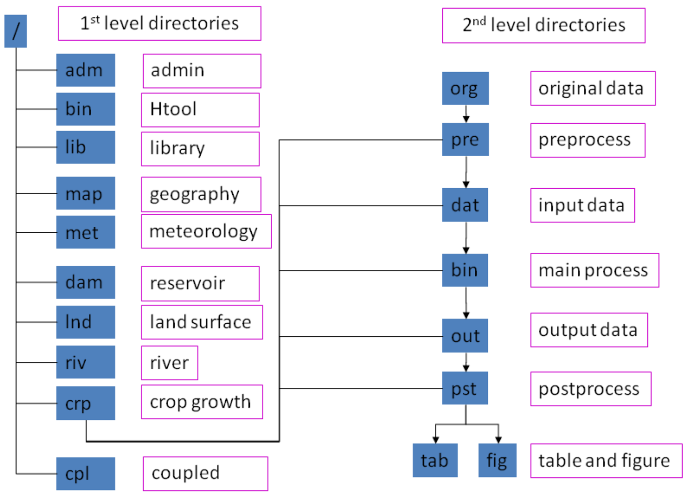

Figure 4-1 Directory structure of H08

4.2 1st Level Directories

Table 4-1 lists 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

You should be able to see that each of the 1st level directories excluding adm/, bin/, and lib/—in other words, map/, met/, lnd/, riv/, crp/, dam/, and cpl/—contains much the same 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 in using H08.

The most important 2nd level directories are dat/, bin/, and out/. They respectively contain input data, the main process, and output data. Files in dat/ are input data that is 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 Format that we explain in the next chapter.

Table 4-2 2nd level directories

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

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

The output files written to out/ are in H08 Format (binary), and cannot be read by us 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 theses 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 module	lnd/bin/main.sh	lnd/bin/main	lnd/bin/calc_leakyb.o
River module	riv/bin/main.sh	riv/bin/main	riv/bin/calc_outflw.o
Crop growth module	crp/bin/main.sh	crp/bin/main	crp/bin/calc_crpyld.o
Reservoir operation module	dam/bin/main.sh	dam/bin/main	dam/bin/calc_resope.o
Coupled model	cpl/bin/main.sh	cpl/bin/main	All of above

4.5 pre/ and pst/ Directories

The 2nd level directories pre/ and pst/ also contain a great many executable shells. pre/ contains executable shells, named pre/prep*.sh, for converting original data written in formats other than H08 Format into H08 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 1 executable file.[8]

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

```
% make all
```

This will cause all of the programs[9] 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 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.

[8] For make basics, see Chapter 1 of *Managing Projects with make*, Second Edition by Andrew Oram and Steve Talbott, pub. O’Reilly Media.
[9] Actually, the programs defined with the shell variable OBJS 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 8 characters. Changing 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 3 spatial dimensions and 1 temporal dimension, making for a total of 4, but we would like you to imagine a further 2 dimensions. Firstly, 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, the quantities representing temperature, precipitation and various other factors in a certain place at a certain time can be all 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 6 dimensions (Figure 5-1).

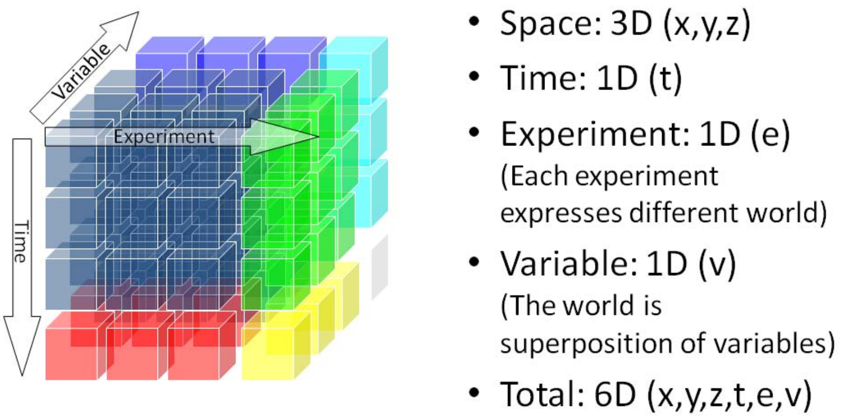

Figure 5-1 Dimensions of data handled by H08

5.2 File Types Handled by H08

If we suppose, for argument's sake, that today's computers could handle the above 6-dimensional array, all H08 input and output data could be contained in 1 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. As such, we divide files into more easily handled sizes for storage and use in H08.

There are basically only 2 methods for dividing 6-dimensional data in an effective way from the perspective of hydrology or water resources. One method is to divide the data into 4 fixed dimensions of time, experiment, variable, and altitude, and 2 flexible dimensions of latitude and longitude for geographical distribution data. The other method is to divide the data into 5 fixed dimensions—the 3 spatial dimensions plus experiment and variable—and 1 flexible dimension of time. The data for the former would best be 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 accordingly refer to the former as H08 Format 2D, and to the latter as H08 Format 1D.

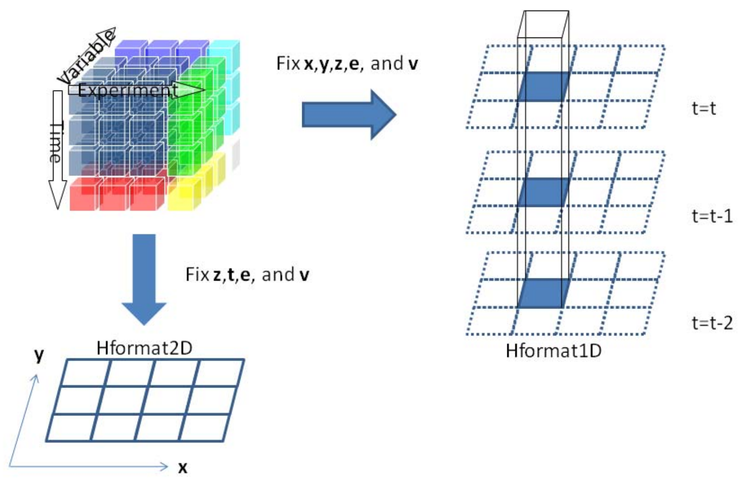

Figure 5-2 H08 Format 1D and H08 Format 2D

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 spatial resolution is 45° latitude × 45° longitude (Figure 5-3). A global spatial domain can be displayed by setting maximum and minimum longitude (lonmax and lonmin) to 180° and −180°, and latitude (latmax and latmin) to 90° and −90°. Spatial resolution can, assuming equal intervals, be displayed by setting the number of cells in the longitudinal axis (nx) at 8, and in the latitudinal axis (ny) at 4.

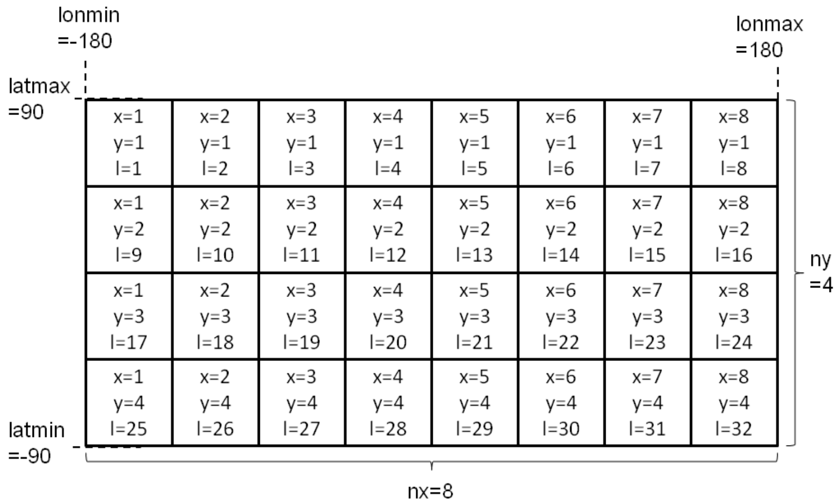

Figure 5-3 Example of spatial domain

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 (pay attention to direction), with Y taking the value of 1 up to NY. The L coordinates are a combination of X and Y coordinates that can basically be expressed as follows:

$$L=NX\times(Y-1)+X \tag{5-1}$$

L takes the value of 1 to NL (= NX × NY). Arranged in this way, a grid cell can be expressed either in terms of 2 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 basically uses a conversion table to convert L coordinates into X coordinates and one to convert L coordinates into Y coordinates.

