

H08 Manual

Analyzer's Edition

Yusuke Saito

Hanasaki Naota



Acknowledgements

This manual has been produced with support from the "Creation of Long-Range Vision for Global Sustainable Water Use" (Shinjiro Kanae, representative) project under the JST Strategic Basic Research Program (CREST) Innovative Technology and System for Sustainable Water Use program and "Guidance for Risk Assessment for Global Water Sustainability by Total Water Cycle / Water Resource Models" (Taikan Oki, representative), a Grant-in-Aid for Scientific Research (S). We would like to express our deep gratitude here.

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

Contents

Chapter 1 Introduction	1
Chapter 2 Software Installation for Manual	2
2.1 Introduction of Training Tool for Manual	2
2.2 ht_semi Directory Structure and Use.....	2
Chapter 3 Arguments.....	4
3.1 Arguments for Specifying Spatial Resolution	4
3.2 Arguments for Specifying File Format.....	5
3.3 Six Arguments for Specifying Time Resolution and Period.....	6
3.4 Arguments for Specifying Location	7
Chapter 4 Base and Drawing Commands	9
4.1 htcal (Base: Display Number of Days for Specified Year and Month).....	9
4.2 ht12x12y (Base: Creation of Coordinate System Conversion Files).....	9
4.3 htdraw (Drawing: Map Creation).....	10
4.4 htdrawts (Drawing: Creation of Line Graphs).....	13
4.5 htconv (Image Format Conversion)	15
Chapter 5 Binary File Processing Commands.....	17
5.1 htcreate (Create New File)	17
5.2 htformat (Format Conversion)	18
5.3 htarray (Change Sequence)	20
5.4 htmath (Four Arithmetic Operations)	22
5.5 htstat (Statistical Calculations).....	28
5.6 httime (Convert Time Resolution).....	30
5.7 htmean (Period Average Calculations)	30
5.8 htid	(X, Y, L, Lon and Lat Coordinate Conversions)..... 31
5.9 htedit (Edit Any Cell)	33
5.10 htpoint (Display Data for Any Location).....	34
5.11 htpointts (Display Period Data for Any Location).....	35
5.12 htmask (Mask Extraction)	37
5.13 htmaskrplc (Mask Replacement).....	41

5.14	htlinear (Linear Interpolation)	44
5.15	htlist2bin (Create H08 Format 2D from List)	49
5.16	htbin2list (Convert H08 Format 2D Binary File to List)	55
5.17	htcatchment (Display a Cell Upstream of Any Location)	59
5.18	htpercentile (Calculate Percentile Value)	62
5.19	htrank (Create Ranking Table for Data for Any Location)	63
Chapter 6 ASCII File Processing Commands		66
6.1	httimetxt (Convert Time Resolution)	66
6.2	htmeantxt (Period Average Calculations)	67
6.3	htstattxt (Statistical Calculations)	68
6.4	htcat (Combination of Table Format Files)	69
6.5	htcatts (Convert H08 Format 2D Files to ASCII)	71
6.6	htranktxt (Create Ranking Table)	72
6.7	htlsmtxt (Least Squares)	74
6.8	htmettxt (Statistical Calculations)	75
Chapter 7 netCDF File Processing Commands		77
7.1	htcreatenc (Create New File)	77
7.2	htputncts (Insert Data)	79
Appendix 1 Abbreviation method for Arguments		83
A.1	Abbreviating Arguments Specifying Spatial Resolution	83

Chapter 1

Introduction

One of the authors has worked on the development of the H08 global water resources model software for computer simulations of the global water cycle and global water resources problems for a long time. Computer simulations are an effective method for estimating past values that were not observed and future values that cannot be observed. However, there are many difficulties in accurately carrying out computer simulations, analyzing the results and tying them to research. Large volumes of data must be scrupulously analyzed, and when one becomes aware of problems, he must revert back to the input data and perform the analysis. This manual explains a program group, commonly known as H08 Analysis Tools, for analyzing input and output data for the H08 global water resources model.¹ Input and output data for the H08 water resources model is described according to rules known as H08 Format.² If H08 Analysis Tools is used, H08 input and output data can be converted to figures and tables and can be analyzed simply. In this manual, assignment of arguments and examples of execution will be described in detail for each of these using figures and tables. In addition, using `ht_semi`,³ which is an educational tool for this book, is convenient for carrying out trial execution of commands.

This manual is written with the assumption of an understanding of the content in all chapters of a UNIX self-study guide and up to Chapter 9 of the H08 Manual User's Edition. In addition, the difficulty of this text increases as one progresses through the pages; therefore, patiently executing commands one by one, starting with the first ones is recommended.

¹ For use of H08, please see the H08 Manual User's Edition.

² For H08 Format, please see Chapter 5 of the the H08 Manual User's Edition.

³ The method for downloading this tool is explained in Chapter 2.

Chapter 2

Software Installation for Using Manual

This chapter explains the installation of the training tool (ht_semi) required for using this manual, its directory structure and methods for using it.

2.1 Introduction of Training Tool for Manual

Introduction of the training tool (ht_semi) for using this manual will be described. First, download ht_semi_yyyymmdd.tar.gz from the H08 fileserver. Place it in any directory.

Next, expand the gzip compressed file and the tar archive. The user expands the gzip file by entering the first line of the following command statements and unpacks the tar archive by entering the second line.

```
% gunzip ht_semi_yyyymmdd.tar.gz
% tar xf ht_semi_yyyymmdd.tar
```

When the commands above are entered, a directory named ht_semi_yyyymmdd will be created. Since the name of this directory is long, change it to ht_semi as follows.

```
% mv ht_semi_yyyymmdd ht_semi
```

Next, move to the directory ht_semi.

```
% cd ht_semi
```

Edit and Execute start.sh (Actually, you don't need to edit this script at all, this time basically).

```
% sh start.sh
```

All of the ascii files in \${DIRSEMI}/data_bin have been converted into H08 Format 2D binary files. Now, you have finished installing the training tool.

2.2 ht_semi Directory Structure and Use

The ht_semi directory structure and use will be described. First of all, the directory structure for ht_semi is shown in Figure 2-1. The Chapter4 – Chapter7 directories are a directory group for the user to run H08 Analysis Tools trials. For example, when the user tries

out the command `htdraw`, it moves to `${DIRSEMI}/Chapter4/HTDRAW`.⁴ At this time, the files required when the user makes trial run are copied from `data_asc`, `data_bin`, and `cpt` for use (Figure 2-2). ASCII files and binary files used in the execution examples for the commands are in `data_asc` and `data_bin`, respectively. In addition, the CPT files used in drawing maps in the examples for command execution by the user are in `cpt`. Since the image files for the results of the execution of commands in the examples are in `data_fig`, the user can compare the results of trial runs of the examples with the image files in `data_fig`.

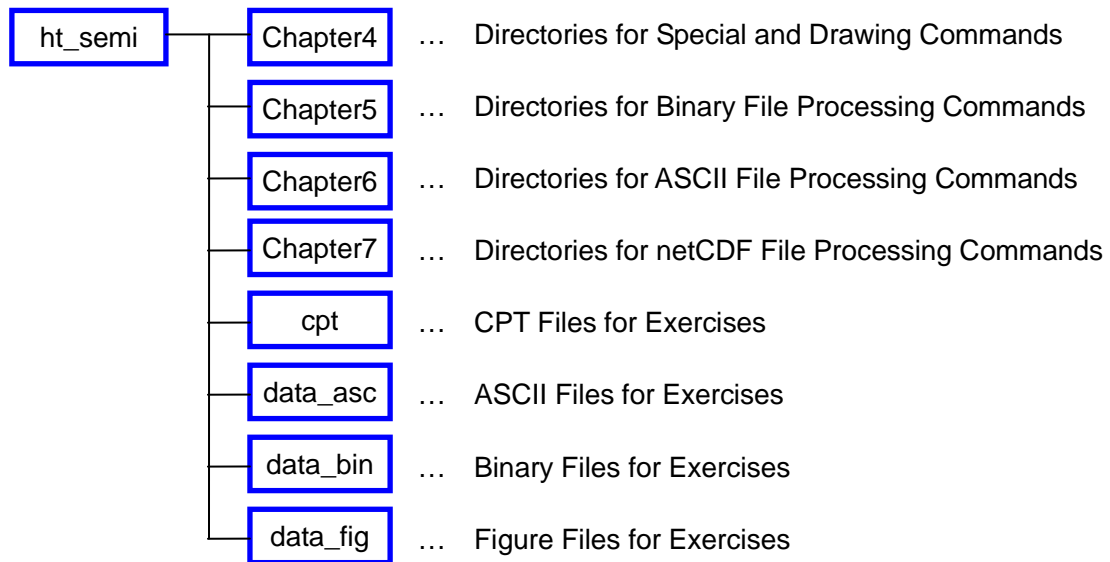


Figure 2-1 ht_semi Directory Structure

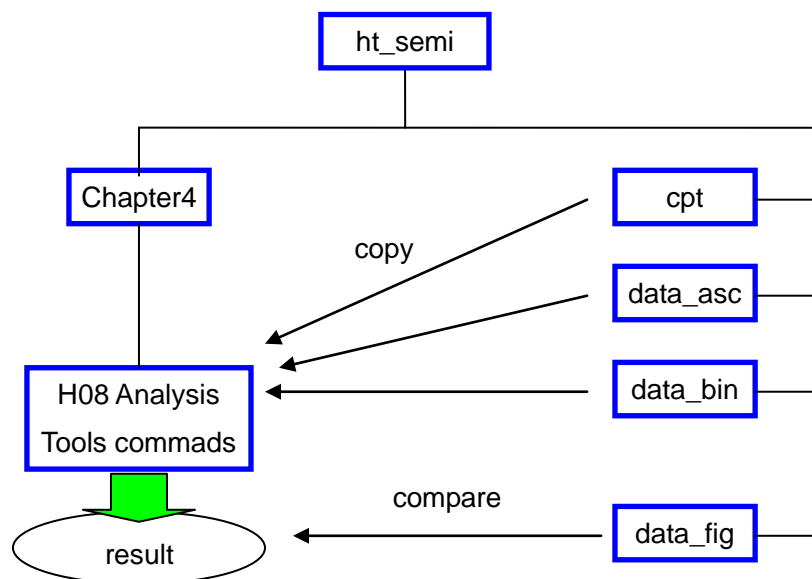


Figure 2-2 Flowchart for Manual Use

⁴ Not limited to this when no directory is specified.

Chapter 3

Arguments

In this chapter, the arguments assigned in H08 Analysis Tools will be described. The arguments for specifying spatial resolution, file format, period and location are frequently used; therefore, they should be understood thoroughly.

3.1 Arguments for Specifying Spatial Resolution

The nine arguments for specifying the spatial resolution⁵ of an H08 Format 2D binary file will be reviewed (Table 3-1). `n0l` is the total number of grid cells when the H08 Format 2D binary file is mapped. `n0x` and `n0y` are the number of cells along the longitudinal axis and the latitudinal axis, respectively. For example, when the entire world is divided at $1^\circ \times 1^\circ$, `n0x=360`, `n0y=180` and `n0l=64800` (360×180). There are two methods for representing any grid position, the L coordinate system and the XY coordinate system. The L coordinate system counts the cells from the upper left the lower right of the map, and any cell is expressed by the number it is in that order. The XY coordinate system sets the origin of the X-axis and Y-axis at the upper left of the map, and represents the position of any cell as an X-coordinate and Y-coordinate. `c0l2x` and `c0l2y` are ASCII files for X and Y coordinate conversions of L coordinates. H08 places these files in `${DIRH08}/map/dat/l2x_l2y_/. In addition, these files are created by the H08 Analysis Tools command htl2x12y. p0lonmin, space p0lonmaax, space p0latmin and p0latmax are the minimum and maximum values for longitude and latitude, respectively, and indicate what region of the world for the data in the H08 Format 2D binary file. For example, if any H08 Format 2D binary file includes global data, the arguments for this are, in order, -180, 180, -90 and 90.`

Table 3-1 Nine Arguments for Specifying Spatial Resolution

Argument name in Fortran code	Description	Argument name in shell
<code>n0l</code>	Number of grid cells (e.g.: with $1^\circ \times 1^\circ$, 64800; with $0.5^\circ \times 0.5^\circ$, 259200)	L
<code>n0x</code>	Number of divisions along longitudinal axis (e.g.: with $1^\circ \times 1^\circ$, 360; with $0.5^\circ \times 0.5^\circ$, 720)	XY
<code>n0y</code>	Number of divisions along latitudinal axis (e.g.: with $1^\circ \times 1^\circ$, 180; with $0.5^\circ \times 0.5^\circ$, 360)	

⁵ See: 5.3 Expressing space in H08 Manual User's Edition

c0l2x	L coordinate-X-coordinate conversion file (e.g.: with 1°x1°, $\{\text{DIRH08}\}/\text{map}/\text{dat}/\text{l2x_l2y_l2x.one.txt}$ 0.5°x0.5° $\{\text{DIRH08}\}/\text{map}/\text{dat}/\text{l2x_l2y_l2x.hlf.txt}$)	L2X
c0l2y	L coordinate-Y-coordinate conversion file (e.g.: with 1°x1°, $\{\text{DIRH08}\}/\text{map}/\text{dat}/\text{l2x_l2y_l2y.one.txt}$ 0.5°x0.5° $\{\text{DIRH08}\}/\text{map}/\text{dat}/\text{l2x_l2y_l2y.hlf.txt}$)	L2Y
p0lonmin	Minimum value for longitude	LONMIN
p0lonmax	Maximum value for longitude	LONMAX
p0latmin	Minimum value for latitude	LATMIN
p0latmax	Maximum value for latitude	LATMAX

3.2 Arguments for Specifying File Format

Table 3-2 shows the arguments for specifying file format. H08 Analysis Tools handles a variety of file formats including binary files, ASCII files and image files. The user must give H08 Analysis Tools the correct path for files. In this manual, input filenames are denoted in a normal font and output filenames in bold. Please note that file names that begin with numbers and minus signs will cause H08 Analysis Tools to malfunction.

Table 3-2 Arguments for Specifying File Format⁶

Argument name	Description
r0dat	Any numerical value
c0cpt	CPT filename for GMT
c0eps	Vector image filename (.eps extension)
c0ras	Raster image filename (.gif, .jpg, .tif or .png extension)
c0bin	H08 Format 2D binary file name (simple)
c0bints	H08 Format 2D binary file name (time series) Displayed in $\{\text{DIR}\}/\{\text{PRJ}\}\{\text{RUN}\}\{\text{SUF}\}\{\text{IDX}\}$ format
c0msk	H08 Format 2D binary file name used as mask file
c0asc	Data list ASCII file name
c0lst	Character and data 2 column ASCII file name
c0cod	Character and code 2 column ASCII file name
c0xyz	Longitude (lon), latitude (lat) and data (data) 3 column ASCII file name
c0ascts	H08 Format 1D ASCII file name given as

⁶ See: 5.5 H08 Format 2D and 5.6 H08 Format 1D in Chapter 5 of H08 Manual User's Edition

\${DIR}\${PRJ}\${RUN}\${SUF}\${IDX}	
c0nc	netCDF format file name
<p>\$DIR: specifies directory.</p> <p>\$PRJ: specifies four character project name.</p> <p>\$RUN: specifies four character test name.</p> <p>\$SUF: specifies three character extension (.one, .hlf, etc.) for H08 Format 2D.</p> <p>\$IDX: any of YR (year), MO (month), DY (day) or 3H (three hourly selections) is selected.</p>	

3.3 Six Arguments for Specifying Time Resolution and Period

The six arguments for specifying time resolution and period are given in Table 3-3. H08 Analysis Tools handles six types of time resolution data⁷, annual average, monthly average, daily average and six, three and one hour averages. c0idx is a two character symbol, and represents these time resolutions. yearmin and yearmax are arguments specifying the start year and the end year, respectively, for a period. For example, when the user specifies an H08 Format 2D binary file for three years from 1986 to 1988, these arguments are 1986 and 1988. yearout specifies the year for which the results of period average calculations are to be output. This argument is normally 0000. i0year and i0mon represent year and month, respectively.

Table 3-3 Arguments for Specifying Time Resolution and Period

Argument name	Description
c0idx	Any of YR (year), MO (month), DY (day) or 3H (three hourly selections)
	Specifies the time resolution of preceding file (c0ascts)
yearmin	Start year for period
yearmax	End year for period
yearout	Year for output as period average (normally 0000)
i0year	Input year
i0mon	Input month

⁷ See 5.4 Expressing Time in Chapter 5 of H08 Manual User's Edition

3.4 Arguments for Specifying Location

The arguments⁸ specifying location are given in Table 3-4. When the user uses locationH08 Analysis Tools, there are three methods for specifying any location. The first is the method of specifying any location using the L coordinate system. The second is the method of specifying any location using the XY coordinate system. The third is the method of specifying any location using longitude and latitude.

Table 3-4 Arguments for Specifying Grid

Method for specifying and argument name		Description
l	i0l	Grid specified using L coordinate system. i0l: number in L coordinate system.
xy	i0x i0y	Grid specified using XY coordinate system. i0x: X coordinate i0y: Y coordinate
lonlat	r0lon r0lat	Grid specified using longitude and latitude. r0lon: longitude r0lat: latitude

To make the correspondences among the coordinate systems easy to understand here, the conversion equations for the coordinate systems used in H08 are shown in the following. (Reread this section when the correspondences become difficult to understand.)

Equation for calculating X-coordinate from longitude⁹

$$i0x = n \text{int} \left\{ \frac{r0lon - p0lon \text{ min}}{p0lon \text{ max} - p0lon \text{ min}} \times \text{real}(n0x) + 0.5 \right\}$$

Equation for calculating Y-coordinate from latitude¹⁰

$$i0y = n \text{int} \left\{ \frac{p0lat \text{ min} - r0lat}{p0lat \text{ max} - p0lat \text{ min}} \times \text{real}(n0y) + 0.5 \right\}$$

Equation for calculating longitude from X-coordinate¹¹ (East)

$$r0lon = p0lon \text{ min} + \frac{p0lon \text{ max} - p0lon \text{ min}}{\text{real}(n0x)} \times \{ \text{real}(i0x) - 1.0 \}$$

Equation for calculating latitude from Y-coordinate¹² (South)

⁸ See: 5.3 Expressing space in Chapter 5 of H08 Manual User's Edition

⁹ See: .../H08/lib/iget0x.f

¹⁰ See: .../H08/lib/iget0y.f

¹¹ See: .../H08/lib/rgetlat.f

$$r0lat = p0lat \min + \frac{p0lat \max - p0lat \min}{real(n0y)} \times \{real(i0y) - 1.0\}$$

An example will be given using Figure 3-1. This figure is an example of the world divided into a 4×4 grid with the value 2 in the green cell. The horizontal axis on the upper side of the figure and the vertical axis on the left side are the XY coordinates, and the horizontal axis on the lower side of the figure and the vertical axis on the right side are longitude and latitude coordinates. Here, the method for expressing the position of the green cell will be considered. The position in the L coordinate system is (7), and in the XY coordinate system, it is (3, 2). With longitude and latitude, this same cell is expressed by any value for longitude of $0 \leq lon < 90$ and any value for latitude of $0 \leq lat < 45$. Therefore, even if the user has the values for longitude and latitude offset from the center values (45° , 22.5°) for the grid cells, values linearly interpolated from surrounding values do not fall inside the cell and the value (2) unique to that grid cell always falls inside.