Summing up the above, space in H08 is defined by the 9 variables listed in Table 5-1.

Table 5-1 Variables defining 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	conversion table L to X
L2Y	c0l2y	conversion table L to Y

Incidentally, L coordinates may be defined other than by 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 8 of the 32 grid cells are land, L coordinates may be allocated only to the land cells. The data distributed with the Global Soil Wetness Project (GSWP2; Dirmeyer et al., 2006), for example, was global 1°×1°, but because sea area cells contain only missing values, the data consists not of all cells (NX×NY=360×180=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 to 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.

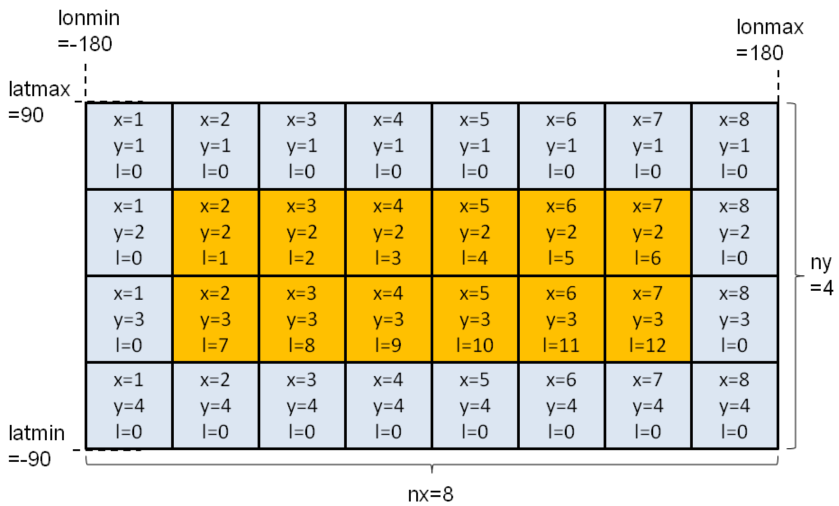

Figure 5-4 Example of spatial domain allocating L to only land

5.4 Expressing Time

We will now consider how to express time. H08 uses the Gregorian calendar—in other words, the standard calendar that includes leap years. It expresses time as year, month, day or year, month, day, hour.[10] 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 an hour to a file.

The time 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). The conservation law is, accordingly, 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.

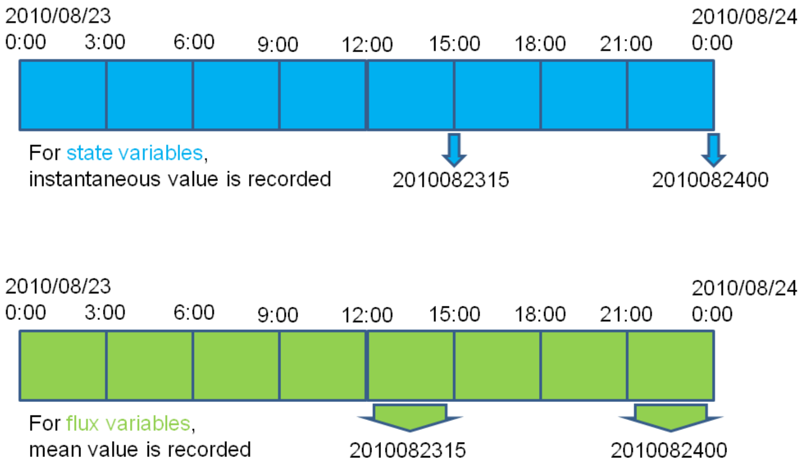

Figure 5-5 Time and state/flux variables

[10] However, within the H08 Fortran programs, calculation is carried out using not year, month, day, hour, but year, month, day, second. The developers are still at a loss as to whether to eliminate this double standard.

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 Format 2D

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

H08 Format 2D files are binary files using 4-byte real numbers. They have no header information, and consist just 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 characters or 22 characters in length, the characters representing project name (4 characters), experiment name (4 characters), year (4 characters), month (2 characters), day (2 characters), hour (2 characters), and file extension (4 characters). As mentioned earlier, time, experiment, variable, and altitude dimensions are fixed in H08 Format 2D. Specifying year, month, day, hour fixes the time, while specifying project name and experiment name fixes the experiment dimension. Specifying the relevant 3rd level directory name (8 characters) fixes the variable dimension. H08 Format 2D file names accordingly express fixed dimensions, and 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 4 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°
.hlf	Global	0.5°×0.5°
.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 Format 1D

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

H08 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, the characters representing project name (4 characters), experiment name (4 characters), location ID (8 characters), and file extension (4 characters). As mentioned earlier, spatial, experiment, and variable dimensions are fixed in H08 Format 1D. Specifying ID fixes the spatial dimensions, while specifying project name and experiment name fixes the experiment dimension. Specifying the relevant 3rd level directory name (8 characters) fixes the variable dimension. Unlike H08 Format 2D files, the only file extension is .txt.[11]

To sum up the above, the name of a file containing, for example, 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

[11] 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 consists of map data and meteorological data. This chapter explains how to create map data.

6.1 Creating H08 model 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	map/dat/l2x_l2y_/	-
GRDARA	Grid cell area	map/dat/grd_ara_/	m^2

1. Change directory to map/pre/.
2. Edit prep_basmap.sh.
3. Execute prep_basmap.sh as follows:

```
% sh prep_basmap.sh
```

As a result, files l2x.one.txt and l2y.one.txt for converting L coordinates into X and Y coordinates will be outputted to directory map/dat/l2x_l2y_/. A file containing area information for each cell (grdara.one) will also be outputted to directory map/dat/grd_ara_/.

・Column 1

● Checking Output Data 1: Analyzing Map Data

As you learned in the last chapter, the file map/dat/grd_ara_/grdara.one created through carrying out the above processes is written in H08 Format 2D. Let's take a look at this file. First check file size to make sure that grdara.one has been outputted correctly. In cases like this covering the whole world at a resolution of 1°×1°, NL = 360 x 180 = 64,800. Because recording 1 real number requires 4 bytes, the file size should be 64,800 × 4 = 259,200 bytes. Use the ls –l command to check that this is the actual size.

Next, take a look inside grdara.one. Files written in H08 Format can be easily analyzed and plotted using H08 Analysis Tools. See Saito and Hanasaki (2012) for details. Using H08 Analysis Tools as it is requires a great many arguments here, and so to save space below, we use the shortened input for H08 Analysis Tools provided in Appendix A of Saito and Hanasaki (2012). 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/
% sumone ../../map/dat/grd_ara_/grdara.one
```

You should obtain 5.09×10^{14} as a 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:

```
% maxone ../../map/dat/grd_ara_/grdara.one
% minone ../../map/dat/grd_ara_/grdara.one
```

You should obtain 1.23×10^{4} km^2 and 1.09×10^{2} 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:

```
% makecpt -T0/15000000000/5000000000 -Z > temp.cpt
% one2eps ../../map/dat/grd_ara_/grdara.one temp.cpt temp.eps
% htconv temp.eps temp.tif rot
```

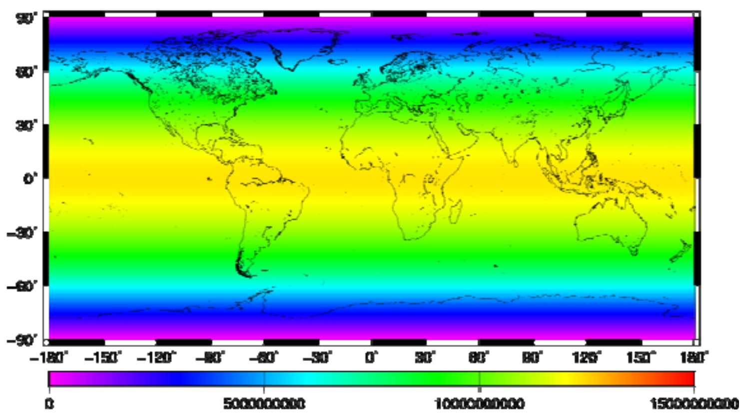

Figure 6-1 A world map of grdara.one [m^2]

6.2 Creating the Map Data Required to Use the Land Module

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

Table 6-2 Data Required to Use the Land Module

File name	Note	Directory	Unit
LNDMSK	Land sea mask	map/dat/lnd_msk_/	-
LNDARA	Land area	map/dat/lnd_ara_/	m^2

1. Download map-org-GSWP2.tar.gz[12] from the file server[13].
2. Put the file to map/org and uncompress it. If you don't have map/org, then make a new directory. Now you will have map/org/GSWP2.
3. Change directory to map/pre/.
4. Edit and execute prep_lnd_GSWP2.sh.[14] GSWP2 in the file name signifies that land and sea distribution conforms to GSWP2 data.

As a result of doing the above, a file that divides the spatial domain into land and sea (lndmsk.GSWP2.one) is created in directory map/dat/lnd_msk_. Figure 6-2 shows the result of plotting this file with htdraw (short form is one2eps). Land domain cells contain 1, and sea domain cells contain 0. You can see from this map that Antarctica is treated as sea.

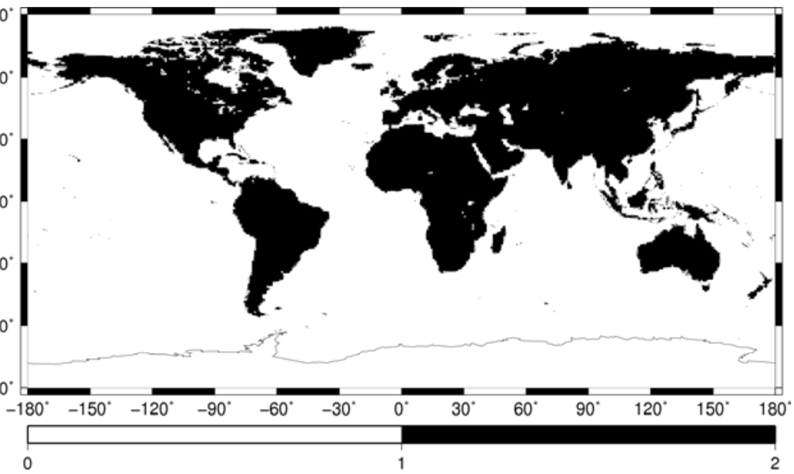

Figure 6-2 A map of land sea mask. One indicates land, and zero indicates sea.

A file containing land area (lndara.GSWP2.one) is also created in 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}).

12 In case you carry out 0.5 degree global simulation, use map-org-WATCH.tar.
13 See H08 web site for details.
14 For 0.5 degree global simulation, use prep_lnd_WATCH.sh.

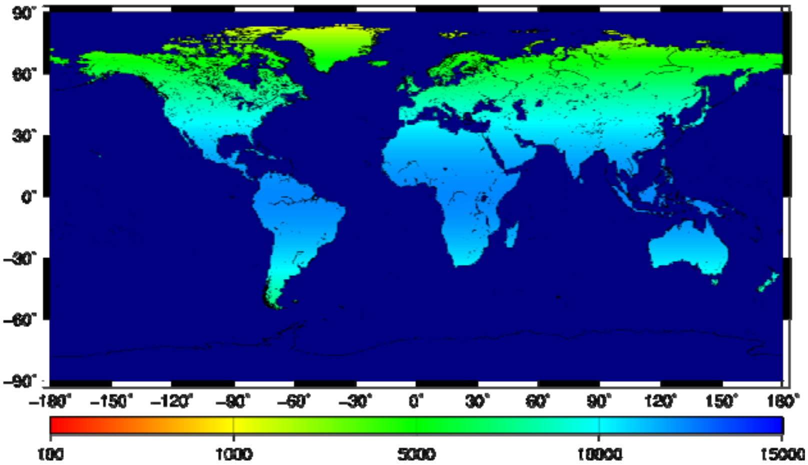

Figure 6-3 A map of land area [km^2]. Note that sea domain cells contain the value assigned to missing data (10^{20}).

6.3 Creating the Map Data Required to Use the River Module

Of the 6 items of map data shown in Table 6-3, 4 are required to use the river module. Use the process detailed below to create this data.

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

File name	Note	Directory	Unit
FLWDIR	Flow direction	map/dat/flw_dir_/	-
RIVSEQ	River Sequence	map/out/riv_seq_/	-
RIVNXL	The next downstream cell's L coordinate	map/out/riv_nxl_/	-
RIVNXD	Distance to the next downstream cell	map/out/riv_nxd_/	m

Table 6-4 Useful map data on river module

File name	Note	Directory	Unit
RIVARA	Catchment area	map/out/riv_ara_/	m^2
RIVNUM	River ID	map/out/riv_num_/	-

1. Change directory to map/pre/.
2. Edit and execute prep_riv_GSWP2.sh.[14]

This will result in the creation of a flow direction file (flwdir.GSWP2.one) in 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. 1 to 8 show flow direction in 45° segments from

[14] For 0.5 degree global simulation, use prep_riv_WATCH.sh.

north to northwest. 0 and 9 are special numbers, with 0 signifying that a cell is sea, and 9 that it is river mouth.

Next,

3. Change directory to map/bin/.
4. Edit and execute main_riv.sh.

This results firstly in the creation of a river sequence file (rivseq.GSWP2.one) in directory map/out/riv_seq_/. In this file, 1 signifies headwaters, 2 immediately downstream from headwaters, 3 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.GSWP2.one) is created in directory map/out/riv_nxl_/. This file shows the L coordinate of the next downstream cell. Lastly, a file giving distance to the next downstream cell file (rivnxd.GSWP2.one) is created in directory map/out/riv_nxd_/. This file provides direct distances from the center of a cell to the center of the next downstream cell.

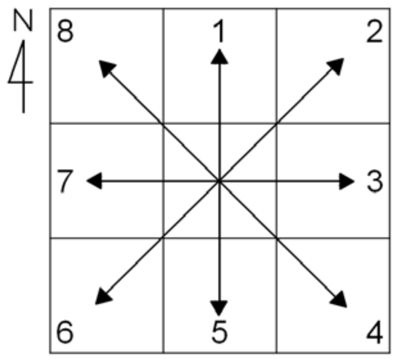

Figure 6-4 Flow direction for each cell with the numbers 0 to 9. 1 to 8 show flow direction in 45° segments from north to northwest. 0 and 9 are special numbers, with 0 signifying that a cell is sea, and 9 that it is river mouth.

Though they are not essential for using the river module, a further 2 files shown in Table 6-4 that are useful for analysis are created. The catchment area file (rivara.GSWP2.one) created in 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 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, Congo 2, and Mississippi 3.

6.4 Creating the Map Data of national border and population

Information such as national border and population is indispensable in water resources assessment. The 4 files shown in Table 6-5 are required for basic geographic analysis. Use the process detailed below to create these files.

Table 6-5 Basic geographic data

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

1. Following the instruction of map/pre/prep_map_C05_nat.sh, download the original data for national boundary and population and save into the directory map/org/C05. Make sure to prepare map/dat/nat_cod_/C05______20000000.txt[15].
2. Edit and execute map/pre/prep_map_C05_nat.sh[15]
3. National mask is outputted to map/dat/nat_msk_.
4. Edit and execute map/pre/prep_map_C05_pop.sh
5. Total population is outputted to map/dat/pop_tot_.

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

The 4 files shown in Table 6-5 are required to use the crop growth module. Use the process detailed below to create these files.

Table 6-5 Map data required to use the crop growth module

File name	Note	Directory	Unit
IRGARA	Irrigated area	map/dat/irg_ara_	m^2
CRPINT	Cropping intensity	map/dat/crp_int_	-
IRGEFF	Irrigation efficiency	map/dat/irg_eff_	-
CRPARA	Cropland area	Map/dat/crp_ara_	m^2
HVSARA	Fraction of harvested area according to crop type	map/dat/hvs_ara_	-

[15] You will be able to successfully prepare a file showing the name of countries and their unique IDs for CIESIN data. However, to run the scripts in Subsection 6.5, you need to add few lines, because the name of countries vary for datasets. A complete C05_____20000000.txt is available at the file server.

[15] Maps for 0.5 degree global simulation will be automatically generated.

1. Obtain the original data map-org-DS02.tar.gz. Also, following the instruction shown in map/pre/prep_crp_R08M08S05.sh, download the original data of Ramankutty et al.(2008), Monfreda et al. (2008), and Siebert et al. (2005).
2. Change directory to map/pre/.
3. Edit and execute prep_crp_R08M08S05.sh[17]

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 directories map/dat/crp_ara_, map/dat/pas_ara_, map/dat/hvs_ara_, and map/dat/irg_ara_ respectively.

4. Edit and execute prep_crp_DS02.sh.

This results in the outputting of Döll and Siebert (2002) irrigated area (DS02____00000000.one), cropping intensity (DS02____00000000.one), and irrigation efficiency (DS02____00000000.one) files in directories map/dat/irg_ara_, map/dat/crp_int_, and map/dat/irg_eff_ respectively.

5. Change directory to map/bin.
6. Edit and execute calc_crptyp.sh.

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

7. Edit and execute calc_crpfrc.sh

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

6.6 Creating the Map Data Required to Use the Reservoir Module

The 5 files shown in Table 6-6 are required to use the reservoir module. The 6 files in Table 6-7 are also useful to the understanding of the features of the data. Use the process detailed below to create these files.

[17] Maps for 0.5 degree global simulation will be automatically generated.

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

File name	Note	Directory	Unit
DAMCAP	Reservoir capacity	map/dat/dam_cap_	kg
DAMID_	Reservoir ID	map/dat/dam_id__	-
DAMPRP	Primary purpose of reservoir	map/dat/dam_prp_	-
DAMSRF	Surface area of reservoir	map/dat/dam_srf_	m^2
DAMALC	Water demand allocation for each reservoir	map/out/dam_alc_	-

Table 6-7 Useful map data on reservoirs

File name	Note	Directory	Unit
DAMNUM	Number of reservoirs in a grid cell	map/dat/dam_num_	-
DAMYR_	Reservoir constructed yea	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

1. Carry out the processes up to Chapter 9, and finish the river flow calculations.[21]
2. Obtain the original data map-org-H06.tar.gz.[22]
3. Change directory to map/pre/.
4. Edit and execute prep_dam_H06.sh.

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

5. Change directory to map/bin/.
6. Edit and execute main_dam.sh.

This results in the creation of several reservoir information items. Firstly a file containing water demand allocation information for each reservoir (*.one) is created in directory map/out/dam_alc_/. Put another way, this file provides information on which

[21] To execute subsequent shell scripts without making any changes requires the calculation of monthly river flow from 1986 to 1995, followed by the use of the meanone command (or htmean command) to calculate the average monthly river flow for these 10 years.
[22] First download map-org-H06.tar.gz, and extract it in map/org.

upstream reservoir a certain cell's water demand has been allocated to. Next, reservoir governing files (*.one) are created in directories dam/out/dam_d2d_/ and dam/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. Lastly, 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 2 files shown in Table 6-8 are required to consider industrial and domestic water withdrawal. Use the process detailed below to create these files.

Table 6-8 Map data required to use the water withdrawal module

File name	Note	Directory	Unit
DEMIND	Industrial water demand	map/dat/dem_ind_	kg s^{-1}
DEMDOM	Domestic water demand	map/dat/dem_dom__	kg s^{-1}

1. Following the instruction in map/pre/prep_map_AQUASTAT.sh[17], download the original data from AQUASTAT.
2. Change directory to map/pre/.
3. Edit and execute prep_map_AQUASTAT.sh.

[17] The style of output seems frequently changed in AQUASTAT service. You need minor modification accordingly. Pay particular attention to the country including accent marks (e.g. Côte d'Ivoire)

・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 at all. It is a good idea if you put memorandum if you considerably changed the script. Settings A include 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 include Ouptut, Output Directory, Macro, and others. If you modify the code, scripts, you might need to change these parts for consistency. About Job, only experts should modify. All of the lines here are important for H08 simulation.

CHAPTER 7
Meteorological Data

H08 input data consists 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 (8 meteorological variables and abledo) required for H08 calculations. The variable abbreviations, names, units, and sign (direction of positive values) used here conform to the ALMA conventions Version 3.[24] 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 Second Global Soil Wetness Project (GSWP2; Dirmeyer et al., 2006) as a reference. Under GSWP2, 2 types of global meteorological data─B0 (Dirmeyer et al., 2006) and B1 (Hanasaki et al., 2008a)─were developed. We use the latter in this manual.[25]

Table 7-1 Meteorological variables

Variable name	Note	Directory	Unit
Albedo	Albedo	met/dat/Albedo__	-
PSurf	Surface pressure	met/dat/PSurf___	Pa
Rainf	Rainfall rate	met/dat/Rainf___	kg m^{-2} s^{-1}
Snowf	Snow fall rate	met/dat/Snowf___	kg m^{-2} s^{-1}
Wind	Wind speed	met/dat/Wind____	m s^{-1}
LWdown	Longwave downward radiation	met/dat/LWdown__	W m^{-2}
Qair	Specific humidity	met/dat/Qair____	kg kg^{-1}
SWdown	Shortwave downward radiation	met/dat/SWdown__	W m^{-2}
Tair	Air temperature	met/dat/Tair____	K

7.2 Preparing Meteorological Data

In this section, we explain how to prepare GSWP2 B1 meteorological and albedo data for use with H08. This basically involves converting the NetCDF data file distributed by GSWP2 into H08 Format using the following process:

[24] http://web.lmd.jussieu.fr/~polcher/ALMA/convention_3.html
[25] In general, GSWP2 meteorological data refers to the B0 data prepared as a standard for GSWP2 experiments, but B0 data harbors a number of problems that cannot be ignored from the global water balance perspective. These problems have been resolved in the B1 data body. For details, see Dirmeyer et al. (2006) and Hanasaki et al. (2008a).

1. Change directory to met/pre/.
2. Download GSWP2 meteorological data files from the file server, and save them at met/org/GSWP2B1[18]. See the instruction in prep_GSWP2B1.sh
3. Edit and execute prep_GSWP2B1.sh.[19]
4. Edit and execute prep_GSWP2B1_Albedo.sh[20]

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

・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 this data, which is 3-hourly. If you want to obtain Tokyo data (E139.5, N35.5), enter:

```
% cd met/dat/Tair____
% punchone lonlat ./GSW2B1b_.one3H 1986 1986 139.5 35.5
```

This results in the display of 3-hourly time series data. Comparing GSWP2B1 input data with observation data will help you to understand its features. Now convert this data into daily data by first converting the H08 Format 2D 3-hourly data into daily data, and then using the above procedure to display the data for just one location. The procedure is thus as follows:

```
% mon2yearone ./GSW2B1b_.one3H 1986 1986 ./GSW2B1b_.oneDY
% punchone lonlat ./GSW2B1b_.oneDY 1986 1986 139.5 35.5
```

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.

[18] For half degree global simulation, use WATCH Forcing Data (WFD; Weedon et al., 2011).
[19] For half degree global simulation, use prep_WFD.sh
[20] For half degree global simulation, use prep_WFD_Albedo.sh

7.3 Exercises

Perform the following tasks using the GSWP2 global meteorological data in met/dat.

1. Choose any land point in the world, and find its latitude and longitude.
2. Use the meteorological data in met/dat to create time series graphs for the point selected in 1.
3. Obtain observation data for the corresponding point and period, and overlay it on the time series graph created in 2. Don't forget that the GSWP2 data is displayed in Universal Time, Coordinated (UTC).
4. Compare GSWP2 data with observation data on a seasonal, daily, and 3-hourly basis.
5. [Difficult] Create a shell script for carrying out tasks 1 to 4.

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 2 points. Firstly, the force-restore method (Bhumralkar, 1975) has been incorporated for surface temperature. Secondly, subsurface runoff has been incorporated. Those wishing to know detailed formulae and theoretical background should refer to Hanasaki et al. (2008a).

Figure 8-1 is a schematic representation of H08's land surface process module. For convenience' sake, we will explain water balance (on the left of the diagram) and heat balance (right) separately. Firstly, 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. There are 2 state variables on water balance: soil moisture (SoilMoist) and snow water equivalent (SWE). Inputting precipitation (Prcp) causes evaporation (Evap), total runoff (Qtot), and other fluxes to be calculated based on land surface water balance and heat balance, and soil moisture and snow water equivalent 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. However, the force-restore method has been incorporated to calculate realistic average surface temperature (AvgSurfT) based on inputted meteorological data in increments shorter than one day (3−6 hour units). There are 2 state variables on heat balance: average surface temperature (AvgSurfT) and soil temperature (SoilTemp). Inputting downward shortwave radiation (SWdown), downward longwave radiation (LWdown), air temperature (Tair), specific humidity (Qair), 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, and the calculation of sensible heat flux (Qh), latent heat flux (Qle), ground heat flux (Qg), upward shortwave radiation (SWup), upward longwave radiation (LWup), and other fluxes; surface temperature and soil temperature are updated as a result.

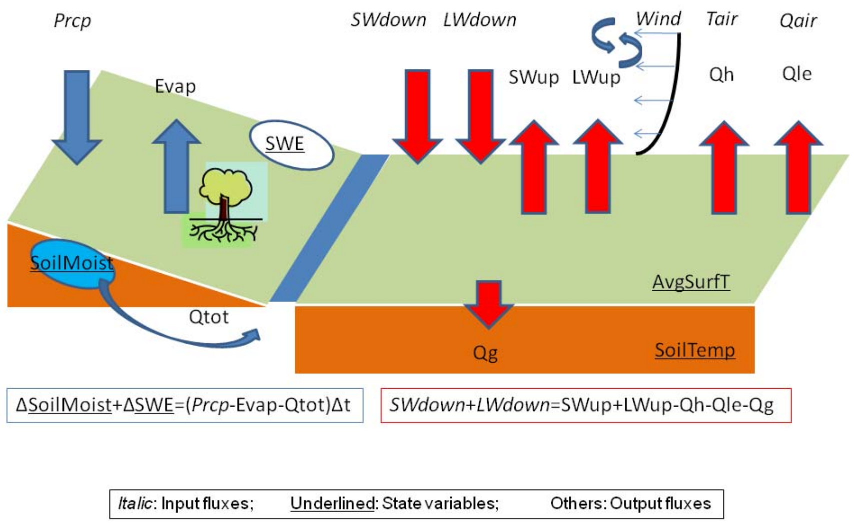

Figure 8-1 Schematic diagram of land surface process module

8.2 Land Surface Process Module Calculation Process

The land surface process module (lnd/bin/main_nomosaic.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 4 state variables.
5 Start the time loop (spinup mode).
   5.1 In the calculation period, set just the calculation start year to repeat.[27]
   5.2 Read the 8 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 spinup termination conditions. 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 with the previous spinup. If this condition is not satisfied, go back to 5.[28]

[27] Which means that, if the calculation period is 1986–1995, you repeat just 1986
[28] Note that depending on the combination of SPNRAT and SPNERR, spinup may continue endlessly.

8 Start the time loop (calculation mode).
  8.1 Set the calculation period as from start year to end year.
  8.2 Read the 8 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 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 Method 1: Preprocessing

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

1. Change directory to lnd/pre/.
2. 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.
3. Execute prep.sh. This results in the outputting of parameters in lnd/dat, and initial values for statevariables in lnd/ini.

Table 8-1 List of model parameters

Variable	Note	Unit
SOILDEPTH	Soil depth	m
FIELDCAP	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

Table 8-2List of files on state variables[29]。

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 Method 2: Calculation

With the completion of instructions in the previous section, you should have all of the files you need for execution of the land surface process module. Execute the land surface process module as follows:

1. Change directory to lnd/bin/.
2. Edit main.sh[21]. Set all shell variables shown in Tables 8-3 to 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.
3. Execute main.sh.[30] 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. Third directories corresponding to module output elements and conforming to the ALMA conventions Version 3 are created directly below lnd/out.

[29] A variable name prefixed by DIR signifies a directory, and variable name ending in INI signifies an initial value file. Variables with neither are output files.
[21] To apply H08 to globally at 0.5 degree spatial resolution, note that following points: (1) Set PRJ to WFD_, (2) set DBGL to smaller than 259200, (3) substitute GSW2B1b_ with WFD_____ for meteorological data, and (4) set the suffix of BALBEDO to MM (mean monthly, or fix year at 0000) which is originally set MO (monthly, year varies).
[30] 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 n01 that signifies array size. If you change 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
RPJ	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 I coordinate	-
SPNFLG	Flag of spinup. One for the condition spin up has been completed, and zero for not completed. If this variable is set at zero, model is executed in spin up mode, then normal calculation mode. If this is set at one, the model is executed in normal calculation model immediately right after 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 with 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	kg $m^{-2}s^{-1}$
SNOWF	Snowfall rate	kg $m^{-2}s^{-1}$
TAIR	Air temperature	K
QAIR	Specific humidity. Set either QAIR or RH. De 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 humidty 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 1986.

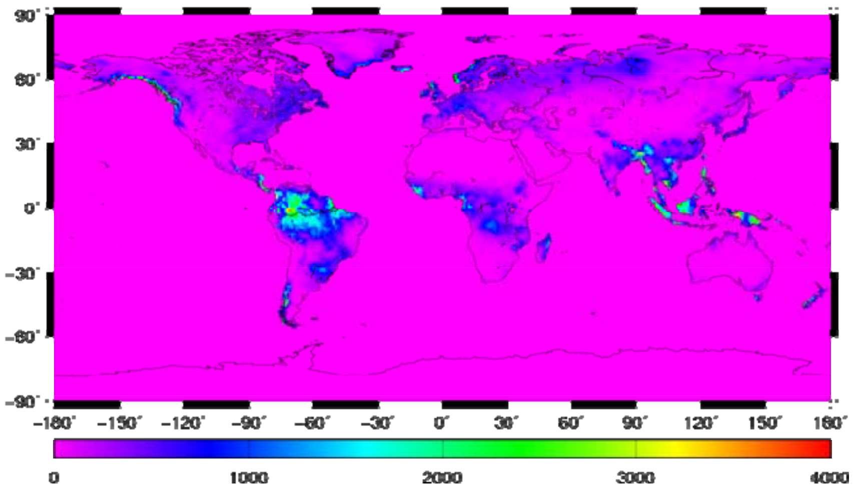

Figure 8-2 Mean annual runoff in 1986 [mm][31]

8.5 Land Surface Process Module Execution Method 3: Postprocessing

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

1. Change directory to cpl/pst/.
2. Edit and execute list_watbal.sh[22].

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. 9999.99 is a missing value that shows that the relevant file does not exist. list_watbal.sh is a program for checking water balance in yearly units, and as such, will not work properly unless rain and snow annual average files are present. Check for the files in met/dat/Rainf___ and met/dat/Snowf___, and if they have not been made, use httime (short form is mon2yearone) to create them first[23].

3. Edit and execute draw_all.sh.

[31] Created using the following procedure:

```
% cd lnd/out/Qtot____
% makecpt -T0/4000/1000 -Z > temp.cpt
% mulone GSW2LR__19860000.one 86400 temp.one
% mulone temp.one 365 temp.one
% one2eps temp.one temp.cpt temp.eps
% htconv temp.eps temp.png rot
```

[22] In case you analyze for the year 1987, set YEAR=1987, MON=00, YEARINI=1986, MONINI=12, YEAREND=1987, MONEND=12. See Figure 5-5 why we need to set like this.

[23] You need following procedure in case you run H08 with GSWP2 dataset

```
% cd met/dat/Rainf___
% mon2yearone ./GSW2B1b_.one3H 1986 1986 ./GSW2B1b_.oneDY
% cd met/dat/Snowf___
% mon2yearone ./GSW2B1b_.one3H 1986 1986 ./GSW2B1b_.oneDY
```

This results in the display of a time series graph showing water balance items for 1 location. This can be useful for the detailed analysis of calculation results for cells in which calculation errors have occurred.

・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 calculation, some kind of 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.

The 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 differences with 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 with the previous year's value is less than 5% in at least 95% of the cells.

The most ideal method is to combine these 2 methods. The period used for analysis by the GSWP2 project was 1986–1995, but 13 years' worth (1983–1995) of meteorological data was prepared. The recommended spinup method was to use the 1983 data to run repeated calculations and generate initial values, after which calculations were run up to 1995, and only the last 10 years 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 (Qtot × 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 calculation cell to the next downstream cell.

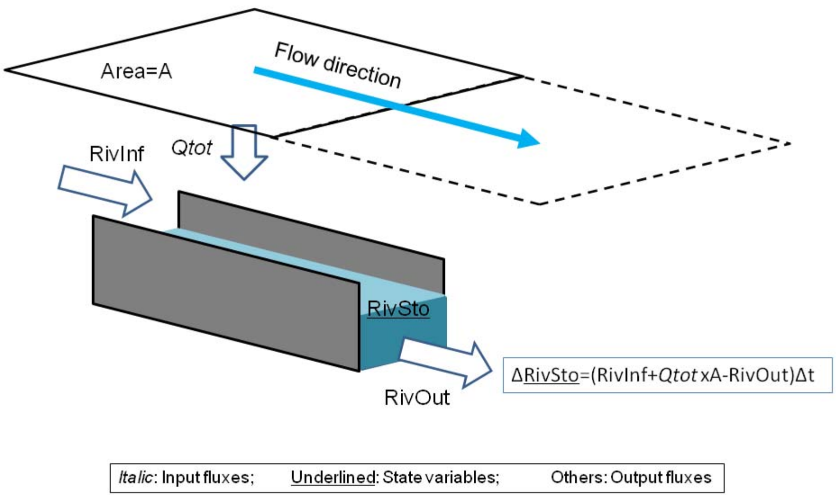

Figure 9-1 Schematic figure of river module

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

9.3 River Module Execution Method 1: Preprocessing

1. Change directory to riv/pre/.
2. Edit 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.
3. This results in the outputting of 2 parameter files for meander ratio and flow speed models 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
RIVSTO	River storage	kg

9.4 River Module Execution Method 2: Calculation

1. Change directory to riv/bin/.
2. Edit and execute main.sh. Set the shell variables shown in Tables 9-3 to 9-5.
3. 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 files on input hydrological variables

File	Note	Unit
QTOT	Total runoff	kg m^{-2} s^{-1}

Table 9-4 List of files on input map

File	Note	Unit
RIVSEQ	River sequence. The origin of river channel is 1, and 1 is added for every grid downstream. Zero 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 9-5List of output files

File	Note	Unit
RIVOUT	River discharge	kg s^{-1}

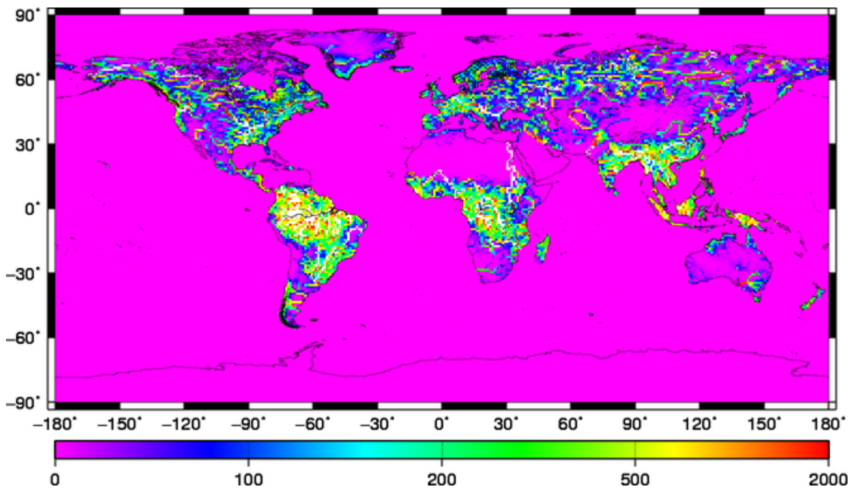

Figure 9-2 Annual river discharge in 1986 [m^3 s^{-1}]

9.5 River Module Execution Method 3: Postprocessing

1. Change directory to cpl/pst/.
2. 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).

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 a model for simulating hydrological cycles and another 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 (Tair) 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 place. The green line shows the length of the cropping period in days (i.e. days needed the accumulated temperature reaches 1500°C) for a crop planted each day from January 1 to December 31. The cropping period becomes shorter as the planting date approaches summer period, and longer as it approaches winter.

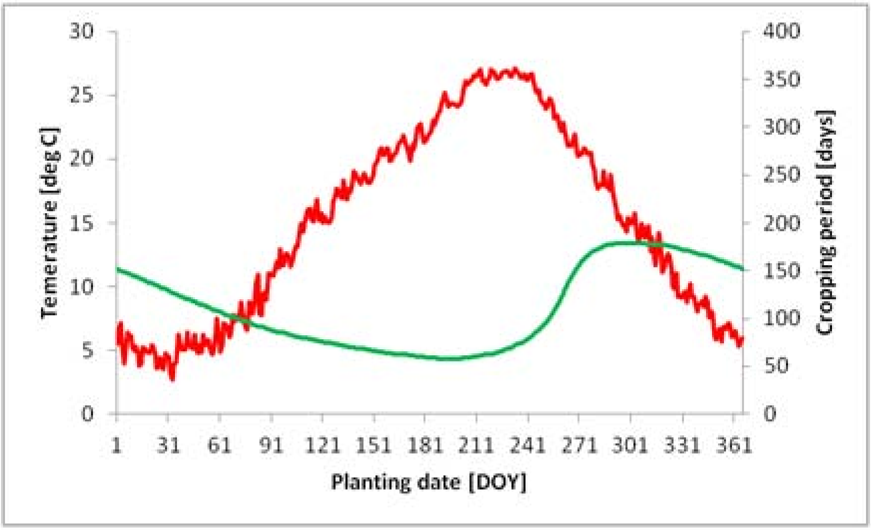

Figure 10-1 Daily average temperature [degree C] (red) and cropping period [day] (green) for an imaginary place.

In general, plants do not grow if the temperature is too low, and 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 adding up. The accumulated temperature in this case is known as effective accumulated temperature. SWIM too uses effective accumulated temperature. The green line of Figure 10-2 shows cropping period with base temperature of 10°C. Here it is assumed that the crop is killed if a day in the cropping period falls below 10°C.

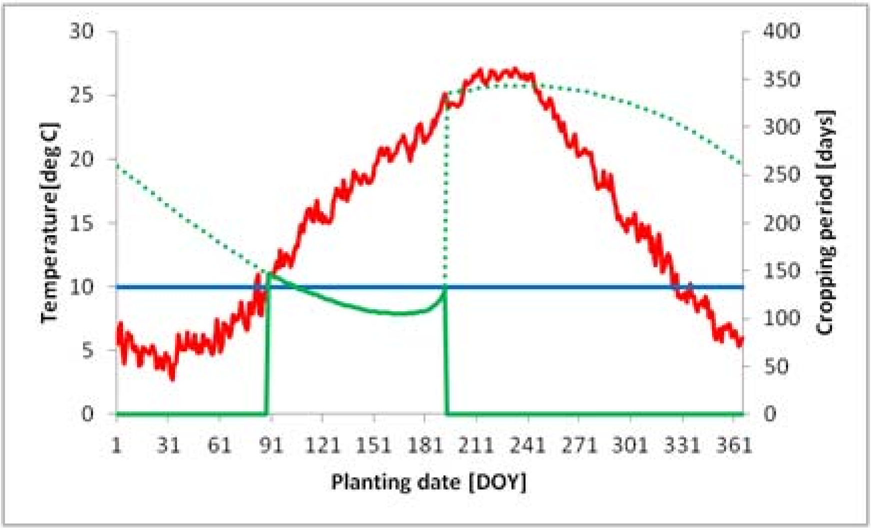

Figure 10-2 Daily average temperature [degree C] (red) and cropping period [day] (green) with effective accumulative temperature.

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).

$$BT = \sum \Delta B \qquad (10\text{-}1)$$

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

$$\Delta B = BE \times PAR \times REGF \qquad (10\text{-}2)$$

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 4 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:

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

Here, BAG is aboveground biomass, HVSTI is a parameter known as harvest index that 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.[32]WSF is water stress factor, and is expressed as follows:

$$WSF = \frac{SWU}{SWP} \times 100 \tag{10-4}$$

Here, SWU is actual evapotranspiration in the latter half of the cropping period,[33], 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). Firstly, the crop growth model calculates harvesting date and yield for a crop planted on every day from January 1 to December 31. If 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 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 provides the average over several years is 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 smoothes 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.

---

[32] 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 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.

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

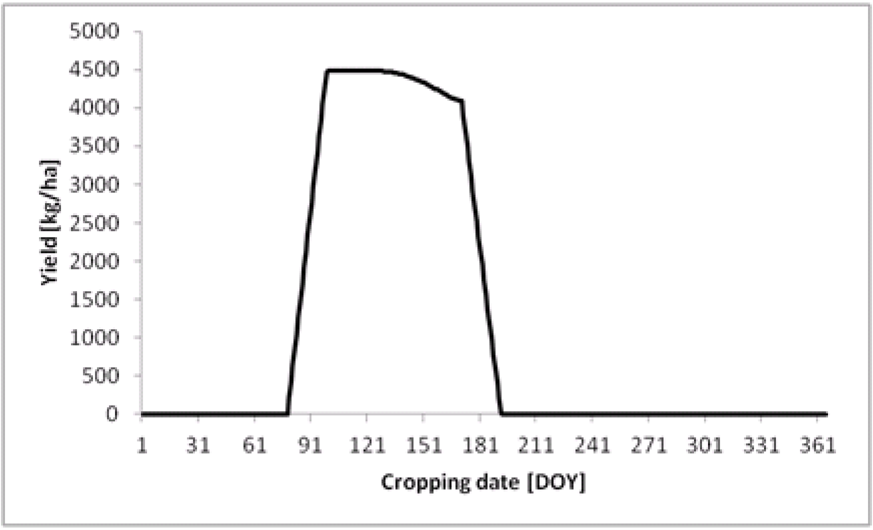

Figure 10-3 Cropping date and yield

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, 19×19=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 ID. Abbreviation ends with “g” indicates crop parameters are NOT available, and substituted by using 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 of 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 planting dates, harvesting dates, and cropping days that provide maximum yield.

11 Output results to a file.

10.3 Crop Growth Module Execution Method 1: Preprocessing

1. Obtain the original data crp-org-SWIM.tar.gz and extract them in crp/org.
2. Change directory to crp/pre/.
3. Edit and execute prep.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. The module creates 3-hourly to daily data by default, but errors will occur if 3-hourly data is not outputted with the land surface process module calculation. If the land surface process module outputted data is monthly data, you will need to recalculate it.

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 the latter half of cropping period	mm
SWP	Potential evaporation in the 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 Method 2: Calculation

1. Change directory to crp/bin/.
2. Edit and execute main.sh[24]. Set the shell variables of Tables 10-3 to 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 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 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_/GSW2__C_00000101.one file.

3. Change directory to crp/pst/.
4. Edit 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, firstly, of the start and end dates of the occupation periods of first crops to crp/out/ocu_ini_ and crp/out/ocu_end_ respectively. 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. 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.

5. Change directory to crp/bin/.
6. Edit and execute main.sh. Set JOBS=2nd. Ensure that ocu_ini_ and ocu_end_ located near the middle of the shell script are read in.

This results in second crop planting date, harvesting date, cropping days, yield, and dominant regulating factor being outputted respectively to crp/out/plt_2nd_, crp/out/hvs_2nd_, crp/out/crp_2nd_, crp/out/yld_2nd_, and crp/out/reg_2nd_.