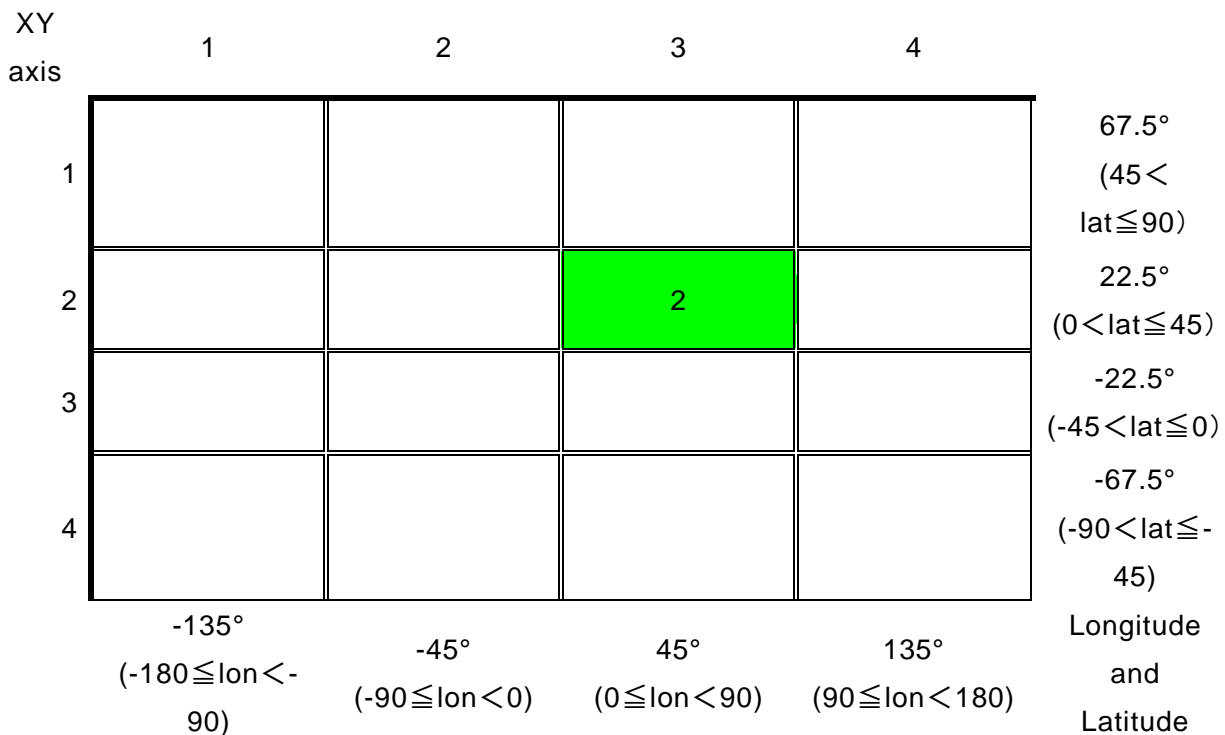


Figure 3-1 Simplified Diagram of Coordinate Conversion Concepts

¹² See: .../H08/lib/rgetlat.f

Chapter 4

Base and Drawing Commands

In this chapter, the base and the drawing commands for H08 will be described. Among the base commands, there are commands for returning the number of days in the month for a specified year and month and commands for creating ASCII files for coordinate system conversions. Among the drawing commands, there are commands for outputting EPS format maps from H08 Format 2D binary files, commands for creating hydrographs from flow rate data and commands for converting EPS files to ones with other extensions.

4.1 htcal (Base: Display Number of Days for Specified Year and Month)

This command displays the number of days of the month for a year and month given by the user.

4.1.1 Order of arguments

```
% htcal i0year i0mon
```

4.1.2 Example of execution

The number of days for February 1988 will be found.

```
% htcal 1988 2
29
```

4.2 ht12xl2y (Base: Creation of Coordinate System Conversion Files)

This command creates a special ASCII file (coordinate system conversion file) for converting H08 Format 2D binary files from the L coordinate system to X coordinate system and from the L coordinate system to the Y coordinate system. These two coordinate system conversion files are used in almost all cases when H08 Analysis Tools is used. Users should create these ASCII files according to the spatial resolution.

4.2.1 Order of arguments

```
% ht12xl2y n0l n0x n0y c012x c012y
```

4.2.2 Example of execution

First move to the directory for the htl2xl2y command.

```
% cd ${DIRSEMI}/Chapter4/HTL2XL2Y
```

Create a coordinate system conversion file for the spatial resolution of $1^\circ \times 1^\circ$.

```
% htl2xl2y 64800 360 180 12x.one.txt 12y.one.txt
```

Create a coordinate system conversion file for the spatial resolution of $0.5^\circ \times 0.5^\circ$.

```
% htl2xl2y 259200 720 360 12x.hlf.txt 12y.hlf.txt
```

4.3 htdraw (Drawing: Map Creation)

This command reads an H08 Format 2D binary file and creates an EPS file map according to the CPT file.

4.3.1 Order of arguments

```
% htdraw n0l n0x n0y c0l2x c0l2y p0lonmon p0lonmax  
p0latmin p0latmax c0bin c0cpt c0eps [TITLE1] [TITLE2] [OPT]
```

TITLE1: any character string, but spaces cannot be used.

TITLE2: any character string, but spaces cannot be used.

OPT : Select any one of whitesea, blacksea, bluesea, graysea, whiteland and blackland.

4.3.2 Example of execution

Use the command htdraw to create an EPS file from an H08 Format 2D binary file ($\${DIRSEMI}/data_bin/Indmsk.GSWP2.one$).

Move to the directory for the htdraw command.

```
% cd ${DIRH08}/ht_semi/Chapter4/HTDRAW
```

First, copy the H08 Format 2D binary file ($\${DIRSEMI}/data_bin/Indmsk.GSWP2.one$) for the land and sea map to the current directory.

```
% cp ../../data_bin/lndmsk.GSWP2.one .
```

Next, prepare the CPT file. Since the CPT file for the land and sea map is in `$(DIRH08)/Ht_semi/cpt`, copy it.

```
% cp ../../cpt/lndmsk.cpt .
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the `htl2x12y` command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`.

```
% htl2x12y 64800 360 180 12x.one.txt 12y.one.txt
```

Next, plot the H08 Format 2D binary file for the land and sea map using the `htdraw` command. In the following command statement, the user specifies the spatial resolution using the first nine arguments. After those arguments, the user continues by specifying the H08 Format 2D binary file, CPT file and output EPS file. Finally, the user specifies `TITLE1` as `landmask`.

```
% htdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
        lndmsk.GSWP2.one lndmsk.cpt lndmsk.eps landmask
```

Using the command `display`, display `lndmsk.eps`. Also, check whether the same image file as the original in Table 4-1 was obtained.

```
% display lndmsk.eps
```

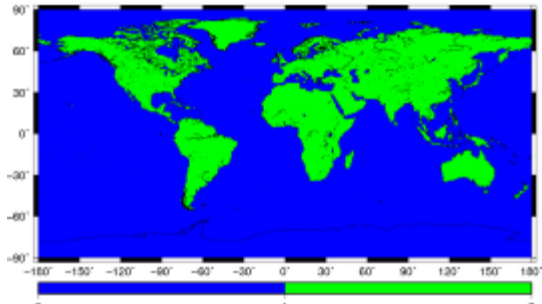
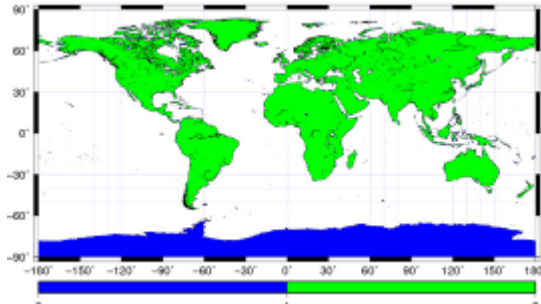
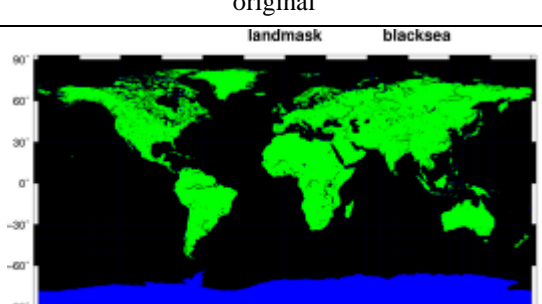
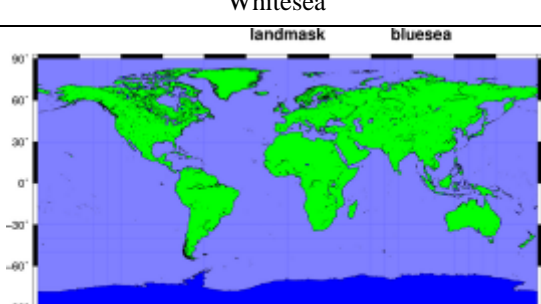
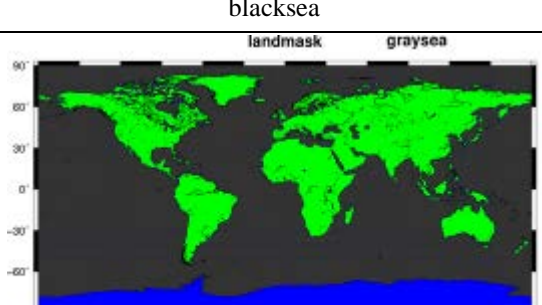
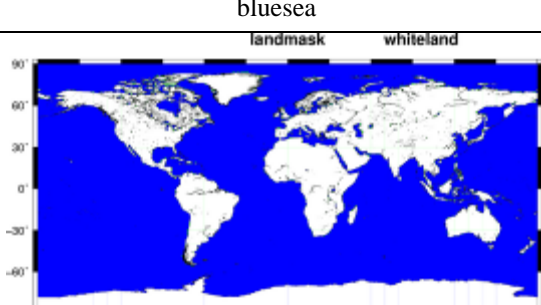
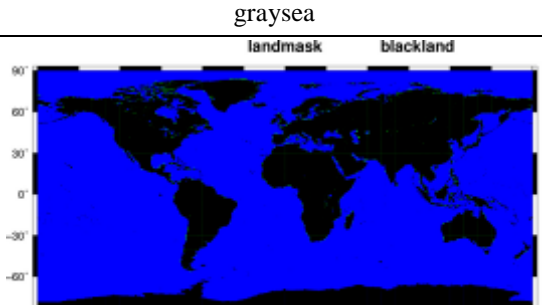
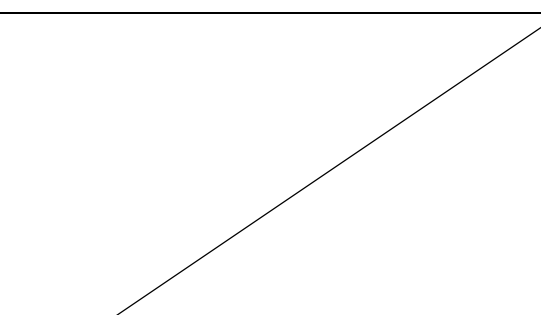
Next, plot the H08 Format 2D binary file for the land and sea map by specifying the `whitesea` option. Specify `landmask` for `TITLE1`, `whitesea` for `TITLE2` and `whitesea` for `OPT`.

```
% htdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
        lndmsk.GSWP2.one lndmsk.cpt lndmsk.eps landmask whitesea whitesea
```

Check whether the same image file as `whitesea` in Table 4-1 was obtained.

The EPS files created when the user uses other options are given in Table 4-1.

Table 4-1 Image Files When Options Are Used

<p>landmask</p> 	<p>landmask whitesea</p> 
<p>original</p> <p>landmask blacksea</p> 	<p>Whitesea</p> <p>landmask bluesea</p> 
<p>blacksea</p> <p>landmask graysea</p> 	<p>bluesea</p> <p>landmask whiteland</p> 
<p>graysea</p> <p>landmask blackland</p> 	<p>whiteland</p> 
<p>blackland</p>	

4.4 htdrawts (Drawing: Creation of Line Graphs)

This command reads an H08 Format 1D ASCII file and creates an EPS file line graph.

4.4.1 Order of arguments

```
% htdrawts c0ascts c0eps RFLAG JFLAG BFLAG TITLE [column]
```

RFLAG: specifies drawing region (range of X-axis and Y-axis).

(-*Rxmin/xmax/ymin/ymax*)

JFLAG: specifies projection method. (-*Jcode num/num*)

BFLAG: specifies scale interval for X-axis and Y-axis.

(-*Banumfnumgnum:Xname:/ anumfnumgnum:Yname:news*)

TITLE: specifies any graph title.

column: specifies a data column displayed in bold.

* RFLAG, JFLAG, BFLAG and TITLE have the same format as GMT command¹³ options and are arguments that specified the format of the graph.

4.4.2 Example of execution

Create a temperature line graph using the htdrawts command.

Move to the directory for the htdrawts command.

```
% cd ${DIRSEMI}/Chapter3/HTDRAWTS
```

First, copy the monthly average temperature ASCII files (tokyo.txt, sapporo.txt) for Tokyo (139.5° East, 35.5° North) and Sapporo (141.5° East, 43.5° North) in 1986 from \${DIRSEMI}/data_asc. Likewise, copy the ASCII file (tokyo_sapporo.txt) that has the monthly average temperature data for the two locations of Tokyo and Sapporo.

```
% cp ../../data_asc/tokyo.txt .
% cp ../../data_asc/sapporo.txt .
% cp ../../data_asc/tokyo_sapporo.txt .
```

Next, create a temperature line graph EPS file for Tokyo in 1986 using the htdrawts command on tokyo.txt.

¹³ See: <http://gmt.soest.hawaii.edu/>

```
% htdrawts tokyo.txt tokyo.eps
-R1/12/260/300 -JX10/15 -Balgl:Month:/a10g10:Tair:neWS Tair_1986_tokyo
```

A more specific description of the RFLAG, JFLAG and BFLAG in the command statement above will be given in the following.

-R1/12/270/300 (RFLAG)

-R specifies the region for the graph. In this example, the X coordinate is specified as 1 – 12 and the Y coordinate as 270 – 300.

-JX10/10 (JFLAG)

-J specifies the size for the graph. The X that is directly after J and specifies the projection method, means linear. Also, in this example, for which the units are cm, the size of the graph is specified as 10 cm×10 cm.

-Balgl:Month:/a10g10:Tair:neWS (BFLAG)

-B specifies the format of the axes for the graph. The meaning of a is annotation (numerical value), f frame (small graduations) and g grid (graduations). The numerical values directly following the letters for these specify the number of divisions for them. In the example, f is not specified. The title of the X-axis is Month and the title of the Y-axis is specified as Tair. news is an abbreviation for north, east, west and south. The user specifies the position of the graph title by making any of n, e, w and s uppercase. In this example, the titles are on the X-axis and Y-axis, so this is neWS.

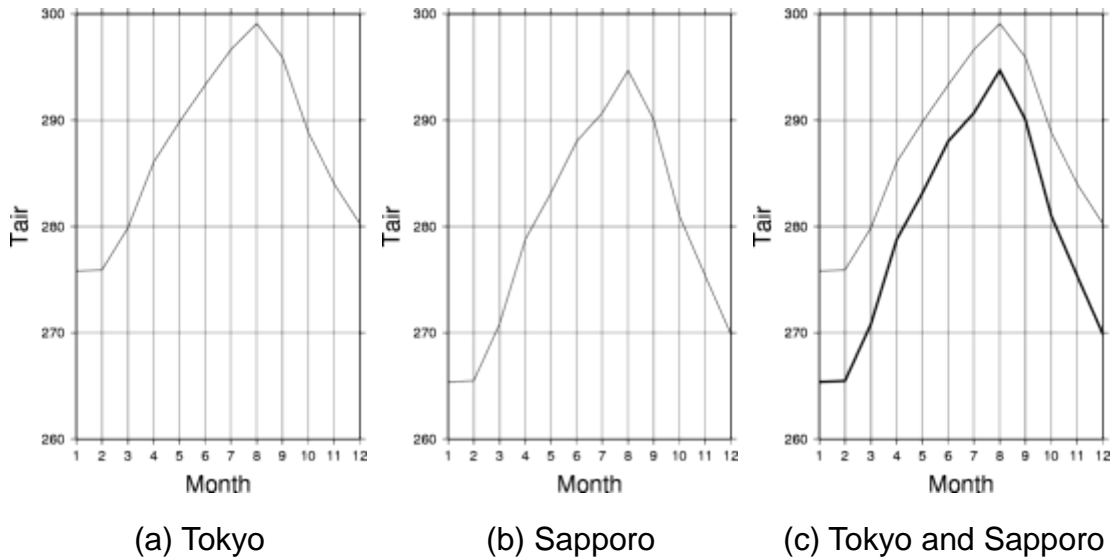
The EPS file created by the example above is shown in Figure 4-1 (a). In addition, the line graph created by executing the htdrawts command on sapporo.txt in a similar manner as in the example above is shown in Figure 4-1 (b).

Next, the H08 Format 1D ASCII file (tokyo_sapporp.txt), which has two data columns, will be graphed. This file gives the temperatures for Tokyo in the first column and the temperatures for Sapporo in the second column. The data column to be displayed in bold in the graph will be specified. In this example, (temperature data for Sapporo) is to be in bold, so column is 2.

```
% htdrawts tokyo_sapporo.txt tokyo_sapporo.eps
-R1/12/260/300 -JX10/15 -Balgl:Month:/a10g10:Tair:neWS Tair_1986 2
```

The EPS file created by the example above is shown in Figure 4-1 (c).

Looking at Figure 4-1 (c), it can be seen that the average monthly temperature in Tokyo is 5 K or higher than that for Sapporo, and in the period from December through March, there is a maximum temperature difference of approximately 10 K. It can also be seen that the temperature variations in the two locations are substantially the same.



(a) Tokyo (b) Sapporo (c) Tokyo and Sapporo
Figure 4-1 Average Monthly Temperature Graph for 1986

4.5 htconv (Image Format Conversion)

This command converts the image format from EPS to any of GIF, JPG, TIF and PNG. This command can also rotate an image 90°.

4.5.1 Order of arguments

```
% htconv c0eps c0ras OPT
```

OPT: option

rot: rotate 90[°]

hr: make high-resolution

rothr: 90[°] rotation + make high-resolution

4.5.2 Example of execution

The EPS file that was output by the htdraw command in the example will be converted into another image file format. In the following example, the EPS file is converted to a PNG file and rotated 90[°].

```
% htconv lndmsk.eps lndmsk.png rot
```


Chapter 5

Binary File Processing Commands

In this chapter, the commands for the H08 Analysis Tools binary file processing will be described. Among the commands for the binary file processing, there are commands for calculations among H08 Format 2D binary files. There are also commands for changing sequences in H08 Format 2D binary files. There are also commands for creating H08 Format 1D ASCII files from H08 Format 2D binary files. These commands are extremely useful and are a command group that forms the nucleus of H08 Analysis Tools.

5.1 htcreate (Create New File)

This command creates a new H08 Format 2D binary file.

5.1.1 Order of arguments

```
% htcreate n01 r0dat c0bin
```

5.1.2 Example of execution

A new H08 Format 2D binary file will be created using the htcreate command. First move to the directory for the htcreate command.

```
% cd ${DIRSEMI}/Chapter5/HTCREATE
```

An H08 Format 2D binary file with a spatial resolution of $1^\circ \times 1^\circ$ filled by 1 will be created. The number of cells is $360 \times 180 = 64800$. In the following example, the output file name is input1.one.

```
% htcreate 64800 1 input1.one
```

An H08 Format 2D binary file with a spatial resolution of $0.5^\circ \times 0.5^\circ$ filled by 100 will be created. The number of cells is $360/0.5 \times 180/0.5 = 259200$. In the following example, the output file name is input100.hlf.

```
% htcreate 259200 100 input100.hlf
```

5.2 htformat (Format Conversion)

This command converts the format of an H08 Format 2D binary file. For example, this command can convert an H08 Format 2D binary file to an ASCII file that lists data. This command can also convert in the opposite direction, an ASCII file that lists data to an H08 Format 2D binary file.

5.2.1 Order of arguments

```
% htformat n0l n0x n0y c0l2x c0l2y p0lonmon p0lonmax
           p0latmin p0latmax OPTION1 OPTION2 IN OUT
```

OPTION1: select any of binary, bytswp, asciiu, ascii3 and xls for the input file option

OPTION2: select any of binary, bytswp, asciiu, ascii3 and xls for the output file option

binary: when an H08 Format 2D binary file is used

bytswp: H08 Format 2D binary file with a different endian¹⁴

asciiu: ASCII file listing only data¹⁵

ascii3: ASCII file with three columns, longitude, latitude and data

IN: input file name

OUT: output file name

5.2.2 Example of execution

The format of the input1.one file created in the execution example for the htcreate command will be converted. Move to the directory for htformat.

```
% cd ${DIRSEMI}/Chapter5/HTFORMAT
```

Next, copy input1.one to the directory for the htcreate command.

```
% cp ../HTCREATE/input1.one .
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the ht2x12y command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is .one.txt.¹⁶

¹⁴ bytswp is only enabled when the following installation is in place.

When Intel Fortran Compiler is used, the environment variable "F_UFMTENDIAN" is "big;little:25-26."

When gfortran is used, the environment variable "GFORTTRAN_CONVERT_UNIT" is

"big_endian;little_endian:25-26."

¹⁵ In H08, asciiu and ascii3 were originally the argument names for c0asc and c0xyz, respectively. When H08 was first developed, asciiu and ascii3 were used, but in drafting this manual, the argument names were changed for these to maintain uniformity in argument names. These argument names should also be changed for the htformat command, but that would require extensive modifications of program statements, and the changes have not been made at the present stage.

¹⁶ When the user converts the file format of input100.hlf, which has a spatial resolution of $0.5^\circ \times 0.5^\circ$, the coordinate conversion ASCII file should be created as follows.

```
% ht12x12y 64800 360 180 12x.one.txt 12y.one.txt
```

Since the file to be used has been prepared, use the `htformat` command to convert `input1.one` to an ASCII file (`c0asc`) that lists only the data.

```
% htformat 64800 360 180 12x.one.txt 12y.one.txt
          -180 180 -90 90 binary asciiu input1.one input1.one.txt
```

Next, convert `input1.one` to an ASCII file (`c0xyz`) showing three columns, longitude, latitude and data.

```
% htformat 64800 360 180 12x.one.txt 12y.one.txt
          -180 180 -90 90 binary ascii3 input1.one input1.one.xyz
```

Convert `input1.one.txt` (`c0asc`) to an H08 Format 2D binary file.

```
% htformat 64800 360 180 12x.one.txt 12y.one.txt
          -180 180 -90 90 asciiu binary input1.one.txt input1_2.one
```

Convert `input1.one.xyz` (`c0xyz`) to an H08 Format 2D binary file.

```
% htformat 64800 360 180 12x.one.txt 12y.one.txt
          -180 180 -90 90 ascii3 binary input1.one.xyz input1_3.one
```

The content of `input1.one.xyz` and `input100.hlf.xyz` is shown in Table 5-1.

Table 5-1 Content of Longitude, Latitude and Data Format ASCII Files

x	y	z	x	y	z
-179.500	89.5000	1.000000	-179.750	89.7500	100.0000
-178.500	89.5000	1.000000	-179.250	89.7500	100.0000
-177.500	89.5000	1.000000	-178.750	89.7500	100.0000
-176.500	89.5000	1.000000	-178.250	89.7500	100.0000
-175.500	89.5000	1.000000	-177.750	89.7500	100.0000
-174.500	89.5000	1.000000	-177.250	89.7500	100.0000
↓	↓	↓	↓	↓	↓
-1.500	89.5000	1.000000	-0.500	89.7500	100.0000
-0.500	89.5000	1.000000	0.000	89.7500	100.0000
0.500	89.5000	1.000000	0.500	89.7500	100.0000

```
% ht12x12y 259200 720 360 12x.hlf.txt 12y.hlf.txt
```

↓	↓	↓	↓	↓	↓	↓
177.500	89.5000	1.000000	178.750	89.7500	100.0000	
178.500	89.5000	1.000000	179.250	89.7500	100.0000	
179.500	89.5000	1.000000	179.750	89.7500	100.0000	
-----			-----			
-179.500	88.5000	1.000000	-179.750	89.2500	100.0000	
-178.500	88.5000	1.000000	-178.250	89.2500	100.0000	
-177.500	88.5000	1.000000	-177.750	89.2500	100.0000	
↓	↓	↓	↓	↓	↓	
↓	↓	↓	↓	↓	↓	
		1.000000			100.0000	
Spatial resolution of 1°x1°			Spatial resolution of 0.5°x0.5°			

5.3 htarray (Change Sequence)

This command changes the sequence in an H08 Format 2D binary file. For example, this command can give an H08 Format 2D binary file centered at 180° East a center at 0° East by shifting along the East-West direction. This command can also invert an H08 Format 2D binary file that lies across the equatorial axis.

5.3.1 Order of arguments

```
% htarray n0l n0x n0y c0l2x c0l2y OPTION c0bin c0bin
```

OPTION: select either upsidedown or shift.

upsidedown: invert North and South centered on X-axis.

shift: shift 180° East-West.

5.3.2 Example of execution

Use the htarray command on a land and sea map (.../H08/map/dat/lnd_msk_/lndmsk.GSWP2.one) to change the sequence. Move to the directory for the htarray command.

```
% cd ${DIRSEMI}/Chapter5/HTARRAY
```

First, copy the H08 Format 2D binary file (\${DIRSEMI}/data_bin/lndmsk.GSWP2.one) for the land and sea map to the current directory.

```
% cp ../../data_bin/lndmsk.GSWP2.one .
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the `htl2x12y` command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`.

```
% htl2x12y 64800 360 180 12x.one.txt 12y.one.txt
```

Specify the `upsidedown` option, and create a file in which the land and sea map has been inverted North-South across the equatorial axis.

```
% htarray 64800 360 180 12x.one.txt 12y.one.txt
                    upsidedown lndmsk.GSWP2.one ud_lndmsk.one
```

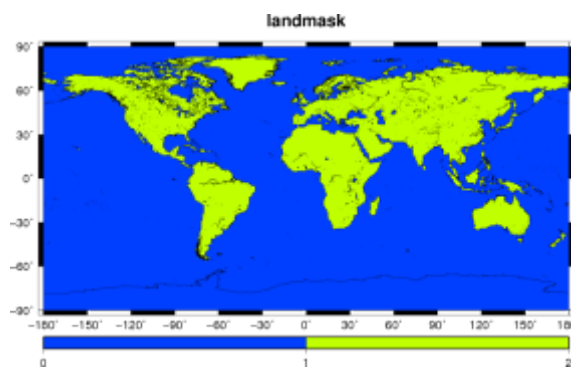
To check `ud_lndmsk.one`, which has been created by the command statement above, prepare a CPT file. Since the CPT file for the land and sea map is in `${DIRSEMI}/cpt`, copy it. Then create an image using the `htdraw` and `htconv` commands.

```
% cp ../../cpt/lndmsk.cpt .
```

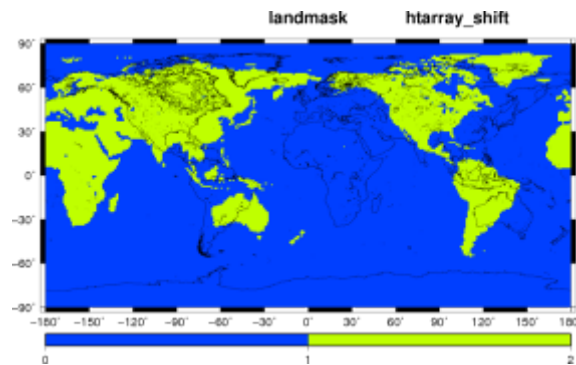
```
% htdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                    ud_lndmsk.one lndmsk.cpt ud_lndmsk.eps landmask htarray_upsidedown
```

```
% htconv ud_lndmsk.eps ud_lndmsk.png rot
```

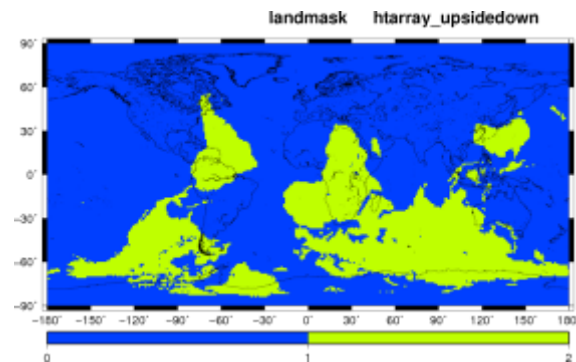
The image that has been created is shown in Figure 5-1 (c). Figure 5-1 (a) is the original `lndmsk` file, and Figure 5-1 (b) is the image file that has been shifted 180° East-West.



(a) Land and sea map



(b) Land and sea map with 180° shift East-West



(c) Land and sea map inverted North-South centered on the equatorial axis

Figure 5-1 Example of Results of htarray Command

5.4 htmath (Four Arithmetic Operations)

This command carries out the four arithmetic operations for an H08 Format 2D binary file. This command can perform the four arithmetic operations between H08 Format 2D binary files and between an H08 Format 2D binary file and any numerical value. For example, this command is convenient for the user to change the units of the data in an H08 Format 2D binary file.

5.4.1 Order of arguments

```
% htmath n01 OPTION c0bin IN c0bin
```

OPTION: select any of add, sub, mul and div.

add: find sum.

sub: find difference.

mul: find product.

div: find quotient.

IN: r0dat or c0bin

r0dat: when performing the four arithmetic functions with an H08 Format 2D binary file and numerical value

r0bin: when performing the four arithmetic functions between H08 Format 2D binary files

5.4.2 Example of execution

The units of an H08 Format 2D binary file will be converted the `htmath` command. Move to the directory for the `htmath` command.

```
% cd ${DIRSEMI}/Chapter5/HTMATH
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the `htl2x12y` command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`.

```
% htl2x12y 64800 360 180 12x.one.txt 12y.one.txt
```

sub Option

Use the `htmath` command with the `sub` option to convert the units in the temperature file from absolute temperature [K] to Celsius [$^\circ\text{C}$]. First, prepare the temperature file. Since the 1986 average temperature H08 Format 2D binary file (units: K) is in `${DIRSEMI}/data_bin/Tair____`, copy it.

```
% cp ../../data_bin/Tair____/GSW2B1b_19860000.one .
```

Change the name of the `GSW2B1b_19860000.one` file.

```
% mv GSW2B1b_19860000.one Tair_1986.one
```

To convert from Kelvin to Celsius, use the `htmath` command to subtract 273.15 from the Kelvin value.

```
% htmath 64800 sub Tair_1986.one 273.15 TairC_1986.one
```

To check that `TairC_1986.one` has been correctly created, use `htdraw` for drawing. First, prepare the CPT file. Since the CPT file for temperature (Celsius) is in `${DIRH08}/Ht_semi/cpt`, copy it.

```
% cp ../../cpt/TairC.cpt .
```

Draw `TairC_1986.one` using the `htdraw` command.

```
% hdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
      TairC_1986.one TairC.cpt TairC_1986.eps Tair_1986 Celsius whitesea
```

Finally, convert TairC_1986.eps to a PNG file using the htconv command.

```
htconv TairC_1986.eps TairC_1986.png rot
```

mul Option

This time, use the htmath command with the mul option to convert the units in the precipitation file from [mm/s] (= kg/m²s) to [mm/year]. First, prepare the precipitation file. Since the 1986 average precipitation H08 Format 2D binary file (units: mm/s) is in \${DIRSEMI}/data_bin/Rainf____, copy it.

```
% cp ../../data_bin/Rainf____/GSW2B1b_19860000.one .
```

Change the name of the GSW2B1b_19860000.one file.

```
% mv GSW2B1b_19860000.one Rainf_1986.one
```

Next, to convert the units of the Rainf_1986.one data from [mm/s] to [mm/year], use the htmath command to multiply by 31536000 (=60×60×24×365) [s].

```
% htmath 64800 mul Rainf_1986.one 31536000 Rainf_1986_year.one
```

To check that Rainf_1986_year.one has been correctly created, use htdraw for drawing. First, prepare the CPT file. Since the CPT file for average annual precipitation is in \${DIRSEMI}/cpt, copy it.

```
% cp ../../cpt/rain.cpt .
```

Create an EPS file from Rainf_1986_year.one using the htdraw command.

```
% htdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
      Rainf_1986_year.one rain.cpt Rainf_1986_year.eps
      Rainf_1986 [mm/year] whitesea
```

Finally, convert Rainf_1986_year.eps to a PNG file using the htconv command.

```
htconv Rainf_1986_year.eps Rainf_1986_year.png rot
```

div (1) Option

This time, convert the units of the surface area file for the grid cells from [m²] to [km²]. First,

prepare the grid surface area file. Since the grid surface area H08 Format 2D binary file (units: m^2) is in $\{\text{DIRSEMI}\}/\text{data_bin}$, copy it.

```
% cp ../../data_bin/grdara.one .
```

Next, use the `htmath` command to convert the units of `grdara.one`.

```
% htmath 64800 div grdara.one 1000000 grdara_km2.one
```

To check that `grdara_km2.one` has been correctly created, use `htdraw` for drawing. First, prepare the CPT file. Since the CPT file for surface area (units: km^2) is in $\{\text{DIRSEMI}\}/\text{cpt}$, copy it.

```
% cp ../../cpt/ grdara_km2.cpt .
```

Create an EPS file from `grdara_km2.one` using the `htdraw` command.

```
% hdraw 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90
      grdara_km2.one grdara_km2.cpt grdara_km2.eps Grid_Area [km2]
```

Finally, convert `grdara_km2.eps` to a PNG file using the `htconv` command.

```
% htconv grdara_km2.eps grdara_km2.png rot
```

div (2) Option

This time, convert the units of the river flow rate file¹⁷ from $[\text{kg}/\text{s}]$ to $[\text{mm}/\text{year}]$. First, prepare the river flow rate file. Since the 1986 river flow rate H08 Format 2D binary file (units: kg/s) is in $\{\text{DIRSEMI}\}/\text{data_bin}$, copy it. Likewise, also copy the basin area H08 Format 2D binary file.

```
% cp ../../data_bin/rivout_1986.one .
```

```
% cp ../../data_bin/rivara.GSWP2.one .
```

Next divide the river flow rate (`rivout_1986.one`) H08 Format 2D binary file by the basin area (`rivara.GSWP2.one`) H08 Format 2D binary file and convert the units from $[\text{kg}/\text{s}]$ to $[\text{kg}/\text{m}^2\text{s}]$ (= $[\text{mm}/\text{s}]$) using the `htmath` command.

```
% htmath 64800 div rivout_1986.one rivara.GSWP2.one rivout_1986_mm_s.one
```

¹⁷ If the H08 Manual User's Edition is executed through Chapter 9, the user can calculate this file himself.

Also convert the units from [mm/s] to [mm/year] as in the example of the mul option using the htmath command.

```
% htmath 64800 mul rivout_1986_mm_s.one 31536000 rivout_1986_mm_year.one
```

Create an EPS file from rivout_1986_mm_year.one using the htdraw command. Moreover, since the CPT file is in `$(DIRSEMI)/cpt`, copy it.

```
% cp ../../cpt/RiverFlow.cpt .
```

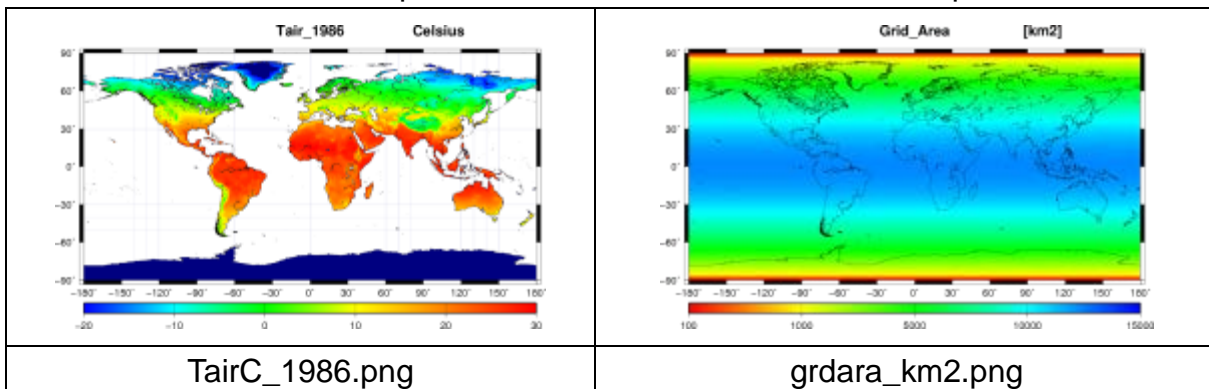
```
% hdraw 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90
    rivout_1986_mm_year.one RiverFlow.cpt rivout_1986_mm_year.eps
                                River_Flow_1986 [mm/year]
```

Finally, convert rivout_1986_mm_year.eps to a PNG file using the htconv command.

```
htconv rivout_1986_mm_year.eps rivout_1986_mm_year.png rot
```

Table 5-2 shows TairC_1986.png and grdara_km2.png. Also, Rainf_1986_year.png and rivout_1986_mm_year.png are shown in Figure 5-2 and Figure 5-3, respectively.

Table 5-2 Output Results for htmath Execution Examples



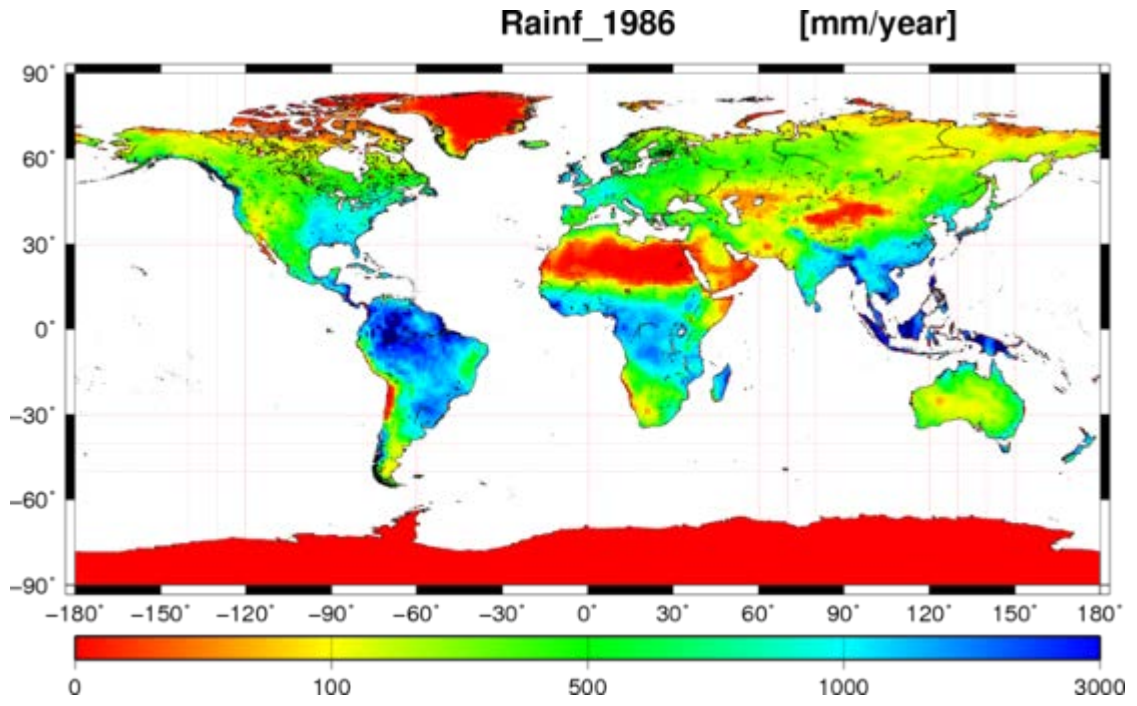


Figure 5-2 Precipitation for 1986

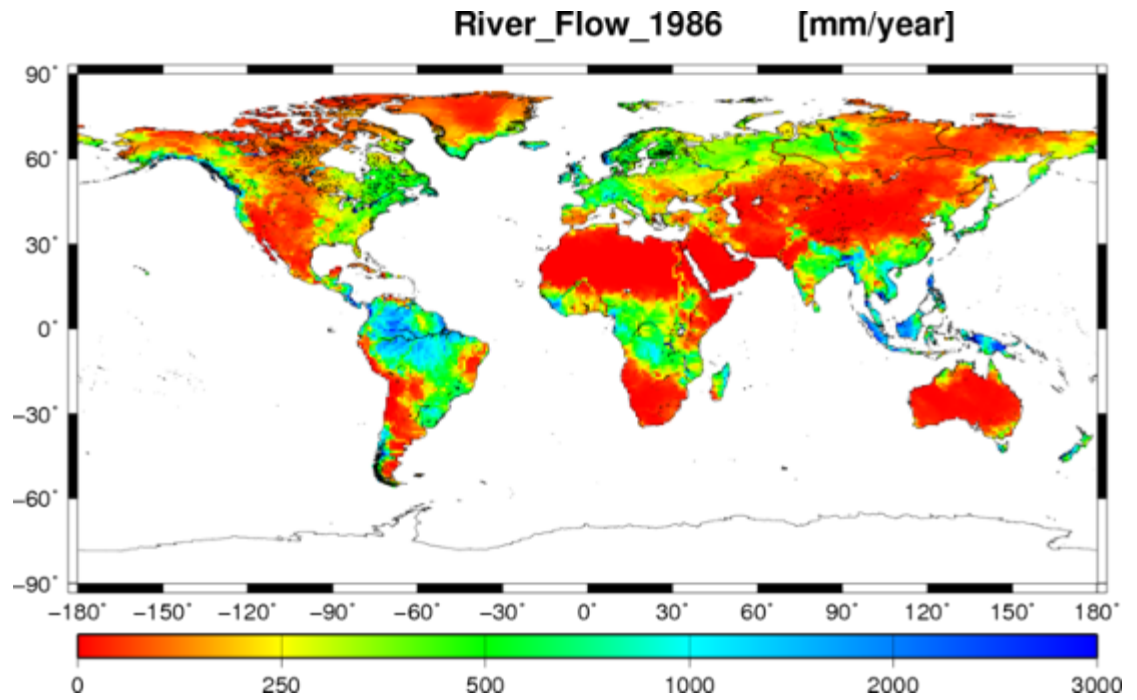


Figure 5-3 River Flow Rate for 1986

5.5 htstat (Statistical Calculations)

This command calculates statistics (maximum value, minimum value, sum and average) for an H08 Format 2D binary file.

5.5.1 Order of arguments

```
% htstat n0l n0x n0y c0l2x c0l2y p0lonmin p0lonmax
           p0latmin p0latmax OPTION c0bin
```

OPTION: select any of sum, ave, max and min.

sum: calculates the sum of data other than missing values¹⁸ in a given H08 Format 2D binary file.

ave: calculates the average of data other than missing values in a given H08 Format 2D binary file.

max: shows the maximum value in data other than missing values in a given H08 Format 2D binary file.

min: shows the minimum value in data other than missing values in a given H08 Format 2D binary file.