---

[24] To run H08 for the Chao Phraya River at 5 minutes spatial resolution, edit main.f and set
parameter (n0lall=60*84)
parameter (n0llnd=1936)
and re-compile. The latter is the total number of land grid cells.

10.5 Crop Growth Module Execution Method 3: Postprocessing

1. Change directory to crp/pst/.
2. Edit and execute draw_crpyld_map.sh.
3. Cropping calendars and maps showing global distribution of yield will be outputted to the crp/fig/ directory.

Table 10-3 List of variables on simulation settings

Variable	Note	Unit
RPJ	Project name : (4 characters)	-
RUN	Run name (4 characters)	-
SUF	Suffix of H08 Format 2D (4 characters)	-
MAP	Type of map. (e.g. “.GSWP2” of flwdir.GSWP2.one)	-
YEAR	Year to simulate. 0000 for average of the period.	-
SECINT	Calculation interval	sec
NL	Total number of cells (e.g. 64800 (360x180) for 1degree by1degree 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 ID of Leff et al. (2004) and of SWIM	-
RAM2NAME	Conversion tables for the crop ID of Leff et al. (2004) and abbreviation shown in Table 10-1.	-
SWIM2RAM	Conversion tables for the crop ID of SWIM and of Leff et al. (2004)	-
CRPPAR	Parameter file of SWIM	-

Table 10-5 List of model parameters (original parameters of H08)[25]。

Variables	Note	Unit
DAYMAV	Days to calculate 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 die from cold	degC
HUNMAX	Maximum daily accumulation of temperature	degC
IHUNMAT	No more used. Set “1.0” any time.	-
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 die of cold　(“yes” to enable the process)	yes/no

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)	$kgm^{-2}s^{-1}$

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

Table 10-7 List of files on maps

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

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, 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 2 assumptions. The 1st is that reservoirs are operated in such a way as to reduce the size and frequency of yearly and seasonal fluctuations in flow. The 2nd 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 km^3, 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[m^3], when river flow is I_{mean}[m^3], is expressed as

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

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

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

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. Firstly, 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' [m^3/s]) is expressed as follows:

$$r' = k_{rls} \cdot i_{mean} \tag{11-3}$$

Here, i_{mean} is the average annual flow [m^3/s]. The provisional value for irrigation reservoir release (r' [m^3/s]) is expressed as follows.

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

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. 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. Σarea is the sum of the calculated cells downstream from the reservoirs.

Lastly, release (r [m^3/s]) is expressed as follows:

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

Here, i is inflow [m^3/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≈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 Method 1: Preprocessing

1. Change directory to riv/pst/.
2. Edit and execute calc_mean.sh
3. Edit and execute calc_flddro.sh.

This results in the use of river flow simulation to divide a year into storage and release periods. Based on this result, riv/out/fld_dro_ is created.

<u>You can skip the following procedures and proceed Chapter 12 unless you wish to develop or modify the reservoir operation module.</u>

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 validation. The module is usually used together with the land surface hydrology and river module as shown in Chapter 13.

4. Download dam-org-H06.tar.gz from the file server. Place it to dam/org and unzip.
5. Change directory to dam/pre/.
6. Execute prep_obsope.sh.

This results in the conversion of reservoir operation records contained in dam/org/H06 into H08 Format.

7. Download map-org-H06.tar.gz from the file server. Place it to map/org and unzip.

8. Execute list_obsope.sh. This results in the creation of the dam information file dam/dat/obsdat.txt that is referenced by dam/bin/main.sh.
9. 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.
10. If you execute draw_obsope.sh, you can plot reservoir operations.

11.4 Reservoir Operation Module Execution Method 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.

1. Change directory to dam/bin/.
2. Edit and execute main.sh.

11.5 Reservoir Operation Module Execution Method 3: Postprocessing

Because 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.

1. Change directory to dam/pst/.
2. Edit and execute draw_results.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) model[26] 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 in Table 12-1 based on monthly runoff height maximums (q_{max}) and minimums (q_{min}) This sets 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≤q<1	q_{env}=0
		1≤q	q_{env}=0.1q
Wet	10[mm month^{-1}] ≤q_{min} and 100[mm month^{-1}] ≤q_{max}		q_{env}=0.4q
Stable	1[mm month^{-1}] ≤q_{min} and q_{max} <100[mm month^{-1}]		q_{env}=0.1q
Variable	Other than above	0≤q<1	q_{env}=0
		1≤q<10	q_{env}=0.1q
		10≤q	q_{env}=0.4q

12.2 Environmental Water Module Calculation Process

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

1. Read in the arguments.
2. Read the catchment area file.
3. Read the monthly river flow file [kg s^{-1}], and use the catchment area data to convert the unit to [mm month^{-1}].
4. Calculate monthly river flow maximums and minimums.

26 But 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. Divide global land areas into the 4 categories according to Shirakawa (2005).
6. 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:

1. Change directory to riv/pst/.
2. Edit and execute calc_envout.sh.
3. Environmental flow is outputted to riv/out/env_out_.

12.5 Environmental Water Module Execution Method 3: Postprocessing

1. Change directory to cpl/pst/.
2. Edit and execute list_watbal.sh.
3. Environmental water totals will be displayed in the cpl/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; (3) write out output data. In the coupled model, subroutines of the 4 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.

Firstly, 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 index[27] $\theta$are related as follows.

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

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 1,000−1,500 km^3 yr^{-1}. Since global irrigated croplands cover a total area of about 2.50 $\times 10^6$ km^2, irrigation water demand per unit area is on average 400−600 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 2 subcells

---

[27] 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.

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 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 4 types of subcell.

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 (4—land, river, crop growth, human activity).
3 Read the map parameter files (4—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 calc_damdem subroutine).
  5.7 Calculate reservoir release (call calc_resope subroutine).
  5.8 Calculate land surface process (call 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 calc_humact subroutine).
  5.12 Output river, reservoir operation, medium-sized reservoir, and water withdrawal data to files.