5.5.2 Example of execution

Statistics for the land and sea map and temperature data H08 Format 2D binary files will be found using the htstat command. Execution examples for the sum and ave options use the land and sea map H08 Format 2D binary file (`${DIRSEMI}/data_bin/lndmsk.GSWP2.one`). Also, execution examples for the max and min options use the temperature data H08 Format 2D binary file (`${DIRSEMI}/data_bin/Tair____/GSW2B1b_19860000.one`).

First move to the directory for the htstat command. Then copy the two files above.

```
% cd ${DIRSEMI}/Chapter5/HTSTAT
```

```
% cp ../../data_bin/lndmsk.GSWP2.one .
% cp ../../data_bin/Tair____/GSW2B1b_19860000.one .
```

Change the name of the GSW2B1b_19860000.one file to Tair_1986.one.

```
% mv GSW2B1b_19860000.one Tair_1986.one
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the htl2xl2y command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is .one.txt.

¹⁸ Missing values are set to 1.0×10^{-20} .

```
% ht12x12y 64800 360 180 12x.one.txt 12y.one.txt
```

sum

Find the sum for lndmsk.GSWP2.one. This file has 1 for land and 0 for sea. Therefore when the user uses the hstat command with the sum option, the total number of land grid cells is output. In the output, the numerical value on the left side is the result of the calculation, and the numerical value on right side is the number of grid cells targeted in the calculation.

```
% htstat 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                                                    sum lndmsk.GSWP2.one
15238.00          64800
```

ave

Find the average for lndmsk.GSWP2.one. This file has 1 for land and 0 for sea, but the hstat command cannot handle 0 as a missing value. Therefore, when the user uses the hstat command with the ave option, the calculation results of 15238/64800 are output. In the output, the numerical value on the left side is the result of the calculation, and the numerical value on right side is the number of grid cells targeted in the calculation.

```
% htstat 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                                                    ave lndmsk.GSWP2.one
0.2351543          64800
```

max

Find the maximum value in the 1986 average temperature file. Using the hstat command with the max option, find the maximum value in Tair_1986.one. The output is, starting with the numerical value on the left side, maximum value, L coordinate of cells showing maximum value, X coordinate, Y coordinate, longitude and latitude.

```
% htstat 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                                                    max Tair_1986.one
3.03474213E+02    27581    221    77    40.5000    13.5000
```

min

Using the hstat command with the min option, find the minimum value in Tair_1986.one. The output is, starting with the numerical value on the left side, minimum value, L coordinate of cells showing minimum value, X coordinate, Y coordinate, longitude and latitude.

```

% htstat 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                                                    min Tair_1986.one
0.00000000E+00          64800          360          180          179.5000          -89.5000

```

5.6 httime (Convert Time Resolution)

This command converts the time resolution in an H08 Format 2D binary file in the direction of lower resolution. For example, it is used when the user wants to create annual average data from monthly average data for certain year.

5.6.1 Order of arguments

```

% httime n0l c0bints i0yearmin i0yearmax c0bints

```

5.6.2 Example of execution

In this example, a 1986 annual average temperature H08 Format 2D binary file is created using the 1986 monthly average temperature H08 Format 2D binary file `${DIRSEMI}/data_bin/Tair____/GSW2B1b_1986??00.one`.

First move to the directory for the httime command.

```

% cd ${DIRSEMI}/Chapter5/HTTIME

```

Using the httime command, create an annual average temperature file from the monthly average temperature file for 1986.

```

% httime 64800 ../../data_bin/Tair____/GSW2B1b_.oneMO 1986 1986
                                                    ./GSW2B1b_.oneYR

```

With the above, the annual average temperature file (`GSW2B1b_19860000.one`) is created.

5.7 htmean (Period Average Calculations)

This command calculates the period averages for an H08 Format 2D binary file. For example, this command can calculate the 10 year average precipitation for each month from 1986 to 1995. This command is convenient when the user wants to know trends in climate and other data for a fixed period of time.

5.7.1 Order of arguments

```
% htmean n0l c0bints i0yearmin i0yearmax i0yearout
```

5.7.2 Example of execution

In this example, the three-year average temperature from 1986 to 1988 will be found. First move to the directory for the htmean command.

```
% cd ${DIRSEMI}/Chapter5/HTMEAN
```

Next, create the three-year average H08 Format 2D binary file for annual average temperature using the htmean command.

```
% htmean 64800 ../../data_bin/Tair____/GSW2B1b_.oneYR 1986 1988 0000
```

Check the three-year average temperature H08 Format 2D binary file created by the command statement above. Since the year for output of the period average is set at 0000, the file name is GSW2B1b_00000000.one. The user can immediately find the file by inputting the following.

```
% ls ../../data_bin/Tair____/*00000000*
```

5.8 htid (X, Y, L, lon and lat Coordinate Conversions)

This command displays the X and Y coordinates and L coordinate for a longitude and latitude input by the user. This command also displays the longitude, latitude and L coordinate for X and Y coordinates input by the user. This command also displays the longitude, latitude and X and Y coordinates for an L coordinate input by the user. This command is convenient when the user wants to track time series data for one location only.

5.8.1 Order of arguments

The arguments for this command vary according to the method for indicating the grid.

Specified by L coordinate system

```
% htid n0l n0x n0y c0l2x c0l2y
           p0lonmin p0lonmax p0latmin p0latmax l i0l
```

Specified by XY coordinate system

```
% htid n0l n0x n0y c0l2x c0l2y
```

```
p0lonmin p0lonmax p0latmin p0latmax xy i0x i0y
```

Specified by longitude and latitude

```
% htid n0l n0x n0y c0l2x c0l2y
      p0lonmin p0lonmax p0latmin p0latmax lonlat r0lon r0lat
```

5.8.2 Example of execution

Use the `htid` command to look up the X, Y and L coordinates for Tokyo (139.5° East, 35.5° North).

First create a directory for executing the `htid` command and create L coordinate system and XY coordinate system conversion ASCII files (`l2x.one.txt` and `l2y.one.txt`) in that directory.¹⁹

```
% mkdir ${DIRSEMI}/Chapter5/HTID
```

```
% cd ${DIRSEMI}/Chapter5/HTID
```

```
% ht12x12y 64800 360 180 l2x.one.txt l2y.one.txt
```

Use the `htid` command to look up the X, Y and L coordinates for Tokyo (139.5° East, 35.5° North). The output results signify, in order from the left, the X, Y and L coordinates.

```
% htid 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90
                                                    lonlat 139.5 35.5
320      55      19760
```

Look up the longitude, latitude and XY coordinates for L coordinate 19760.

```
% htid 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90 1 19760
139.5000      35.50000      320      55
```

Look up the longitude, latitude and L coordinate for XY coordinates (320, 55).

```
% htid 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90 xy 320 55
139.5000      35.50000      19760
```

¹⁹ Alternatively, move to any directory that has L coordinate system and XY coordinate system conversion ASCII files (`l2x.one.txt` and `l2y.one.txt`).

5.9 htedit (Edit Any Cell)

This command edits the data in any one cell location in an H08 Format 2D binary file. htedit is a command for directly editing an H08 Format 2D binary file. The author recommends that a backup be made of the file to be edited before using it to preserve the file in the original state.

5.9.1 Order of arguments

The arguments for this command vary according to the method for indicating the grid.

Specified by L coordinate system

```
% htedit n0l n0x n0y c0l2x c0l2y
      p0lonmin p0lonmax p0latmin p0latmax l c0bin r0dat i0l
```

Specified by XY coordinate system

```
% htedit n0l n0x n0y c0l2x c0l2y
      p0lonmin p0lonmax p0latmin p0latmax xy c0bin r0dat i0x i0y
```

Specified by longitude and latitude

```
% htedit n0l n0x n0y c0l2x c0l2y p0lonmin p0lonmax
      p0latmin p0latmax lonlat c0bin r0dat r0lon r0lat
```

5.9.2 Example of execution

Use the command htid to some of the data in an EPS file from an H08 Format 2D binary file (`${DIRSEMI}/data_bin/lndmsk.GSWP2.one`).

First move to the directory for the htedit command.

```
% cd ${DIRSEMI}/Chapter5/HTEDIT
```

Next, copy lndmsk.GSWP2.one.

```
% cp ../../data_bin/lndmsk.GSWP2.one .
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the htl2x12y command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is .one.txt.

```
% htl2x12y 64800 360 180 12x.one.txt 12y.one.txt
```

The htedit command is used to insert 1 into the cell for 3.5° West, 34.5° south.

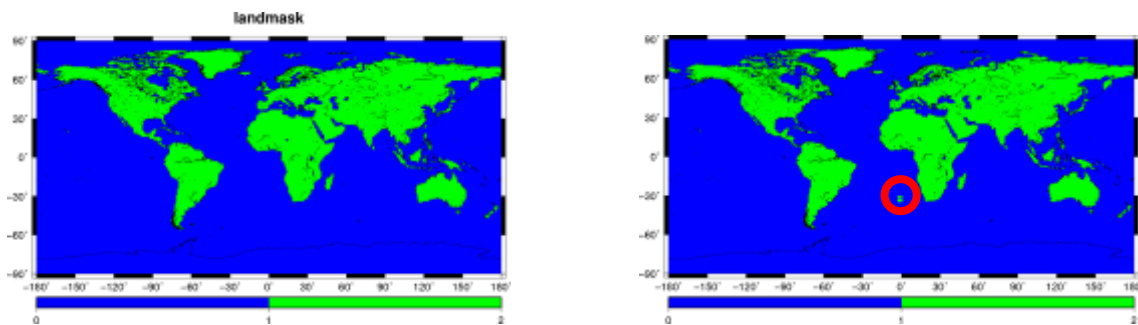
```
% htedit 64800 360 180 l2x.one.txt l2x.one.txt -180 180 -90 90
                                lonlat lndmsk.GSWP2.one 1 -3.5 -34.5
```

When the user inputs the command above,

```
htedit: original r1dat(      44817 )= 0.0000000E+00
htedit: revised  r1dat(      44817 )= 1.0000000
```

is displayed, and it means that the data in that cell has been edited.

However, when the user uses this command to edit just one cell, the change will hardly be noticeable even if checked by converting to an EPS file with the `htdraw` command. Therefore, change the latitude and longitude of the cell being edited from 0.5 to -3.5 and from -30.5 to -34.5, respectively. An EPS image created from the original land and sea map H08 Format 2D binary file (`.../H08/mat/dat/lnd_msk_/lndmsk.GSWP2.one`) is shown in Figure 4-6, and the EPS image created from the H08 Format 2D binary file edited by `htedit` is shown in Figure 4-7.



(a) Land and sea map

(b) Land and sea map from example

Figure 5-4 Comparison of Map before and after Executing `htedit`

5.10 `htpoint` (Display Data for Any Location)

This command displays the data for any location in an H08 Format 2D binary file. It is used when the user wants to see the data for a certain location.

5.10.1 Order of arguments

The arguments for this command vary according to the method for indicating the grid.

Specified by L coordinate system

```
% htpoint n0l n0x n0y c0l2x c0l2y
           p0lonmin p0lonmax p0latmin p0latmax l c0bin i0l
```

Specified by XY coordinate system

```
% htpoint n0l n0x n0y c0l2x c0l2y
           p0lonmin p0lonmax p0latmin p0latmax xy c0bin i0x i0y
```

Specified by longitude and latitude

```
% htpoint n0l n0x n0y c0l2x c0l2y
           p0lonmin p0lonmax p0latmin p0latmax lonlat c0bin r0lon r0lat
```

5.10.2 Example of execution

Use the htpoint command to find the average annual precipitation for Tokyo in 1986.

Using an H08 Format 2D binary file (`${DIRSEMI}/data_bin/Rainf___/GSW2B1b_19860000.one`) with the average precipitation for 1986, look up the value for Tokyo (139.5° East, 35.5° North).

First, move to `${DIRSEMI}/Chapter5/HTPOINT`.

```
% cd ${DIRSEMI}/Chapter5/HTPOINT
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the htl2x12y command. In this example, the spatial resolution is 1°×1°, so the extension is .one.txt.

```
% htl2x12y 64800 360 180 12x.one.txt 12y.one.txt
```

Next, copy the H08 Format 2D binary file (`${DIRSEMI}/data_bin/Rainf___/GSW2B1b_19860000.one`) for 1986 average precipitation to the current directory.

```
% cp ../../data_bin/Rainf___/GSW2B1b_19860000.one .
```

Finally, use the htpoint command to find the average annual precipitation for Tokyo in 1986.

```
% htpoint 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
           lonlat GSW2B1b_19860000.one 139.5 35.5
4.4085849E-05
```

5.11 htpointts (Display Period Data for Any Location)

This command displays the time series data for any location in an H08 Format 2D binary file. This command is convenient when the user wants to track time series data for one location. In addition, the results output for this command are displayed in the four columns of

year, month, day and data, so the command is useful for the user in creating an H08 Format 1D ASCII file for that location.

5.11.1 Order of arguments

The arguments for this command vary according to the method for indicating the grid.

Specified by L coordinate system

```
% htpointts n0l n0x n0y c0l2x c0l2y p0lonmin p0lonmax
           p0latmin p0latmax l c0bints i0yearmin i0yearmax i0l
```

Specified by XY coordinate system

```
% htpointts n0l n0x n0y c0l2x c0l2y p0lonmin p0lonmax
           p0latmin p0latmaxxy c0bints i0yearmin i0yearmax i0x i0y
```

Specified by longitude and latitude

```
% htpointts n0l n0x n0y c0l2x c0l2y p0lonmin p0lonmax p0latmin
           p0latmax lonlat c0bints i0yearmin i0yearmax r0lon r0lat
```

5.11.2 Example of execution

In this example, the htpointts command is used to output the value for the average temperature for each month in 1986 in Tokyo (139.5° North, 35.5° South).

First, move to `${DIRSEMI}/Chapter5/HTPOINTTS`.

```
% cd ${DIRSEMI}/Chapter5/HTPOINTTS
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the `htl2xl2y` command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`.

```
% htl2xl2y 64800 360 180 12x.one.txt 12y.one.txt
```

Next, display the temperature data for each month in 1986 in Tokyo using the `htpointts` command. Also create an H08 Format 1D ASCII file by writing the output results to `T_MO_tokyo1986.txt` with a redirect (`>`).

```
% htpointts 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
           lonlat ../../data_bin/Tair____/GSW2B1b_.oneMO 1986 1986 139.5 35.5
                                                    > T_MO_tokyo1986.txt
```

The content of `T_MO_tokyo1986.txt` is as follows. In order from the left are year, month, day

and data.

1986	1	0	275.8090
1986	2	0	275.9088
1986	3	0	279.8837
1986	4	0	286.0588
1986	5	0	289.8337
1986	6	0	293.3336
1986	7	0	296.6338
1986	8	0	299.1088
1986	9	0	295.9088
1986	10	0	288.8837
1986	11	0	284.0339
1986	12	0	280.2838

Figure 5-5 Content of T_MO_tokyo1986.txt

5.12 htmask (Mask Extraction)

This command outputs only cell values within the data in an H08 Format 2D binary file that satisfy conditions into a mask file of the same size. There are two methods for specifying the mask file, using the input file itself and using a separate file. The missing value of 10^{20} is entered for value for cells that do not satisfy the conditions.

5.12.1 Order of arguments

The order of the arguments for this command varies according to the application.

SIGN: sign (any of equals eq, inequality ne, greater than gt, greater than or equal ge, less than lt, less than or equal le)

Input file = mask file

The data for cells with condition (SIGN) at a certain value (r0dat) in the input file data is written in the output file **c0bin**. The rest are output as 10^{20} .

```
% htmask n0l n0x n0y c0l2x c0l2y p0lonmin p0lonmax
          p0latmin p0latmax c0bin c0bin SIGN r0dat c0bin
```

Input file \neq mask file

When the mask data is set to a condition (SIGN) at a certain value (r0dat), the input

file `c0bin` corresponding to that is written to the output file `c0bin`. 10^{20} is entered for the rest.

```
% htmask n0l n0x n0y c0l2x c0l2y p0lonmin p0lonmax
      p0latmin p0latmax c0bin c0msk SIGN r0dat c0bin
```

5.12.2 Example of execution

Input file = mask file

In this example, the `htmash` command is used to show only regions with annual precipitation of greater than 500 [mm/year] in 1986. Therefore, first prepare the annual precipitation file for 1986.

First, move to `${DIRSEMI}/Chapter5/HTMASK`.

```
% cd ${DIRSEMI}/Chapter5/HTMASK
```

Copy the H08 Format 2D binary file (`${DIRSEMI}/data_bin/Rainf___/GSW2B1b_19860000.one`) for 1986 average annual precipitation to the current directory.

```
% cp ../../data_bin/Rainf___/GSW2B1b_19860000.one .
```

Change the filename for `GSW2B1b_19860000.one` to `Rainf_1986.one`.

```
% mv GSW2B1b_19860000.one Rainf_1986.one
```

Since the units for this file, if not changed are [mm/s], change them to [mm/year]. Using the `htmath` command, multiply all of the data in `Rainf_1986.one` by 31536000 (=60×60×24×365) and output to `y_Rainf_1986.one`. This completes the H08 Format 2D binary file for annual precipitation in 1986.

```
% htmath 64800 mul Rainf_1986.one 31536000 y_Rainf_1986.one
```

Finally, create the XY coordinate and L coordinate conversion ASCII file using the `htl2x12y` command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`. This completes the preparations.

```
% htl2x12y 64800 360 180 l2x.one.txt l2y.one.txt
```

Next, with `y_rainf1986.one` as the input file and the input file itself as the mask file, extract only the locations where the annual precipitation is greater than 500 [mm/year].

```
% htmask 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90
          y_Rainf_1986.one y_Rainf_1986.one gt 500 temp.mask.one
```

Plot temp.mask.one using the htdraw command. Since the CPT file is in `$(DIRSEMI)/cpt`, copy it.

```
% cp ../../cpt/rain.cpt .
```

```
% htdraw 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90
          temp.mask.one rain.cpt temp.mask.eps rain_gt500[mm/y]
% htconv temp.mask.eps temp.mask.png rot
```

The PNG file temp.mask.png created above is shown in Figure 5-6.

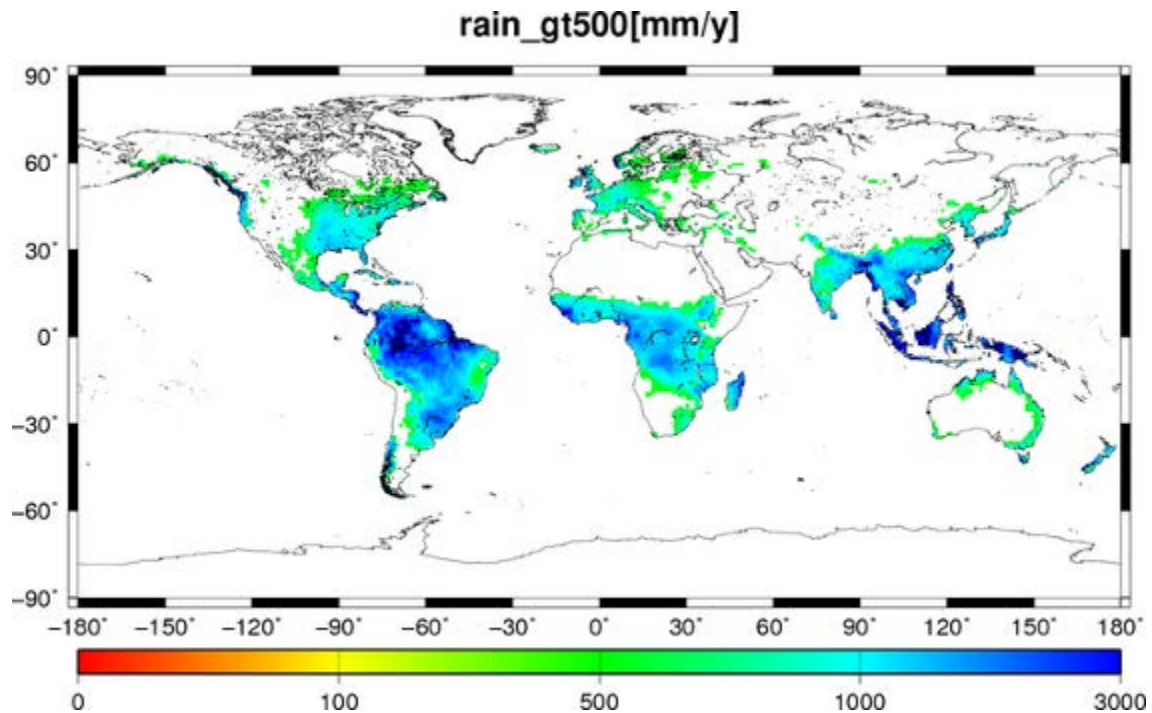


Figure 5-6 Precipitation Distribution (1986) for Regions with More Than 500 mm/year Precipitation

Input file \neq mask file

In this example, the htmask command is used to show the annual precipitation in the Congo River basin in 1986. Therefore, prepare an H08 Format 2D binary file with the basin IDs in it. First, move to `$(DIRSEMI)/Chapter5/HTMASK`.

```
% cd $(DIRSEMI)/Chapter5/HTMASK
```

Copy the H08 Format 2D binary file (`{DIRSEMI}/data_bin/rivnum.GSWP2.one`) with basin IDs into the current directory.

```
% cp ../../data_bin/rivnum.GSWP2.one .
```

Next, with `y_Rainf_1986.one` as the input file and `rivnum.GSWP2.one` as the mask file, extract only the Congo River basin out of the annual precipitation file for 1986. The ID of the Congo River basin is 2.

```
% htmask 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90
        y_Rainf_1986.one rivnum.GSWP2.one eq 2 Congo.rainf1986.one
```

Plot `Congo.rainf1986.one` using the `htdraw` command. For the CPT file, use `rain.cpt`, which was used when creating Figure 5-6.

```
% htdraw 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90
        Congo.rainf1986.one rain.cpt Congo.rainf1986.eps Congo_Rainf [mm/y]
% htconv Congo.rainf1986.eps Congo.rainf1986.png rot
```

The PNG file `Congo.rainf1986.png` created above is shown in Figure 5-7.