  5.13 Calculate crop growth (call 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

1. Change directory to cpl/pre/.
2. Edit and execute prep.sh.

Table 13-1 List of files on state variables

File	Note	Unit
DAMSTO	Storage of large reservoirs	kg
PNDSTO	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 calculation globally at a resolution of 1°×1° of the ability to obtain the desired quantity of water at the desired time from rivers.

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

3. Change directory to cpl/pre, and execute prep_mean.sh. Then change directory again to cpl/bin.
4. Make sure map/bin/main_dam.sh (Chapter 6) has been executed.
5. Edit main.sh, and execute the LECD experiment. The features of LECD experiment settings are as follows:
   - Set RUN to "LECD"
   - Enable the reservoir operation model. It means set correct path to the variables DAMID_, DAMPRP, DAMCAP, DAMMON1ST, DAMALC, DAMRIVOUTFIX, and DAMDEMAGRFIX. Note that DAMSRF should be kept as "NO". Look at carefully the shell script: everything is already described. Only you need to do is add eight #.
   - Disable non-local and non-renewable blue water (virtual inexhaustible water sources): Set OPTNNB to "No". This results in 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 mosaic	-
ARAFRC	Areal fraction	-
OPTLNDUSE	Land use option (See Table 13-3)	-

Table 13-3 List of land use options for subcell

Option	Note
sci	Single crop irrigated
dci	Double crop irrigated
scr	Single crop rainfed
non	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 set to "H06" any time)	H06/M98/nokrls/nodem

OPTDAMWBC	Reservoir water balance calculation (Basically set to "NO" any time)	yes/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 daty 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 1st 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 size 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}
SUPAGRPND	Agricultural water withdrawal from medium size reservoir:	kg s^{-1}

SUPINDPND	Industrial water withdrawal from medium size reservoir:	$kg\ s^{-1}$
SUPDOMPND	Domestic water withdrawal from medium size 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 just water balance.

1. Change directory to cpl/pst/.
2. Edit and execute list_watbal.sh.[28]

[28] In the case of global calculation, river water inputs and outputs may sometimes fail to balance by 2–3 km^3 (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 (2010) at an extremely high level, and H08 was designed to be able to be applied to 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 means mainly examining the results of the model's output with respect to changes in meteorological data. This goes for H08 too. If users prepare the 8 items of meteorological data listed in Table 7-1 that shows a world in which global warming is taking place, along with any necessary map data, and supply H08 with this data, the impact of global warming can be assessed. In general, meteorological data that shows 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 perhaps one of the simplest climate scenario creation methods is 'shifting' and 'scaling' (Lehner et al., 2006). This method takes a time series of current climate and, by adding to or multiplying meteorological elements changed by global warming, creates the meteorological data of a world undergoing global warming. The changes in meteorological elements caused by global warming referred to here (such as the difference between average temperatures for 2071−2100 and 1961−1990, or a comparison of precipitation for the same years) are almost all generated by global warming numerical experiments using global climate models. These features are incorporated into H08, and so we will introduce them here.

1 Obtain the results of global warming numerical experiments using a global climate model. They can be obtained relatively easily using the global water resource model input/output data server.[29] The current standard method is to obtain 30 years' worth of future and current data for temperature, precipitation, and downward longwave radiation. See Table 14-1 for details.
2 Change directory to met/pre/, and execute prep_CMIP3_mean.sh to create 30-year averages respectively for future (A2 scenario) temperature, precipitation, and downward longwave radiation. Set parameters as, RUN=a21_, RUNOUT=a213, YEARMIN=2071, YEARMAX=2100, and PID (process ID). PID is shown in URL.

[29] http://h08.nies.go.jp (user: cmip5; password: CMIP5)

3 Again execute prep_CMIP3_mean.sh for the past (20C3M scenario). Set parameters as, RUN=201_, RUNOUT=201_, YEARMIN=1961, YEARMAX=1990, and PID.
4 Execute prep_CMIP3_delta.sh to calculate differences between current and future temperature and downward longwave radiation, and calculate rates of change for current and future precipitation. This results in outputting temperature difference in met/dat/Tair__DF, longwave difference in met/dat/LWdownDF, and precipitation change in met/dat/Prcp__RT.
5 Execute prep_RH.sh to calculate relative humidity.
6 Change directory to lnd/bin/ and edit "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 climate change simulation, humidity is expressed in relative humidity, therefore, QAIR will contain NO, and RH will contain file names.
7 Edit "Climate change input" and "Climate change output" sections of main.sh. Add temperature differences, precipitation rates of change, and downward longwave radiation differences respectively to TCOR, PCOR, and LCOR. Set TAIROUT, RAINFOUT, SNOWFOUT, and LWDOWNOUT. This will give you temperature, precipitation, and downward longwave radiation after shifting/scaling.
8 Set PRJ in main.sh. In this case, set "mm23" short for "MIROC 3.2 medres, A2 scenario, Period 3". Do not change YEARMIN and YEARMAX. The concept of shifting and scaling is add/multiply the difference to the simulation for the past. The output files will be named such as "mm23LR__19860101.one", and these are the simulation results for 2071-2100.
9 Execute main.sh. You can run riv/bin/main.sh, crp/bin/main.sh [30], and cpl/bin/main.sh[31] similarly.
10 Change directory to cpl/pst/, and edit and execute list_watbal.sh. Do not forget to change precipitation and snowfall to the values set with RAINFOUT and SNOWFOUT. Otherwise water balance does not close. Set RAINFOUT=../../lnd/out/Rainfout, SNOWFOUT=../../lnd/out/Snowfout, PRJMET=mm23, and RUNMET=LR__.

[30] Don't forget to execute crp/pre/prep.sh before running crp/bin/main.sh. For crp/bin/main.sh, what you need to edit are basically, PRJ, POTEVAP, EVAP, TCOR, and TAIROUT.

[31] For cpl/bin/main.sh, what you need to edit are basically, PRJ, PRJLR__, PRJ__C_, TCOR, PCOR, LCOR, TAIROUT, RAINFOUT, SNOWFOUT, LWDOWNOUT. If you wish to run fully coupled mode (RUN=LECD), execute map/bin/main_dam.sh with parameter PRJDIS=mm23.

Table 14-1 Data to retrieve

Options	Settings
Dataset	CMIP3
Model	MIROC3.2 medres (m32m)
Scenarios	A2 and 20C3M
Year:	2070-2099 for A2, 1961-1990 for 20C3M
Ensemble run	1
Variable	pr, tas, rlds
Temporal resolution	Monthly
Domain	-180, 180, -90, 90 for global (97, 102, 13, 20 for the Chao Phraya River)
Spatial resolution	360 x 180 for one degree global (720 x 360 for half degree global and 60 x 84 for five minutes Chao Phraya)
Meridional sequence	North to South
File type	Plain binary (little endian) with short name
Compress	Yes

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 (n0l=64800)
   ```