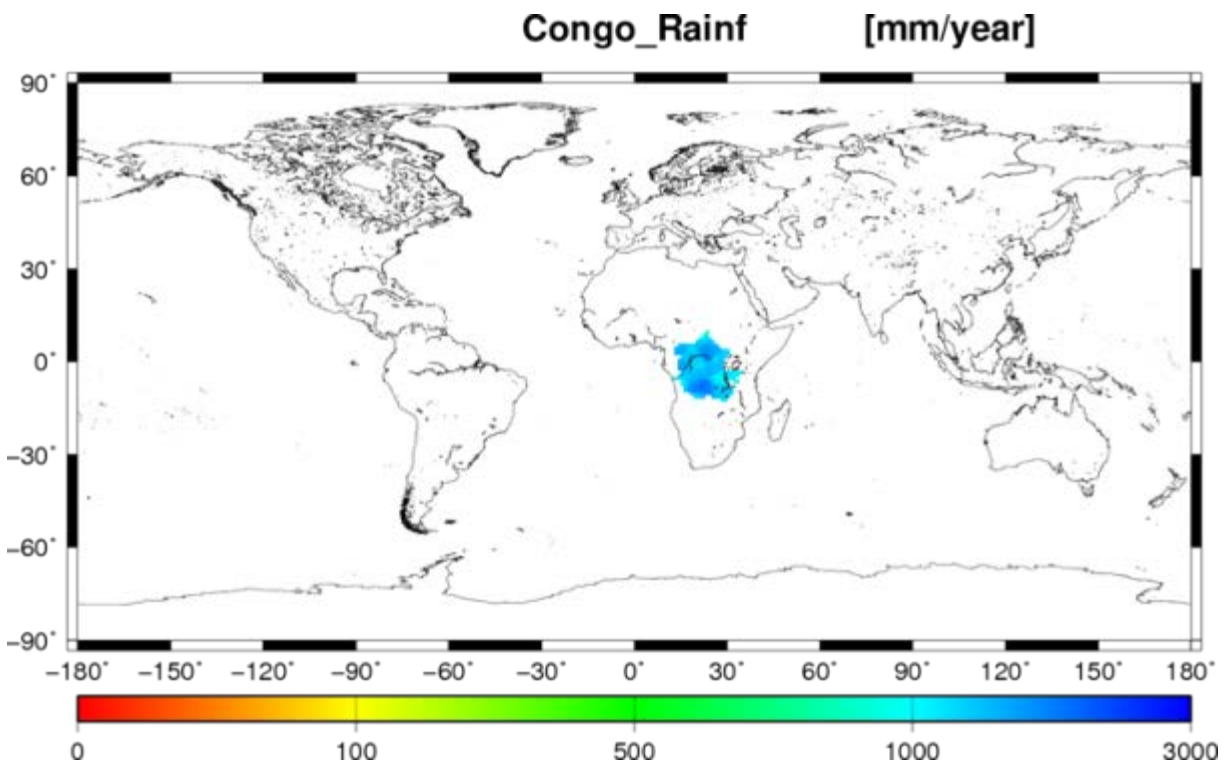


Figure 5-7 1986 Precipitation in Congo River Basin [mm/year]

5.13 htmaskrplc (Mask Replacement)

This command changes and outputs cell values within the data in an H08 Format 2D binary file that satisfy conditions into a mask file of the same size. There are two methods for specifying the mask file, using the input file itself and using a separate file. The original value is output as is for value for cells that do not satisfy the conditions.²⁰

5.13.1 Order of arguments

The order of the arguments for this command varies according to the application.

SIGN: sign (any of equals eq, inequality ne, greater than gt, greater than or equal ge, less than lt, less than or equal le)
 r0dat1: number given for mask file condition
 r0dat2: number substituted for numerical value in cell satisfying mask file condition

Input file = mask file

When r0dat1 and SIGN (condition) are set with the data in the input file c0bin, the cell data satisfying that condition is replaced by r0dat2 and written to the output file **c0bin**. For cells other than those cells, the data in the input file is output as is.

```
% htmaskrplc n0l n0x n0y c0l2x c0l2y p0lonmin p0lonmax
           p0latmin p0latmax c0bin c0bin SIGN r0dat1 r0dat2 c0bin
```

Input file ≠ mask file

When the data in the mask file sets r0dat1 and SIGN (condition), the cell data in the input file c0bin corresponding to the positions of cells satisfying that condition are replaced by r0dat2 and written to the output file c0bin. For cells other than those cells, the data in the input file is output as is.

```
% htmaskrplc n0l n0x n0y c0l2x c0l2y p0lonmin p0lonmax
           p0latmin p0latmax c0bin c0msk SIGN r0dat1 r0dat2 c0bin
```

5.13.2 Example of execution

In this example, the htmaskrplc command is used to replace data in the average annual precipitation file for 1986 with any value corresponding to the range of the amount of rainfall. First, all of the files used in this example will be prepared; next, a trial will be run of the

²⁰ See: HO8 Documentaiton 1, Chapter 6 Grid Data Processing (2)

htmaskrplc command for input file = mask file; next, a trial run for input file \neq mask file; and finally, the output results drawn and checked using the htdraw command.

Prepare the files to be used in this example. First, move to `${DIRSEMI}/Chapter5/HTMASKRPLC`.

```
% cd ${DIRSEMI}/Chapter5/HTMASKRPLC
```

Copy the H08 Format 2D binary file (`${DIRSEMI}/data_bin/Rainf___/GSW2B1b_19860000.on`) for 1986 average annual precipitation and the land and sea map binary file (`${DIRSEMI}/data_bin/lndmsk.GSWP2.one`) to the current directory.

```
% cp ../../data_bin/Rainf___/GSW2B1b_19860000.one .
```

```
% cp ../../data_bin/lndmsk.GSWP2.one .
```

Change the filename for `GSW2B1b_19860000.one` to `Rainf_1986.one`.

```
% mv GSW2B1b_19860000.one Rainf_1986.one
```

Since the units for this file, if not changed are [mm/s], change them to [mm/year]. Using the `htmath` command, multiply all of the data in `Rainf_1986.one` by 31536000 (=60×60×24×365) and output to `y_Rainf_1986.one`. This completes the H08 Format 2D binary file for annual precipitation in 1986.

```
% htmath 64800 mul Rainf_1986.one 31536000 y_Rainf_1986.one
```

Finally, create the XY coordinate and L coordinate conversion ASCII file using the `htl2xl2y` command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`. This completes the preparations.

```
% htl2xl2y 64800 360 180 l2x.one.txt l2y.one.txt
```

The `htmaskrplc` command is then used to replace data in the average annual precipitation file for 1986 with the value corresponding to the range of the amount of rainfall (Table 5-3).

Table 5-3 Replacement Correspondence Table

Precipitation [mm/year]	Value after replacement
0–50	1
50–100	2

100–500	3
500–1000	4
1000–3000	5
Sea	-999

First, run a trial of the `htmaskrplc` command for input file = mask file. Replace the data in `y_Raimf_1986.one` where cell values are an annual precipitation of 50 [mm/year] or less with 1 and write to `temp.mskrplc.one`.

```
% htmaskrplc 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                y_Rainf_1986.one y_Rainf_1986.one le 50 1 temp.mskrplc.one
```

Next, run a trial of the `htmaskrplc` command for input file \neq mask file. Replace the data in `y_Raimf_1986.one` where cell values are an annual precipitation of greater than 50 [mm/year] with 2 and add to `temp.mskrplc.one`. Therefore, the input and output file are `temp.mskrplc.one` and the mask file is `y_Raimf_1986.one`.

```
% htmaskrplc 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                temp.mskrplc.one y_Rainf_1986.one gt 50 2 temp.mskrplc.one
```

Also replace the data in `y_Raimf_1986.one` where cell values are an annual precipitation of greater than 100 [mm/year] with 3 and add to `temp.mskrplc.one`. Therefore, as with the command statement above, the input and output file are `temp.mskrplc.one` and the mask file is `y_Raimf_1986.one`.

```
% htmaskrplc 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                temp.mskrplc.one y_Rainf_1986.one gt 100 3 temp.mskrplc.one
```

Repeat the command statement above referring to Table 5-3.

```
% htmaskrplc 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                temp.mskrplc.one y_Rainf_1986.one gt 500 4 temp.mskrplc.one
```

```
% htmaskrplc 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                temp.mskrplc.one y_Rainf_1986.one gt 1000 5 temp.mskrplc.one
```

Finally, replace the cells for the sea in `temp.mskrplc.one` with -999. Here, the input and output file are `temp.mskrplc.one` and the mask file is `Indmsk.GSWP2.one`.

```
% htmaskrplc 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
temp.mskrplc.one lndmsk.GSWP2.one eq 0 -999 temp.mskrplc.one
```

Check the H08 Format 2D binary file created above. For the CPT file copy and use `${DIRSEMI}/cpt/rain_2.cpt`.

```
% cp ../../cpt/rain_2.cpt .
```

```
% htdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
temp.mskrplc.one rain_2.cpt temp.mskrplc.eps
Rainf1986 Rank5
% htconv temp.mskrplc.eps temp.mskrplc.png rot
```

The PNG file `temp_mskrplc.png` created above is shown in Figure 5-8.

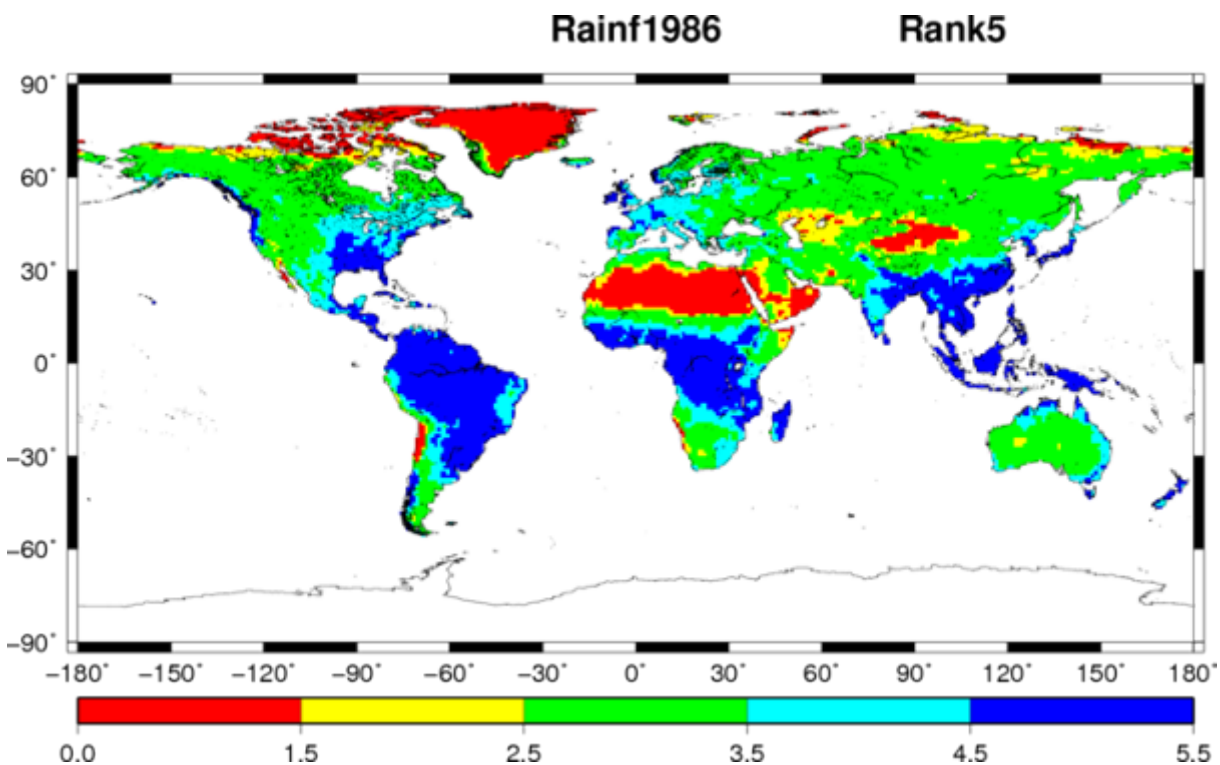


Figure 5-8 1986 Annual Precipitation Divided into Five Ranks (Table 5-3)

5.14 htlinear (Linear Interpolation)

Using this command, the user can extract rectangular data assigned in an H08 Format 2D binary file. In addition, this command can increase the resolution of the H08 Format 2D binary file by linear interpolation. For example, the user can create an H08 Format 2D binary file for rainfall in the Chao Phraya River basin with a spatial resolution of $5' \times 5'$ from an H08

Format 2D binary file for global rainfall with a spatial resolution of $1^\circ \times 1^\circ$ by using this command. This command reads the surrounding four pieces of data for a certain location, and calculates a linear interpolated value by weighting each using the distance ratio. When even one of the four pieces of data surrounding a certain location is missing (1.0E20), calculations cannot be done for that cell.

5.14.1 Order of arguments

```
% htlinear n0l n0x n0y c0l2x c0l2y p0lonmin
           p0lonmax p0latmin p0latmax n0l n0x n0y c0l2x c0l2y
           p0lonmin p0lonmax p0latmin p0latmax c0bin c0bin
```

This command is given the resolution for the input file and the resolution for the output file.

5.14.2 Example of execution

Using the htlinear command cut out the area around the Chao Phraya River basin at a spatial resolution of $5' \times 5'$ from an H08 Format 2D binary file with a spatial resolution of $1^\circ \times 1^\circ$. The longitude and latitude of the rectangle are $97 - 102^\circ$ East and $13 - 20^\circ$ North, respectively.

First, move to `${DIRSEMI}/Chapter5/HTLINEAR`.

```
% cd ${DIRSEMI}/Chapter5/HTLINEAR
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the ht12x12y command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`.

```
% ht12x12y 64800 360 180 12x.one.txt 12y.one.txt
```

Also create the XY coordinate and L coordinate conversion ASCII file for the area around the Chao Phraya River basin. The extension for this file is `.cp5.txt`. 5° of longitude and 7° of latitude are both divided into $5'$ resolution. Therefore, the number of divisions along the X-axis and the number of divisions along the Y-axis are $n0x = 60$ and $n0y = 84$, respectively.

```
% ht12x12y 5040 60 84 12x.cp5.txt 12y.cp5.txt
```

Since the H08 Format 2D binary file (`${DIRSEMI}/data_bin/Tair____/GSW2B1b_19860000.one`) for 1986 average annual temperature is used, copy it to the current directory.

```
% cp ../../data_bin/Tair____/GSW2B1b_19860000.one .
```

First, draw the H08 Format 2D binary file for the 1986 annual average temperature using the `htdraw` command. For the CPT file copy and use `${DIRSEMI}/cpt/Tair.cpt`.

```
% cp ../../cpt/Tair.cpt .
% htdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                                     GSW2B1b_19860000.one Tair.cpt temp.eps
% htconv temp.eps temp.png rot
```

Figure 5-9 shows `temp.png`, which was created by the command statement above.

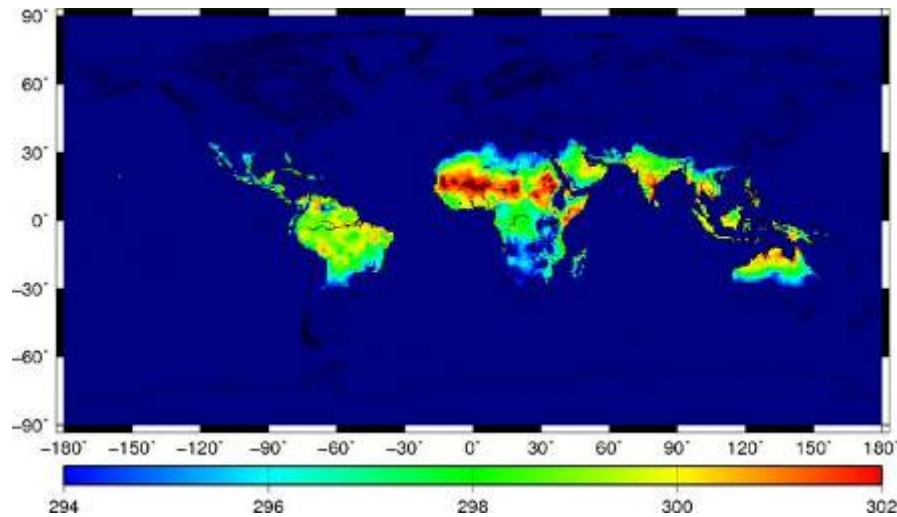


Figure 5-9 Average Temperatures for 1986

Here, the focus is on the area around the Chao Phraya River basin. Figure 5-10 (`${DIRSEMI}/data_fig/cp_one.jpg`) is an enlarged image of the area around the Chao Phraya River basin in Figure 5-9. One can see that temperature data at a resolution of $1^\circ \times 1^\circ$ is there.

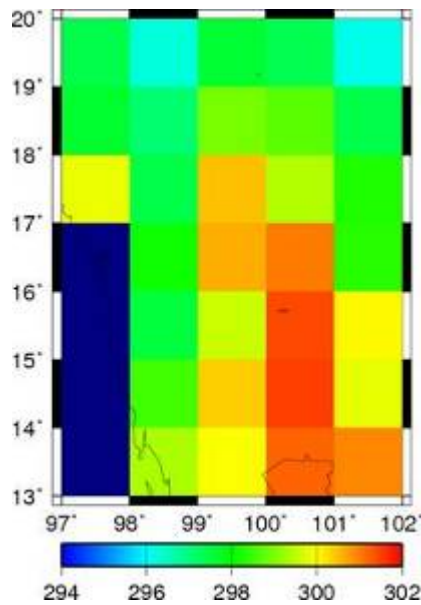


Figure 5-10 1986 Average Temperatures around the Chao Phraya River basin
 (\${DIRSEMI}/data_fig/cp_one.jpg)

However, since the resolution is coarse in this image, the `htlinear` command is used to create an H08 Format 2D binary file for only the area around the Chao Phraya River basin at a spatial resolution of 5'x5' from the H08 Format 2D binary file with a spatial resolution of 1°x1°. The `htlinear` command specifies the spatial resolution of the original H08 Format 2D binary file by nine arguments given at the beginning and specifies the spatial resolution of the rectangle (area around the Chao Phraya River basin in this example) H08 Format 2D binary file by the following nine arguments. Finally, the input file name and the output file name are given in this command.

```
% htlinear 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
           5040 60 84 12x.cp5.txt 12y.cp5.txt 97 102 13 20
           GSW2B1b_19860000.one GSW2B1b_19860000.cp5
```

Create an EPS file from the H08 Format 2D binary file (`GSW2B1b_19860000.cp5`) created by the `htlinear` command using `htdraw`. Also, convert the EPS file created to a PNG file using `htconv` and rotate it 90°.

```
% htdraw 5040 60 84 12x.cp5.txt 12y.cp5.txt 97 102 13 20
           GSW2B1b_19860000.cp5 Tair.cpt temp.cp5.eps
% htconv temp.cp5.eps temp.cp5.png rot
```

The image file output by the command statements above is shown in Figure 5-11.

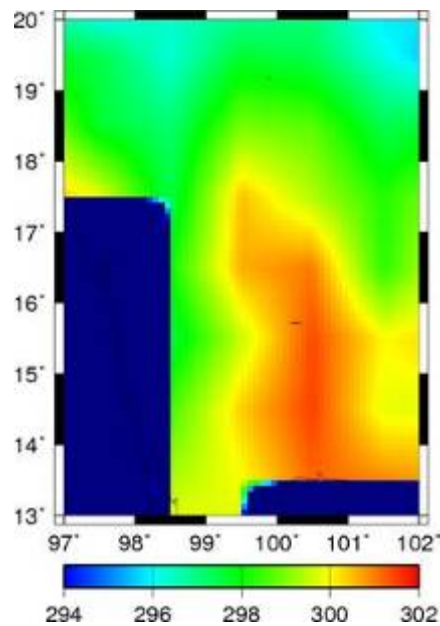


Figure 5-11 1986 Average Annual Temperatures around the Chao Phraya River basin (after Data Interpolation)

Figure 5-11 is a higher resolution H08 Format 2D binary file than Figure 5-10. However, 0 is in the sea cells in the H08 Format 2D binary file for the 1986 annual average temperature. Therefore, the values in the vicinity of the coast in GSW2B1b_19860000.cp5 are in the form of interpolations toward absolute zero and are not realistic. Therefore, the missing values (1.0E20) for the cells for the sea in GSW2B1b_19860000.one are replaced using the `htmaskrplc` command. Since `$(DIRSEMI)/data_bin/lndmsk.GSWP2.one` is used for the mask file, current directory.