   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 this data. It enables easy plotting and calculation of H08 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. Firstly, the first character denotes the type of variable: i, r, c, n, p, and s respectively stand for variables integer, real, and character, and constants number, parameter, and string. 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. Though there are exceptions, as a rule, variables are expressed with 3 characters. For example, irg denotes irrigation, and ara, area. To sum up, r1irgara 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 which are given from arguments and namelist	Initialized by arguments and namelist
in (map)	Input data which are given from map files	Initialized by files
in (flux)	Input data which 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 (n0l). Exceptionally, the crop parameter arrays that appear in cpl/bin/main.f and crp/bin/main.f have dimension of crop types (e.g. r1icnum(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 your mind are as follows.

- crop variables in cpl/bin/main.f and crp/bin/main.f. They have dimensions of space and mosaic (e.g. r2huna(n0l,0:n0m)).
- crop parameters in cpl/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 cpl/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, r2swdown, r2potevap, r2evap, r2hvsdoy, 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 H08directories. 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.

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. 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. (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., T. Nitta (2009) UNIX/FORTRAN/Bourne Shell Script Self-study Text, 78 pp, National Institute for Environmental Studies, Tsukuba, Japan.(https://sites.google.com/site/h08model/english/manual)

Hanasaki, N., and T. Yamamoto (2010), H08 manual user’s edition, 75 pp, National Institute for Environmental Studies, Tsukuba, Japan.

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.
Lehner, B., P. Döll, J. Alcamo, T. Henrichs, and F. Kaspar (2006), Estimating the Impact of Global Change on Flood and Drought Risks in Europe: A Continental, Integrated Analysis, Climatic Change, 75, 273-299.
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.
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., S. Gomes, P. Viterbo, W. J. Shuttleworth, E. Blyth, H. Österle, J. C. Adam, N. Bellouin, O. Boucher, and M. Best (2011), Creation of the WATCH Forcing Data and Its Use to Assess Global and Regional Reference Crop Evaporation over Land during the Twentieth Century, J. Hydromet., 12(5), 823-848.