```
% cp ../../data_bin/lndmsk.GSWP2.one .
```

```
% htmaskrplc 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
              GSW2B1b_19860000.one lndmsk.GSWP2.one
                                              eq 0 1.0E20 mskd_Tair1986.one
```

As with the H08 Format 2D binary file (GSW2B1b_19860000.one) used when drawing Figure 5-11, use the `htlinear` command to pull out the area around the Chao Phraya River basin in `mskd_Tair1986.one` and increase the spatial resolution.

```
% htlinear 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
           5040 60 84 12x.cp5.txt 12y.cp5.txt 97 102 13 20
                                              mskd_Tair1986.one mskd_Tair1986.cp5
```

Create an EPS file from `mskd_Tair1986.cp5` using the `htdraw` command. Also, convert the

EPS file created to a PNG file using `htdraw` and rotate it 90°.

```
% htdraw 5040 60 84 l2x.cp5.txt l2y.cp5.txt 97 102 13 20
           mskd_Tair1986.cp5 Tair.cpt mskd_temp.eps Temperature[K]
% htconv mskd_temp.eps mskd_temp.png rot
```

Figure 5-12 shows `mskd_temp.png`, which was created by the command statement above. None of the commands in H08 Analysis Tools are configured to process missing values; therefore, abnormal interpolations are not carried out for this figure. From the above, the user can create a 1986 average temperature H08 Format 2D binary file for the area around the Chao Phraya River basin at a high spatial resolution. Moreover, the `htlinear` command does not perform calculations for cells when even one of the four pieces of data surrounding a certain location is missing (1.0E20).

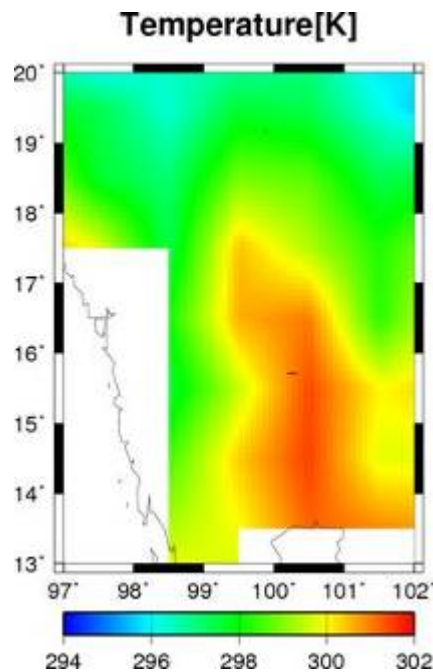


Figure 5-12 1986 Average Annual Temperatures around the Chao Phraya River basin (after Mask Processing and Data Interpolation)

5.15 htlist2bin (Create H08 Format 2D from List)

This command converts an ASCII file (`c0lst`) written in a table format to an H08 Format 2D binary file.

5.15.1 Order of arguments

```
% htlist2bin n01 c0lst c0msk c0cod c0bin OPTION (WGT)
```

OPTION: select from perarea, conserve and weight.

perarea: numerical values given by c0dat are inserted into all cells within an area.

conserve: numerical values given by c0dat are thought of as total numbers within an area.

weight: numerical values given by c0dat are given a special weighting within an area.

WGT: specifies a binary file that provides weighting when weight is selected.

5.15.2 Example of execution

Consider the creation of an H08 Format 2D binary file from an ASCII file for a table of population by country.

First, move to `${DIRSEMI}/Chapter5/HTLIST2BIN`.

```
% cd ${DIRSEMI}/Chapter5/HTLIST2BIN
```

File preparation

1. Copy the table of population by country (`${DIRSEMI}/data_asc/pop_list.txt`) to the current directory.

```
% cp ../../data_asc/pop_list.txt .
```

2. Prepare a table of codes by country name. Note that the country codes are determined by the ISO (International Organization for Standardization).

```
% cp ../../data_asc/C05____20000000.txt .
```

Also prepare an H08 Format 2D binary file with country codes in each cell.

```
% cp ../../data_bin/C05____20000000.one .
```

Execution example for using perarea option

1. Try using the perarea option and executing the htlist2bin command.

```
% htlist2bin 64800 pop_list.txt
                C05____20000000.one C05____20000000.txt co_pop.one perarea
```

After executing the command, the following sentence is output.

```
htlist2bin: All records transferred successfully.
```

2. Try drawing the `co_pop.one` that was created. First convert to easily handled numerical values by dividing `co_pop.one` by 10^9 using the `htmath` command.

```
% htmath 64800 div co_pop.one 1000000000 divd_co_pop.one
```

Next, prepare the CPT file

```
% cp ../../cpt/co_pop.cpt .
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the `htl2xl2y` command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`.

```
% htl2xl2y 64800 360 180 l2x.one.txt l2y.one.txt
```

Finally, create an EPS file from `divd_co_pop.one` using the `htdraw` command. Also, convert the EPS file created to a PNG file and rotate it 90° .

```
% htdraw 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90
                    divd_co_pop.one co_pop.cpt co_pop.eps Population 10^8
% htconv co_pop.eps co_pop.png rot
```

The image file created by the commands above is shown in Figure 5-13.

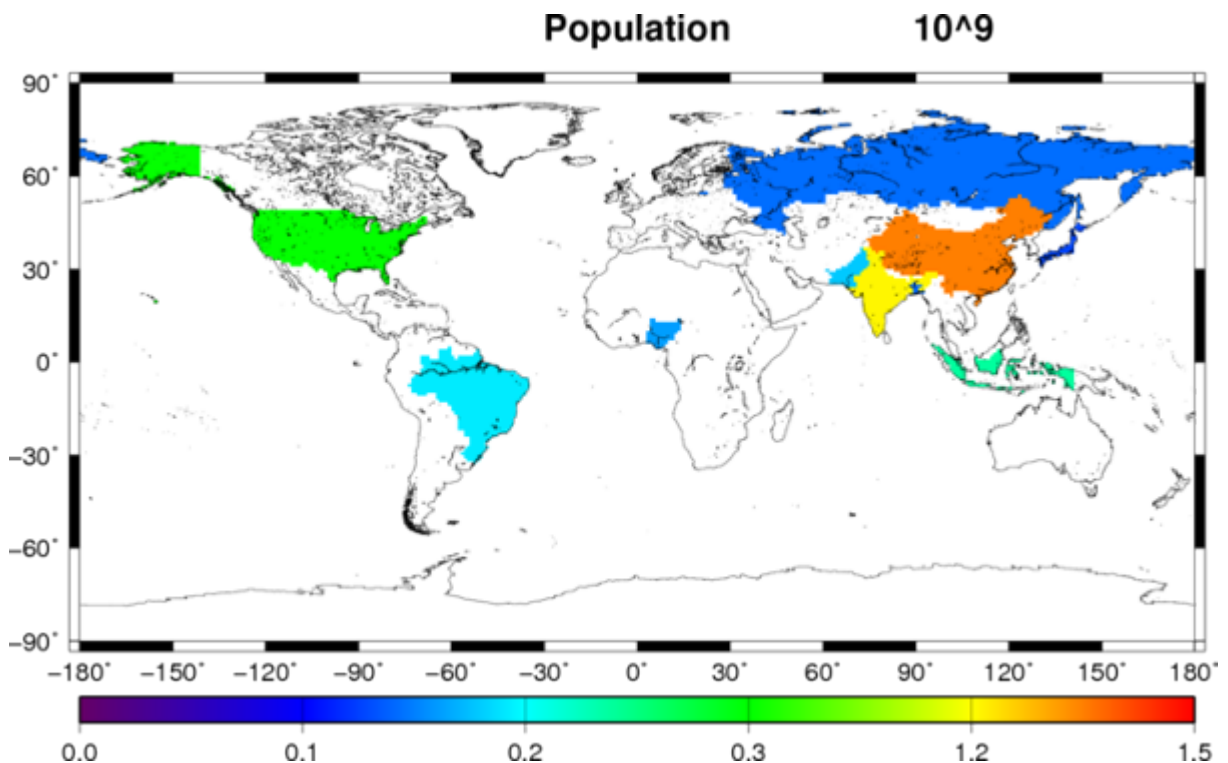


Figure 5-13 Map of Population by Country

Since the population of China in `pop_list.txt` is 1.35 billion, China as a whole is colored orange in Figure 5-13. Also, since the population of the United States in `pop_list.txt` is 313 million, the United States as a whole is colored a greenish yellow in Figure 5-13. Thus, if the user uses the `htlist2bin` command with the `perarea` option, the entire grid for each of the classifications specified by the mask file have the data specified by `c0lst` entered in them.

Execution example for using conserve option

With this `conserve` option, the value that results from the data being divided by the total number of mask cells is output to the target cell.

1. Try using the `conserve` option and executing the `htlist2bin` command.

```
% htlist2bin 64800 pop_list.txt
                C05____20000000.one C05____20000000.txt co_pop_con.one conserve
```

After executing the command, the following sentence is output.

```
htlist2bin: All records transferred successfully.
```

2. Try drawing the binary file that was created.

Prepare the CPT file.

```
% cp ../../cpt/co_pop_con.cpt .
```

Finally create an EPS file from `co_pop_con.one`. Also, convert the EPS file that has been created to a PNG file using `htconv` and rotate it 90°.

```
% htdraw 64800 360 180 l2x.one.txt l2y.one.txt -180 180 -90 90
                co_pop_con.one co_pop_con.cpt co_pop_con.eps con_Population pop/grid
% htconv co_pop_con.eps co_pop.png_con rot
```

The PNG file created by the commands above is shown in Figure 5-14.

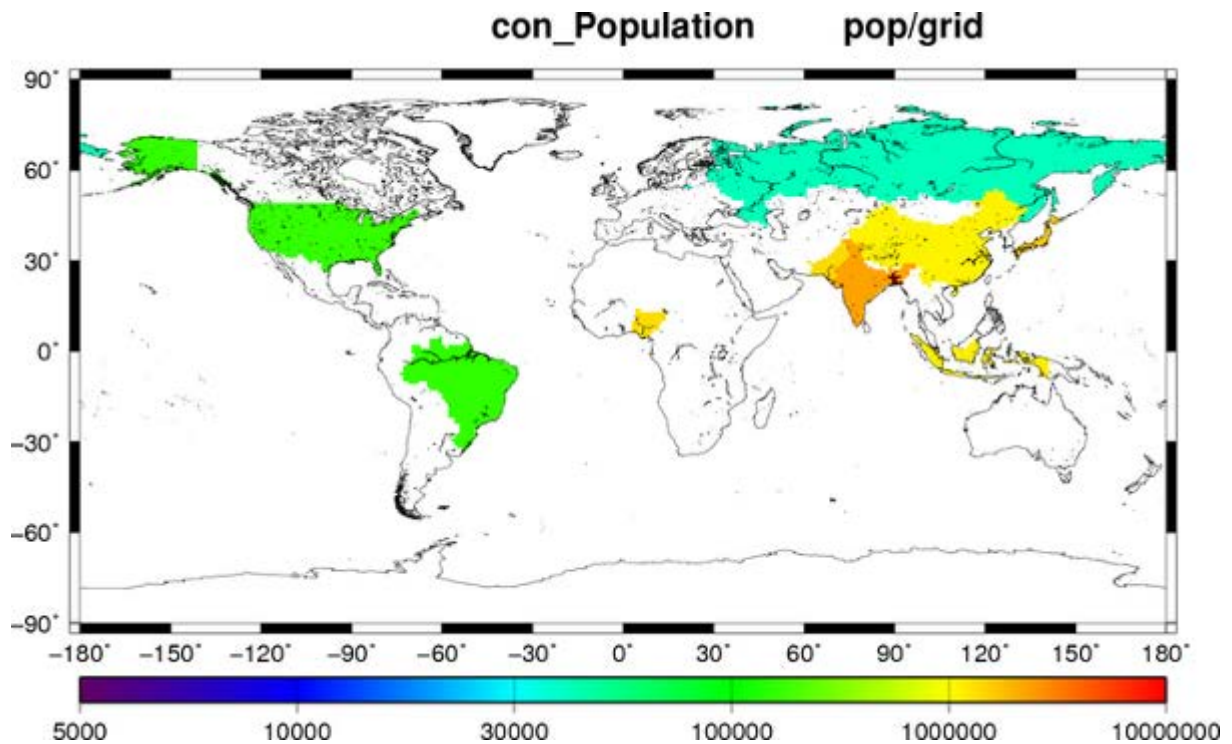


Figure 5-14 Population per Grid Cell by Country

Compare China and Bangladesh in Figure 5-14. The population of China is 1.35 billion, and the population of Bangladesh is 142 million; therefore, the population of China is roughly 10 times that of Bangladesh. However, looking at Figure 5-14, Bangladesh is redder, and can be seen to have a higher numerical value. This is because that country has a higher population per grid cell. The population of Russia is 143 million and is roughly the same as the population of Bangladesh, but since the population per grid cell is smaller, it is light blue. Thus, if the user uses the `htlist2bin` command with the `conserve` option, the data given by `c0lst` is divided by the number of grid cells for each area as specified by the mask file.

Execution example for using weight option

`htlist2bin` with the `weight` option can weight data within an area if a binary file is provided. Consider industrial water use in this example. Since industrial water use is water use by people, it would be presumed that industrial water use would be large in areas with large populations. Prepare a list of industrial water usage in the 10 countries with the largest populations, and convert it to an H08 Format 2D binary file weighted by population.

First, prepare industrial water usage list.

```
% cp ../../data_asc/Industrial_Water_list.txt .
```

Also prepare an H08 Format 2D binary file with population numbers in each cell.

```
% cp ../../data_bin/C05_a___20000000.one .
```

1. Using the `htlist2bin` command, convert `Industrial_Water_list.txt` to anH08 Format 2D binary file. At this time set the weight option, and weight by population.

```
% htlist2bin 64800 Industrial_Water_list.txt
      C05___20000000.one C05___20000000.txt Indust.one weight
                                           C05_a___20000000.one
```

After executing the command, the following sentence is output.

```
htlist2bin: All records transferred successfully.
```

2. Try drawing the binary file that was created.

First, convert the units in `Indust.one` from $[10^9\text{m}^3/\text{year}]$ to $[10^3\text{m}^3/\text{year}]$ using the `htmath` command.

```
htmath 64800 mul Indust.one 1.0E+6 Indust.one
```

`Indust.one` is values based on the quantity of water intake. Since the proportion²¹ of consumption in the quantity of water intake for industrial water usage is approximately 1/10 worldwide, use the `htmath` command to convert the values to a consumption basis.

```
htmath 64800 div Indust.one 10 Indust.one
```

Next, copy `Industrial_Water.cpt` from `#{DIRSEMI}/cpt` to the current directory.

```
% cp ../../cpt/Industrial_Water.cpt .
```

Then create an EPS file from `Indust.one` using the `htdraw` command. Also, convert the EPS file that has been created to a PNG file using `htconv` and rotate it 90° .

```
% htdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
      Indust.one Industrial_Water.cpt Indust.eps
                                           Industrial_Water [10^3kg/year]
% htconv Indust.eps Indust.png rot
```

The image created by the commands above is shown in Figure 5-15.

²¹ Shiklomanov, I.A., (2000) Appraisal and assessment of world water resources, *Water Int.* 25, 11-32.

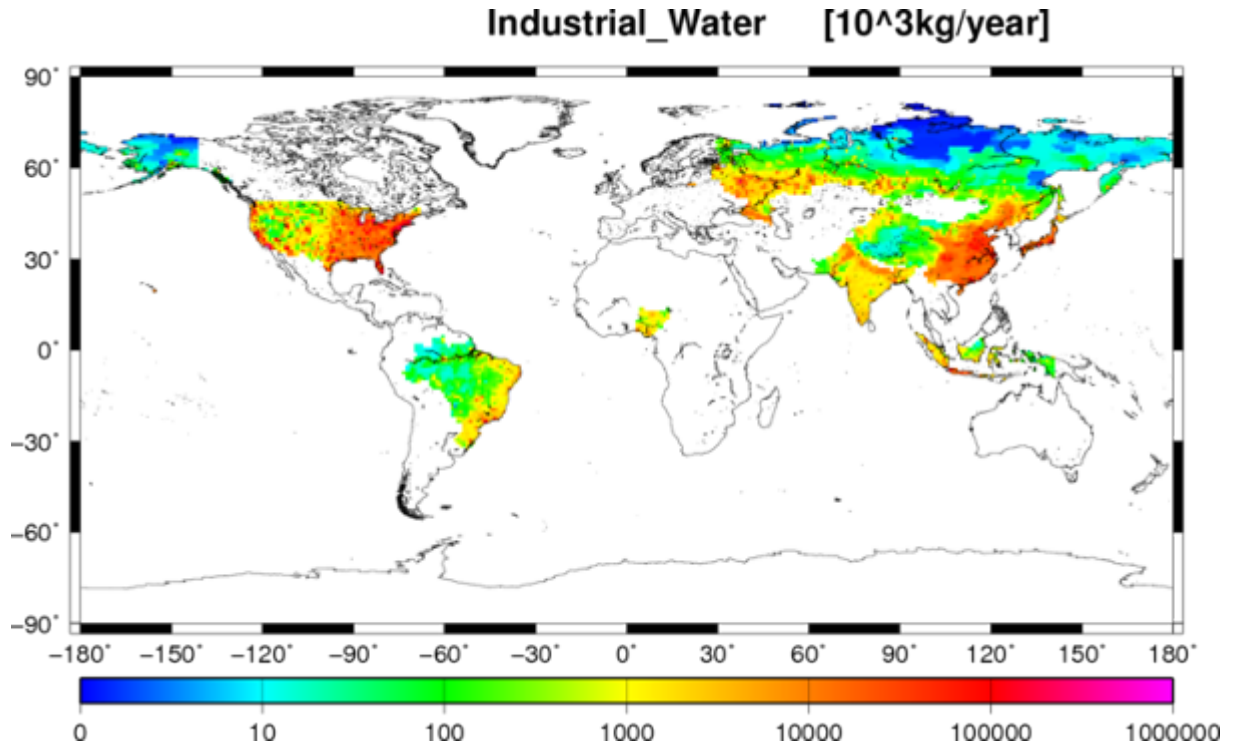


Figure 5-15 Industrial Water Use (consumption amount base) weighted by population

5.16 htbin2list (Convert H08 Format 2D Binary File to List)

This command creates a table format ASCII file (c0lst) for an H08 Format 2D binary file. In other words, the result is the exact opposite of the htlist2bin command; therefore, it is an effective command when one wants to make a table of data by country, by basin, or another category.

5.16.1 Order of arguments

```
% htbin2list n01 c0bin c0msk c0cod c0lst OPTION (WGT)
```

OPTION: select from sum, pergrid and weight.

sum: calculates the sum of data in all cells within an area.

pergrid: applies weighting by number of grid cells within an area and outputs the sum and weighted value.

weight: applies specific weighting within an area and outputs the sum and weighted value.

(WGT): specifies an H08 Format 2D binary file that provides weighting when weight is selected.

5.16.2 Example of execution

Using the `htbin2list` command with various options, create a table of population by country, table of population per unit grid and population density table.

File preparation

Prepare all of the files to be used in this example.

First, move to `${DIRSEMI}/Chapter5/HTBIN2LIST`.

```
% cd ${DIRSEMI}/Chapter5/HTBIN2LIST
```

Prepare a table of codes by country name.

```
% cp ../../data_asc/C05____20000000.txt .
```

Also prepare an H08 Format 2D binary file with country codes in each cell.

```
% cp ../../data_bin/C05____20000000.one .
```

Execution example for using sum option

When the user uses the `htbin2list` command, the sum of all the data in the cells within the area is output. Therefore, if the user provides `co_pop_con.one`²² to the `htbin2list` command, a file similar to `pop_list.txt`²³ is obtained.

```
% cp ../HTLIST2BIN/co_pop_con.one .
```

```
% htbin2list 64800 co_pop_con.one
                C05____20000000.one C05____20000000.txt list_sum.txt sum
```

Check the content of `list_sum.txt`. In the results of the output, the left column is the name of the country, the middle the total population and the right column the weighting value. When the user selects the sum option, no weighting is applied; therefore, all of the weighting values are `1.000E+00`.

²² See execution example in 5.15.2 for using the `conserve` option.

²³ See file preparation in 5.15.2.

```
% more list_sum.txt
Brazil                1.920E+08  1.000E+00
United_States_of_America  3.130E+08  1.000E+00
Russia                1.430E+08  1.000E+00
Bangladesh            1.420E+08  1.000E+00
China                  1.350E+09  1.000E+00
Indonesia              2.380E+08  1.000E+00
India                  1.210E+09  1.000E+00
Pakistan               1.790E+08  1.000E+00
Japan                  1.280E+08  1.000E+00
Nigeria               1.620E+08  1.000E+00
```

Execution example using pergrid option

When the user uses the `htbin2list` command, a value for the total of the data in the cells within the area divided by the number of cells in the area is output. Therefore, if the user makes an H08 Format 2D binary file with population numbers in each cell the input file for the `htbin2list` command with this option and provides a country mask H08 Format 2D binary file as the mask file, the value for the total population of any country divided by the number of cells for that country can be obtained.

First, prepare an H08 Format 2D binary file with population numbers in each cell.

```
% cp ../../data_bin/C05_a___20000000.one .
```

Next, use the `htbin2list` command with the `pergrid` option.

```
% htbin2list 64800 C05_a___20000000.one
      C05_____20000000.one C05_____20000000.txt list_pergrid.txt pergrid
```

Check the content of `list_pergrid.txt`. In the results of the output, the left column is the name of the country, the middle column the average value for population (total population divided by the number of mask cells for that country) and the right column the number of mask cells. Part of the results of the output is shown in the following.

```
% more list_pergrid.txt
Argentina          1.293E+05  2.840E+02
Bolivia            8.850E+04  9.400E+01
Brazil             2.235E+05  7.110E+02
Chile              1.564E+05  9.000E+01
Colombia           4.485E+05  8.800E+01
Ecuador            6.210E+05  2.000E+01
Falkland_Islands  5.798E+02  3.000E+00
French_Guiana      1.982E+04  6.000E+00
Guyana             3.472E+04  1.700E+01
Peru               1.870E+05  1.050E+02
```

Execution example for using weight option

When the user uses the `htbin2list` command, the data weighted by the mask file is output.

In this example, a H08 Format 2D binary file for average annual precipitation in 1986 is weighted by a H08 Format 2D binary file for land surface area, and a list of precipitation by country is output. First, copy the H08 Format 2D binary file (`${DIRSEMI}/data_bin/Rainf____/GSW2B1b_19860000.one`) for 1986 average annual precipitation to the current directory. Likewise, copy the H08 Format 2D binary file (`${DIRSEMI}/data_bin/lndara.GSWP2.one`) for land surface area.

```
% cp ../../data_bin/Rainf____/GSW2B1b_19860000.one .
```

```
% cp ../../data_bin/lndara.GSWP2.one .
```

Use the `htbin2list` command to weight `GSW2B1b_19860000.one` by `lndara.GSWP2.one`, and create an average annual precipitation list by country.

```
% htbin2list 64800 GSW2B1b_19860000.one
          C05____20000000.one C05____20000000.txt temp.txt weight
                                                lndara.GSWP2.one
```

Confirm the content of `temp.txt`. In the results of the output, the left column is the name of the country, the middle the precipitation weighted by land surface area and the right column the surface area for the weighting value.

Argentina	2.433E-05	2.822E+12
Bolivia	3.541E-05	1.108E+12
Brazil	5.917E-05	8.449E+12
Chile	2.326E-05	7.316E+11
Colombia	7.882E-05	1.067E+12
Ecuador	4.547E-05	2.460E+11
Falkland_Islands	1.888E-05	7.710E+09
French_Guiana	8.234E-05	7.367E+10
Guyana	5.023E-05	2.084E+11
Peru	4.073E-05	1.261E+12
Paraguay	4.287E-05	3.739E+11

5.17 htcatchment (Display a Cell Upstream of Any Location)

If the user specifies any cell when using this command, an H08 Format 2D binary file with 1 inserted in the cell on the upstream side of that cell and 0 inserted in the other cells is obtained. For example, if the user sets any location in the mouth of the river or elsewhere, a mask file for that basin is obtained. By using this command in combination with the htmask, htstat or other commands, the user can check the water budget for that basin.

5.17.1 Order of arguments

The arguments for this command vary according to the method for indicating the grid.

Specified by L coordinate system

```
% htcatchment n0l n0x n0y c0l2x c0l2y
                p0lonmin p0lonmax p0latmin p0latmax
                l FLWDIR RIVSEQ RIVNUM c0bin i0l
```

Specified by XY coordinate system

```
% htcatchment n0l n0x n0y c0l2x c0l2y
                p0lonmin p0lonmax p0latmin p0latmax
                xy FLWDIR RIVSEQ RIVNUM c0bin i0x i0y
```

Specified by longitude and latitude

```
% htcatchment n0l n0x n0y c0l2x c0l2y
                p0lonmin p0lonmax p0latmin p0latmax
                lonlat FLWDIR RIVSEQ RIVNUM c0bin r0lon r0lat
FLWDIR: H08 Format 2D binary file with downstream direction in it
```

RIVSEQ: H08 Format 2D binary file with downstream order in it

RIVNUM: H08 Format 2D binary file with basin ID in it

5.17.2 Example of execution

In this example, the `htcatchment` command is used to create a mask binary file for the Amazon River basin, and that mask file is used to find the River basin surface area for the Amazon River.

First, move to `${DIRSEMI}/Chapter5/HTCATCHMENT`.

```
% cd ${DIRSEMI}/Chapter5/HTCATCHMENT
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the `htl2x12y` command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`.

```
% htl2x12y 64800 360 180 12x.one.txt 12y.one.txt
```

Next, prepare H08 Format 2D binary files with downstream direction in one, the downstream order in another and the basin IDs in another. Since these files are in `${DIRSEMI}/data_bin/`, copy them.²⁴

```
% cp ../../data_bin/flwdir.GSWP2.one .
% cp ../../data_bin/rivseq.GSWP2.one .
% cp ../../data_bin/rivnum.GSWP2.one .
```

Use the `htcatchment` command to create a mask file for the Amazon River basin. The mouth of the Amazon River is located at 50.5° West and 0.5° North.

```
% htcatchment 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
lonlat flwdir.GSWP2.one rivseq.GSWP2.one rivnum.GSWP2.one
mask.amazon.one -50.5 0.5
```

Try drawing `mask.amazon.one` with the `htdraw` command. Copy the CPT file from `${DIRSEMI}/cpt/`.

```
% cp ../../cpt/lndmsk.cpt .
```

Next, create an EPS file from `mask.amazon.one` using the `htdraw` command. Also, convert the EPS file that has been created to a PNG file using `htconv` and rotate it 90° . `mask.amazon.png`

²⁴ These files are created by carrying out the operations in H08 Manual User's Edition Chapter 6, Section 6.3.

is shown in Figure 5-16.

```
% htdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
      mask.amazon.one lndmsk.cpt mask.amazon.eps
                                     Mask_Amazon by_htcatchment
% htconv mask.amazon.eps mask.amazon.png rot
```

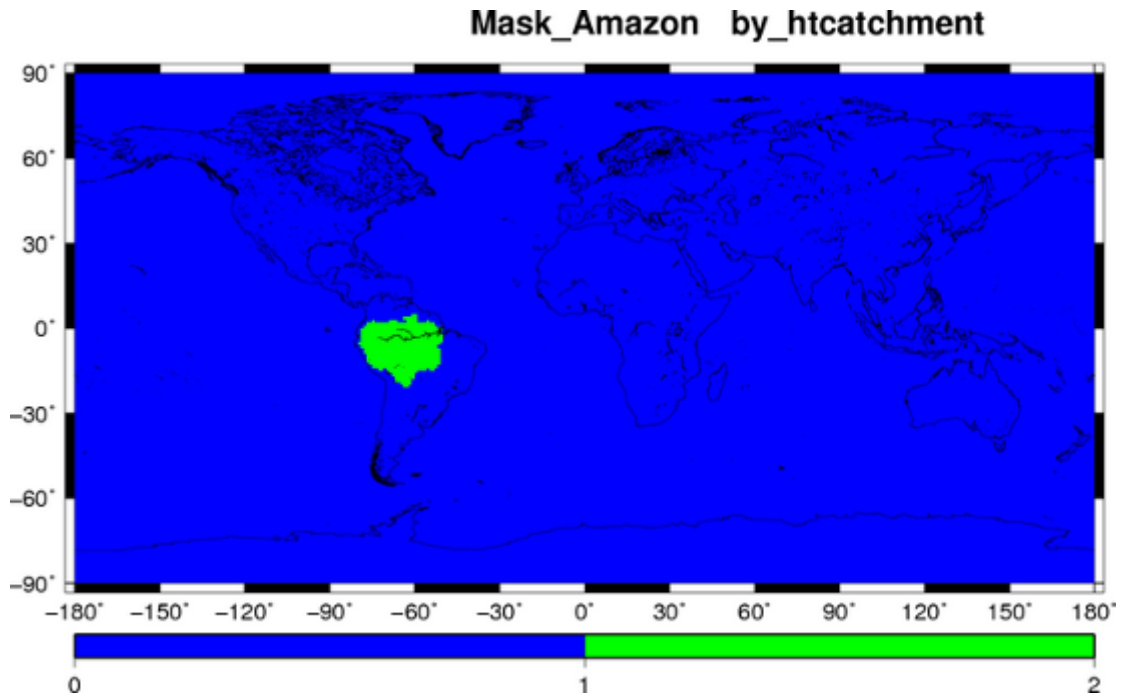


Figure 5-16 Land and Sea Map for Amazon River Basin Only

Thus, `htcatchment` is a command that can create an H08 Format 2D binary file in which 1 is inserted in grid cells positioned on the upstream of a specified grid cell and 0 is inserted in other grid cells.

Next, with `mask.amazon.one` as the mask file, use the `htmask` command on the H08 Format 2D binary file (`${DIRSEMI}/data_bin/grdara.one`) containing surface area per grid cell.

```
% cp ../../data_bin/grdara.one .
```

```
% htmask 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
      grdara.one mask.amazon.one eq 1 amazon.grdara.one
```

Finally, use the `htstat` command to find the surface area of the Amazon River basin.

```
% htstat 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
                                     sum amazon.grdara.one
6.1084784E+12          502
```

5.18 htpercentile (Calculate Percentile Value)

This command reads an H08 Format 2D binary file for a certain period and creates an H08 Format 2D binary file formed from the specified percentile values.

5.18.1 Order of arguments

```
% htpercentile n01 c0bints c0bin
                                i0yearmin i0yeaarmax THRESHOLD
```

THRESHOLD: percentile value

5.18.2 Example of execution

This example calculates 10th percentile values from monthly average temperature binary files for 1986 (total of 12). First, move to `${DIRSEMI}/Chapter5/HTPERCENTILE/`.

```
% cd ${DIRSEMI}/Chapter5/HTPERCENTILE
```

Set the percentile value at 10 and use the htpercentile command.

```
% htpercentile 64800 ../../data_bin/Tair____/GSW2B1_.oneMO percentile.one
                                1986 1986 10
```

Try mapping percentile.one using the htdraw command.

First, create the XY coordinate and L coordinate conversion ASCII file using the ht12x12y command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is .one.txt.

```
% ht12x12y 64800 360 180 12x.one.txt 12y.one.txt
```

Next, copy the CPT file from `${DIRSEMI}/cpt/`.

```
% cp ../../cpt/Tair_percentile.cpt .
```

Plot percentile.one using the htdraw command. The results of the output are shown in Figure 5-17.

```
% htdraw 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
    percentile.one Tair_percentile.cpt percentile.eps Tair1986 10percentile
```

```
% htconv percentile.eps percentile.png rot
```

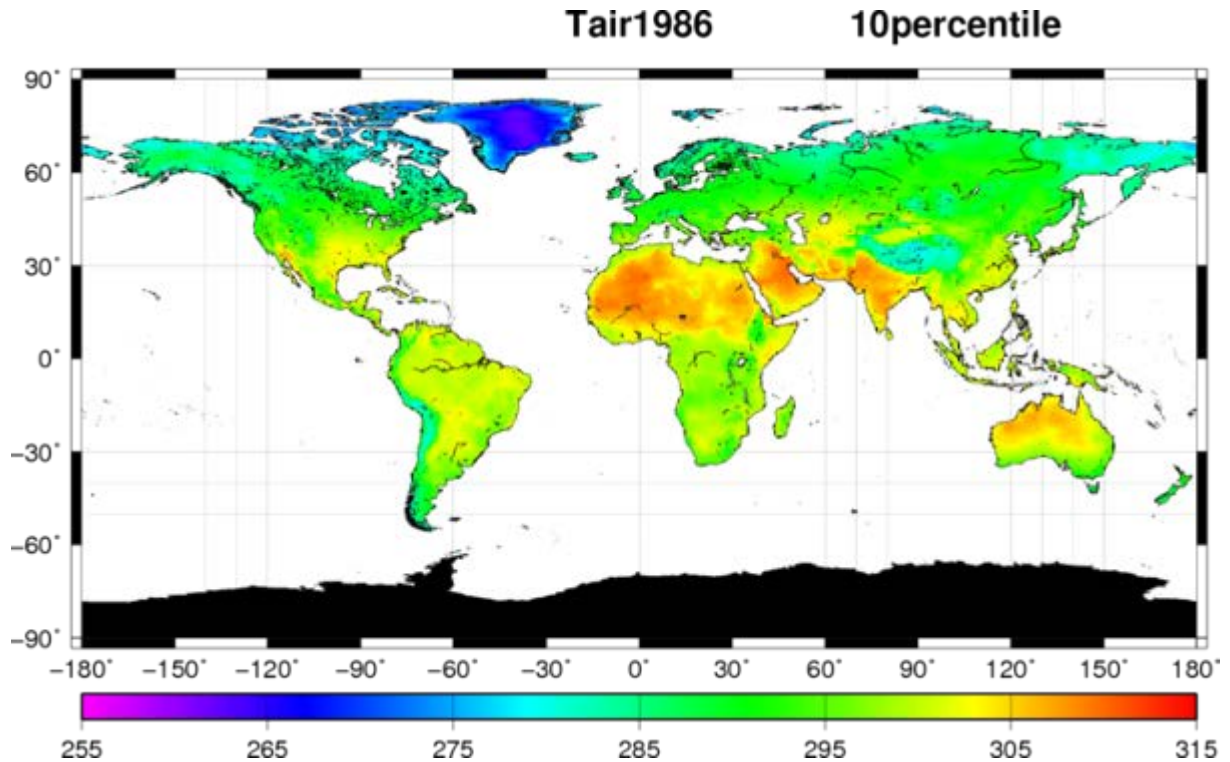


Figure 5-17 10th Percentile Values [K] for 1986 Temperatures

5.19 htrank (Create Ranking Table for Data for Any Location)

This command reads an H08 Format 2D binary file for a period specified by the user and displays the 10 largest or smallest data points for any location.

5.19.1 Order of arguments

The arguments for this command vary according to the method for indicating the grid.

OPTION1: any of largest10, smallest10, incord and decord.

largest10: outputs data up to 10th largest starting with the largest value.

smallest10: output data up to 10th smallest starting with the smallest value.

incord: arranges all data in the input file in ascending order.

decord: arranges all data in the input file in descending order.

Specified by L coordinate system

```
% htrank n0l n0x n0y c0l2x c0l2y
      p0lonmin p0lonmax p0latmin p0latmax OPTION1
      l c0bints i0yearmin i0yearmax i0l
```

Specified by XY coordinate system

```
% htrank n0l n0x n0y c0l2x c0l2y
      p0lonmin p0lonmax p0latmin p0latmax OPTION1
      xy c0bints i0yearmin i0yearmax i0x i0y
```

Specified by longitude and latitude

```
% htrank n0l n0x n0y c0l2x c0l2y
      p0lonmin p0lonmax p0latmin p0latmax OPTION1
      lonlat c0bints i0yearmin i0yearmax r0lon r0lat
```

5.19.2 Example of execution

In this example, the htrank command will be used on the average monthly temperature for 1986 and the top 10 months for average monthly temperature in Bangkok (100.5° East, 13.5° North) will be found.

First, move to `${DIRSEMI}/Chapter5/HTRANK/`.

```
% cd ${DIRSEMI}/Chapter5/HTRANK/
```

Next, create the XY coordinate and L coordinate conversion ASCII file using the `htl2x12y` command. In this example, the spatial resolution is $1^\circ \times 1^\circ$, so the extension is `.one.txt`.

```
% htl2x12y 64800 360 180 12x.one.txt 12y.one.txt
```

Using the htrank command find the top 10 months for average monthly temperature in Bangkok (100.5° East, 13.5° North).

```
% htrank 64800 360 180 12x.one.txt 12y.one.txt -180 180 -90 90
      largest10 lonlat ../../data_bin/Tair____/GSW2B1b._oneMO
      1986 1986 100.5 13.5
```

The output is as follows. From the left are rank, data, year, month and day.

```
htrank: i0recmax:      12
  1  303.5222          1986      4      0
  2  302.4223          1986      6      0
  3  302.3972          1986      5      0
  4  302.1223          1986      8      0
  5  301.7722          1986      9      0
  6  301.7473          1986      3      0
  7  301.5472          1986      7      0
  8  301.4221          1986     10      0
  9  300.3973          1986     11      0
 10  300.3223          1986      2      0
```

Chapter 6

ASCII File Processing Commands

In this chapter, the commands for the H08 Analysis Tools ASCII file processing will be described. The ASCII file processing commands mainly read H08 Format 1D ASCII files. Among the ASCII file processing commands are commands that can process in the same manner as the binary file processing commands `httime`, `htmean` and `htstat`. There are also commands that are convenient for the user to manage table data that is classified by country or basin.

6.1 `httimetxt` (Convert Time Resolution)

This command converts the time resolution in an H08 Format 1D ASCII file in the direction of lower resolution. In other words, this command carries out calculations similar to those for the `httime` binary file processing system command on H08 Format 1D ASCII files. For example, an H08 Format 1D ASCII file containing the monthly average data for a certain location is read and annual average data is calculated.

6.1.1 Order of arguments

```
% httimetxt c0ascts c0ascts
```

The user provides any of 3H, DY, MO and YR at the end of the file name to both the input file and output file according to the time resolution of the file. The time resolution of the output file must be lower than the time resolution of the input file.

6.1.2 Example of execution

In this example, the `httimetxt` command will be used on an ASCII file (`${DIRSEMI}/data_asc/test_MO_00000000.txt`) containing values for monthly average temperature in Tokyo (139.5° East, 35.5° North) from 1986 to 1988 to create an ASCII file with the annual average temperature for each year. First, move to `${DIRSEMI}/Chapter6/HTTIMETXT/`.

```
% cd ${DIRSEMI}/Chapter6/HTTIMETXT
```

Next, copy the ASCII file (`${DIRSEMI}/data_asc/test_MO_00000000.txt`) containing values for monthly average temperature in Tokyo (139.5° East, 35.5° North) from 1986 to 1988 to the current directory.

```
% cp ../../data_asc/test_MO_00000000.txt .
```

Provide test_MO_00000000.txt to the htmetxt command and create an ASCII file with annual average temperatures for each year in it. Since the time resolution for the input file is monthly units, provide MO at the end of the input file name. Also, since the time resolution for the input file will be yearly units, provide YR at the end of the output file name.

```
% htmetxt test_MO_00000000.txtMO test.yr.txtYR
```

By inputting the command above, the user obtains test.yr.txt containing values such as the following. In order from the left are year, month, day and data. Since the values are annual averages, there are zeros for month and day.

1986	0	0	287.2028
1987	0	0	288.3040
1988	0	0	287.3882

6.2 htmeantxt (Period Average Calculations)

This command calculates period averages for the data when the user provides an H08 Format 1D ASCII file. In other words, this command carries out calculations similar to those for the htmean binary command on H08 Format 1D ASCII files.

6.2.1 Order of arguments

```
% htmeantxt c0ascts c0ascts
```

- * The user provides any of 3H, DY, MO and YR for both the input file and output file according to the time resolution of the file. The time resolutions for both files must be the same.

6.2.2 Example of execution

In this example, three-year average values for each month will be calculated from the monthly average temperature data for Tokyo (139.5° East, 35.5° North) from 1986 to 1988.

First, move to `${DIRSEMI}/Chapter6/HTMEANTXT/`.

```
% cd ${DIRSEMI}/Chapter6/HTMEANTXT/
```

Next, copy the ASCII file (`${DIRSEMI}/data_asc/test_MO_00000000.txt`) containing values for monthly average temperature in Tokyo (139.5° East, 35.5° North) from 1986 to 1988 to the current directory.

```
% cp ../../data_asc/test_MO_00000000.txt .
```

Next, find the three-year average temperature for each month from 1986 using htmeantxt. Since the input file has the average temperature data for each month from 1986 to 1988 in it, the user provides MO at the end of the file name. Since the output file is also average data for each month, the user provides MO at the end of the file name as was done previously.

```
% htmeantxt test_MO_00000000.txtMO htmeantxt.out.txtMO
```

Check the content of htmeantxt.out.txt. That content is shown in the following. In order from the left are year, month, day and data. Since the values are average values for each month, there are zeros for year and day. Thus, the htmeantxt command can calculate the period averages from the starting year to the ending year.

```
% more htmeantxt.out.txt
0      1      0  277.6339
0      2      0  277.2005
0      3      0  280.4587
0      4      0  286.1421
0      5      0  290.3588
0      6      0  293.9837
0      7      0  296.8755
0      8      0  299.1838
0      9      0  295.4254
0     10      0  289.7421
0     11      0  283.9254
0     12      0  280.0421
```

6.3 htstattxt (Statistical Calculations)

This command calculates statistics (minimum value, maximum value, sum and average) for an H08 Format 1D ASCII file. In other words, this command can carry out calculations similar to those for the htstat binary file processing system command.

6.3.1 Order of arguments

```
% htstattxt OPTION c0ascts
```

OPTION: any of min, max, sum and ave is input.

The user provides any of 3H, DY, MO and YR according to the time resolution of the c0ascts.

6.3.2 Example of execution

In this example, various statistics will be found for the H08 Format 1D ASCII file (htmeantxt.out.txt) created by htmeantxt in section 5.1.

First, move to `${DIRSEMI}/Chapter6/HTMEANTXT/`.

```
% cd ${DIRSEMI}/Chapter6/HTMEANTXT/
```

Find the minimum value for htmeantxt.out.txt. Since this file has monthly average temperatures in it, provide MO at the end of the input file name. The output displays in order from the left data, year, month and day.

```
% htstattxt min htmeantxt.out.txtMO
277.2005      0      2      0
```

Find the maximum value for htmeantxt.out.txt. The output displays in the same order as that for the minimum value.

```
% htstattxt max htmeantxt.out.txtMO
299.1838      0      8      0
```

Find the sum for htmeantxt.out.txt.

```
% htstattxt sum htmeantxt.out.txtMO
3450.972
```

Finally, find the average for htmeantxt.out.txt.

```
% htstattxt ave htmeantxt.out.txtMO
287.5810
```

6.4 htcat (Combination of Table Format Files)

This command combines multiple table format ASCII files (c0lst). This command is convenient when the user wants to make a user-friendly table, such as one for Excel, from multiple sets of data.

6.4.1 Order of arguments

```
% htcat c01st1 c01st2 c0cod
```

The user can specify up to 100 c01st files.

6.4.2 Example of execution

In this example, the ASCII files for population and surface area for China, Brazil, the United States, Japan and Russia will be combined. The populations, surface areas and country codes for the various countries are given in Table 6-1, and Table 6-2 and Table 6-3, respectively. In addition, the files are in `${DIRSEMI}/data_asc/`.

Table 6-1 Population
[people] cat_pop.txt

China	1.35000000E+09
USA	3.13000000E+08
Japan	1.28000000E+08
Russia	1.43000000E+08
Brazil	1.92000000E+08

Table 6-2 Surface Area
[km²] cat_ara.txt

China	9.59696100E+06
Japan	3.77847000E+05
Russia	1.70982420E+07
Brazil	8.51487700E+06
USA	9.62909100E+06

Table 6-3 Country Codes
cat_code.txt

China	156
USA	840
Japan	392
Brazil	76
Russia	643

Move to `${DIRSEMI}/data_asc/`.

```
% cd ${DIRSEMI}/data_asc/
```

Then combine `cat_pop.txt` and `cat_ara.txt` using the `htcat` command.

```
% htcat cat_pop.txt cat_ara.txt cat_code.txt
```

The following output is obtained. Thus, the `htcat` command processes the data for each country and displays that data in the order the files were provided as arguments. In other words, the origin of this command is the UNIX command `cat`.

```
Brazil      1.92000000E+08  8.51487700E+06
China       1.35000000E+09  9.59696100E+06
Japan       1.28000000E+08  3.77847000E+05
Russia      1.43000000E+08  1.70982420E+07
USA         3.13000000E+08  9.62909100E+06
```

6.5 htcatts (Combination of H08 Format 1D Files)

This command can combine multiple H08 Format 1D ASCII files.

6.5.1 Order of arguments

```
% htcatts c0ascts c0ascts
```

The user provides any of 3H, DY, MO and YR at the end of the file name of the input H08 Format 1D ASCII file according to the time resolution of the file. In the current specifications, the user can specify up to 100 files to be combined.

6.5.2 Example of execution

Two H08 Format 1D ASCII files one having the monthly average temperature for Tokyo (139.5° East, 35.5° North) and the other for Sapporo (141.5° East, 43.5° North) for 1986 will be combined.

First, move to `${DIRSEMI}/data_asc/`.

```
% cd ${DIRSEMI}/data_asc/
```

In this directory, the `tokyo.txt` file is a monthly average temperature H08 Format 1D ASCII file for Tokyo in 1986. In addition, `sapporo.txt` is a monthly average temperature H08 Format 1D ASCII file for Sapporo.

Combine these two H08 Format 1D ASCII files using the `htcatts` command. Since both these files have monthly average temperatures in them, provide `MO` at the end of the input file names. Also, write the results of the output to `test_catts.txt` using a redirect (`>`).

```
% htcatts tokyo.txtMO sapporo.txtMO > test_catts.txt
```

The content of `test_catts.txt` output by the command statement above is as follows.

```

1986  1  0  2.75808990E+02  2.65368896E+02
1986  2  0  2.75908813E+02  2.65493713E+02
1986  3  0  2.79883789E+02  2.70843811E+02
1986  4  0  2.86058899E+02  2.78768799E+02
1986  5  0  2.89833801E+02  2.83143585E+02
1986  6  0  2.93333710E+02  2.88018799E+02
1986  7  0  2.96633911E+02  2.90693695E+02
1986  8  0  2.99108795E+02  2.94693695E+02
1986  9  0  2.95908813E+02  2.90018799E+02
1986 10  0  2.88883698E+02  2.81018707E+02
1986 11  0  2.84033997E+02  2.75368713E+02
1986 12  0  2.80283813E+02  2.69893585E+02

```

Try comparing the differences in test_catts.txt and tokyo_sapporo.txt here using the diff command. No differences between the content in the two files should be shown.

```
% diff test_catts.txt tokyo_sapporo.txt
```

6.6 htranktxt (Create Ranking Table)

This command rearranges the data in an H08 Format 1D ASCII file provided by the user into ascending order or descending order. In other words, this command carries out calculations similar to those for the htrank binary file processing system command on H08 Format 1D ASCII files.

6.6.1 Order of arguments

```
% htranktxt OPTION c0ascts i0yearmin i0yearmax
```

OPTION: select any of largest10, smallest10, incord and decord.

largest10: outputs data up to 10th largest starting with the largest.

smallest10: outputs data up to 10th smallest starting with the smallest.

incord: arranges all data in the input file in ascending order.

decord: arranges all data in the input file in descending order.

The user provides any of 3H, DY, MO and YR at the end of the file name of the input H08 Format 1D ASCII file according to the time resolution of the file.

6.6.2 Example of execution

This example outputs a table of rankings from a daily average temperature H08 Format 1D ASCII file (`${DIRSEMI}/data_asc/temp.bangkok.txt`) for 1986 in Bangkok.

First, move to `${DIRSEMI}/data_asc/`.

```
% cd ${DIRSEMI}/data_asc/
```

Next, output the temperature data from the highest to the 10th highest in `temp.bangkok.txt` using the `htranktxt` command with the `largest10` option. Moreover, since the input file has daily average data in it, provide `DY` at the end of the input file name.

```
% htranktxt largest10 temp.bangkok.txtDY 1986 1986
```

With the command statement above, the user obtains the following output. The output displays in order from the left year, month, day, rank and data.

1986	8	20	1	300.5875
1986	9	4	2	300.4642
1986	8	30	3	300.4246
1986	8	29	4	300.3256
1986	8	13	5	300.045
1986	8	6	6	300.0032
1986	8	27	7	299.9726
1986	8	14	8	299.9682
1986	8	21	9	299.9346
1986	8	28	10	299.9122

This time, output the temperature data from the smallest to the 10th smallest in `temp.bangkok.txt` using the `htranktxt` command with the `bottom10` option.

```
% htranktxt bottom10 temp.bangkok.txtDY 1986 1986
```

The user obtains the following output with the command statement above.

1986	2	16	1	272.4087
1986	1	5	2	273.1015
1986	1	22	3	273.4067
1986	1	27	4	273.4773
1986	1	9	5	273.4997
1986	1	10	6	273.6649
1986	1	6	7	273.6945
1986	1	26	8	273.7628
1986	2	25	9	273.7638
1986	1	12	10	273.9615

6.7 htlsmtxt (Least Squares)

This command can execute the least squares method on two H08 Format 1D ASCII files. The lsm in htlsmtxt is an abbreviation for least squares method. This command is convenient when comparing observed values and calculated values.

6.7.1 Order of arguments

```
% htlsmtxt c0ascts(X) c0ascts(Y) OPTION
```

OPTION: in $(Y) = a(X) + b$, the user selects which of a or b that he wants to know.

6.7.2 Example of execution

The monthly average temperature H08 Format 1D ASCII files (tokyo.txt, sapporo.txt) for Tokyo (139.5° East, 35.5° North) and Sapporo (141.5° East, 43.5° North) in 1986 will be compared by the least squares method.

First, move to `${DIRSEMI}/data_asc/`.

```
% cd ${DIRSEMI}/data_asc/
```

Then find a and b by using htlsmtxt, letting tokyo.txt be c0ascts(X) and sapporo.txt be c0ascts(Y) in $(Y) = a(X) + b$.

```
% htlsmtxt tokyo.txt sapporo.txt a
      1.25
% htlsmtxt tokyo.txt sapporo.txt b
     -79.66
```

Therefore, the following relational equation arises between the two files.

$$\boxed{\text{sapporo.txt}} = 1.25 \times \boxed{\text{tokyo.txt}} - 79.66$$

6.8 htmettxt (Statistical Calculations)

This command can find statistics such as the bias and correlation coefficient for two H08 Format 1D ASCII files provided by the user.

Caution: the user is to rewrite the program according to the application.

6.8.1 Order of arguments

```
% htmettxt OPTION c0ascts(sim) c0ascts(obs)
                                     i0yearmin i0yearmax
```

OPTION: specify options from the following. In the initial state, the results of calculations for all options are output no matter which option is selected.

bias: find bias

biasabs: find absolute value for bias.

cc: find correlation coefficient.

delay: find delay for peak value.

rmse: find root mean square error.

* The user provides any of 3H, DY, MO and YR at the end of the file name according to the time resolution of c0ascts (sim) and c0ascts (obs). The time resolutions for c0ascts (sim) and c0ascts (obs) must agree.

6.8.2 Example of execution

Find the correlation coefficient for calculated values in observed values for the flow rate of the Chao Phraya River.

First, move to `${DIRSEMI}/data_asc`.

```
% cd ${DIRSEMI}/data_asc
```

In this directory, temp.sim.txt is the file of calculated values for the Chao Phraya River flow rate and sample.obs.txt is the file of observed values. The units for the data in both files are mm/day.

Using the command htmettxt, find the bias between the calculated flow rates and observed

flow rates for the Chao Phraya River.

```
htmettxt bias temp.sim.txtMO sample.obs.txtMO 1986 1986
```

The results of the output are shown in the following.

```
5.69977403E-01    12
```

Chapter 7

netCDF File Processing System Commands

In this chapter, the commands for the netCDF file processing system will be described. The netCDF file system commands are group of commands for inserting data from H08 Format 2D binary files output into netCDF format files.

7.1 htcreatenc (Create New File)

This command creates a new netCDF format file. When the user wants to convert an H08 Format 2D binary file created by H08 into a netCDF format file, an empty netCDF format file is first created by htcreatenc, and then data is inserted by htputncts, which is in the next section.

7.1.1 Order of arguments

The arguments provided by the user for this command vary according to the time resolution for the netCDF format file being created.

* When the user creates a netCDF format file with daily units or three-hour units, a file is created for each month to prevent the file from becoming huge.

Annual units

```
% htcreatenc n0x n0y p0lonmin p0lonmax p0latmin p0latmax
                c0nc YR i0yearmin i0yearmax
```

Monthly units

```
% htcreatenc n0x n0y p0lonmin p0lonmax p0latmin p0latmax
                c0nc MO i0yearmin i0yearmax
```

Daily units

```
% htcreatenc n0x n0y p0lonmin p0lonmax p0latmin p0latmax
                c0nc DY i0year i0mon
```

Three-hour units

```
% htcreatenc n0x n0y p0lonmin p0lonmax p0latmin p0latmax
                c0nc 3H i0year i0mon
```

When the user inserts six-hour unit data, he provides an argument in which the 3H for inserting three-hour units is changed to 6H.

7.1.2 Example of execution

Creating a netCDF format file in which the time resolutions are empty.

Annual units

Create an empty netCDF format file for entering two sets of data, 1986 annual average and 1987 annual average.

```
% htcreatenc 360 180 -180 180 -90 90 temp.year.nc YR 1986 1987
```

Monthly units

Create an empty netCDF format file for entering a total of 24 pieces of data, the monthly averages for 1986 and the monthly averages for 1987.

```
% htcreatenc 360 180 -180 180 -90 90 temp.mon.nc MO 1986 1987
```

Daily units

Create an empty netCDF format file for entering for entering daily average data for January 1986, that is, 31 days of data. To prevent the file from becoming huge, the user should create a netCDF format file for each month.

```
% htcreatenc 360 180 -180 180 -90 90 temp.1986.01.nc DY 1986 01
```

Three-hour units

Create an empty netCDF format file for entering for entering data every three hours for January 1986, that is, 248 (31 days worth of 1 day = 3 hours \times 8 pieces of data, so $8 \times 31 = 248$). To prevent the file from becoming huge, the user should create a netCDF format file for each month.

```
% htcreatenc 360 180 -180 180 -90 90 temp.1986.01.nc 3H 1986 01
```

7.2 htputncts (Insert Data)

This command inserts data from an H08 Format 2D binary file into an empty netCDF format file created by htcreatenc in the previous section.

7.2.1 Order of arguments

The arguments provided by the user for this command vary according to the time resolution for the H08 Format 2D binary file being inserted.

```
var_name: variable name
var_units: units
var_long_name: detailed file name
```

Annual units

```
% htputncts n0l n0x n0y c0l2x c0l2y c0bints i0year i0mon
      c0nc var_name var_units var_long_name YR i0yearmin
```

Monthly units

```
% htputncts n0l n0x n0y c0l2x c0l2y c0bints i0year i0mon
      c0nc var_name var_units var_long_name MO i0yearmin
```

Daily units

```
% htputncts n0l n0x n0y c0l2x c0l2y c0bints i0year i0mon
      c0nc var_name var_units var_long_name DY
```

Three-hour units

```
% htputncts n0l n0x n0y c0l2x c0l2y c0bints i0year i0mon
      c0nc var_name var_units var_long_name 3H
```

When the user inserts six-hour unit data, he provides an argument in which the 3H for inserting three-hour units is changed to 6H.

7.3.2 Example of execution

Insert the temperature data²⁵ for 1986 through 1987 into the netCDF format file created in

²⁵ See the method for preparing climate data in the H08 Manual User's Edition 7.24 acquisition and editing of files after 1987.

the previous section. Since there is a lot of data to be inserted, try defining it in a shell script.

Annual units

```
#!/bin/sh
source ~/.bashrc
L2X=~ /H08/map/dat/l2x_l2y_/l2x.one.txt
L2Y=~ /H08/map/dat/l2x_l2y_/l2y.one.txt
TAIR=~ /H08/met/dat/Tair____/GSW2B1b_.one
# year 1986-87
NC=temp.year.nc
htcreatenc 360 180 -180 180 -90 90 $NC YR 1986 1987
  for YEAR in 1986 1987; do
    for MON in 00; do
      htputncts 64800 360 180 $L2X $L2Y ${TAIR}YR $YEAR $MON $NC tair K
    Air_Temperature_in_K YR 1986
    done
  done
```

Monthly units

```
# mon 1986-87
NC=temp.mon.nc
htcreatenc 360 180 -180 180 -90 90 $NC MO 1986 1987
  for YEAR in 1986 1987; do
    for MON in 01 02 03 04 05 06 07 08 09 10 11 12; do
      htputncts 64800 360 180 $L2X $L2Y ${TAIR}MO $YEAR $MON $NC tair K
    Air_Temperature_in_K MO 1986
    done
  done
```

Daily units

```

# day 1986-87
for YEAR in 1986 1987; do
  for MON in 01 02 03 04 05 06 07 08 09 10 11 12; do
    NC=temp.${YEAR}.${MON}.nc
    htcreatenc 360 180 -180 180 -90 90 $NC DY $YEAR $MON
    htputncts 64800 360 180 $L2X $L2Y ${TAIR}DY $YEAR $MON $NC tair K
Air_Temperature_in_K DY
  done
done

```

Three-hour units

```

# 3 hour 1986-1987
for YEAR in 1986 1987; do
  for MON in 01 02 03 04 05 06 07 08 09 10 11 12; do
    NC=temp.${YEAR}.${MON}.3H.nc
    htcreatenc 360 180 -180 180 -90 90 $NC 3H $YEAR $MON
    htputncts 64800 360 180 $L2X $L2Y ${TAIR}3H $YEAR $MON $NC tair K
Air_Temperature_in_K 3H
  done
done

```

- Column 1

-- Checking netCDF format file: GrADS

This column gives an introduction to using GrADS for checking whether the netCDF format file created by htcreatenc and htputncts has been created normally. GrADS is an abbreviation of Grid Analysis and Display System and is a drawing tool for multidimensional grid data. GrADS is used worldwide by meteorologists. GrADS handles Fortran direct access format, netCDF format and GRIB format files. Launch GrADS.

```
% grads
```

The version information is displayed and finally
Landscape mode? ('n' for portrait):

is displayed, so enter "n" and press "Enter."

```
ga=>
```

is displayed, and a drawing window is displayed at the same time. This is the GrADS command input mode. To open a netCDF format file, the user inputs the file name after an sdfopen command. Use the file containing monthly unit data for 1986 to 1987 that was created in the execution example in 6.2 here.

```
ga=> sdfopen temp.mon.nc
```

Next, specify the coloring method.

```
ga=> set gxout shaded
```

Next, set the display time (t). Since the file being used here contains 24 hours of data, input the following.

```
ga=> set t 1 24
```

Finally, display it with the d (abbreviated form for "display") command. Since the name of the variable for the file being used here is tair, input the following.

```
ga=> d tair
```

An animation that changes in 24 stages should be displayed.

When the user wants to stop displaying the image that is being displayed, he uses the c (abbreviated form for "clear") command.

```
ga=> c
```

Finally, use the quit command to exit GrADS.

```
ga=> quit
```

Appendix 1

Abbreviation Method for Arguments

In this appendix, a method for abbreviating the arguments assigned in H08 Analysis Tools command groups will be described.

A.1 Abbreviating Arguments Specifying Spatial Resolution

In H08 Analysis Tools, the user provides many arguments, so it takes time and effort, and mistakes in input often arise. Therefore, we can consider how to abbreviate the arguments. For example, when a map with a $1^\circ \times 1^\circ$ spatial resolution is drawn by `htdraw`, arguments such as

```
% htdraw 64800 360 180 ~/H08/map/dat/12x_12y_/12x.one.txt
      ~/H08/map/dat/12x_12y_/12y.one.txt -180 180 -90 90
      lndmsk.GSWP2.one lndmsk.cpt lndmsk.eps landmask
```

are normally provided. The underlined part is a command group that specifies a spatial resolution of $1^\circ \times 1^\circ$ for the H08 Format 2D binary file. Here, the following command alias is established for the underlined part.

```
% alias one2eps=' htdraw 64800 360 180
      ~/H08/map/dat/12x_12y_/12x.one.txt ~/H08/map/dat/12x_12y_/12y.one.txt
      -180 180 -90 90'
```

Enter the following.

```
% one2eps lndmsk.GSWP2.one lndmsk.cpt lndmsk.eps landmask
```

With the command statement above, the same output is obtained as when the user created the land and sea map with a spatial resolution of $1^\circ \times 1^\circ$ using `htdraw`. Thus, the user can abbreviate arguments using a command alias. H08 has similar settings for almost all of the binary file processing system commands in this manual.²⁶ Table A1-1 gives list of them.

²⁶ See `.../H08/adm/sample.bash`.

Table A1-1 List of Abbreviated Commands

Section	Command	Spatial resolution		
		1° × 1°	0.5° × 0.5°	Chao Phraya River basin 5' × 5'
4.3	htdraw	one2eps	hlf2eps	cp52eps
5.1	htcreate	createone	createhlf	createcp5
5.2	htformat	one2asc	hlf2asc	cp52asc
		one2xyz	hlf2xyz	cp52xyz
		asc2one	asc2hlf	asc2cp5
5.3	htarray	xyz2one	xyz2hlf	xyz2cp5
		shiftone	shifthlf	shiftcp5
		upsidedownone	upsidedownhlf	upsidedowncp5
5.4	htmath	addone	addhlf	addcp5
		subone	subhlf	subcp5
		mulone, proone	mulhlf, prohlf	mulcp5, procp5
5.5	htstat	divone, ratone	divhlf, rathlf	divcp5, ratcp5
		maxone	maxhlf	maxcp5
		minone	minhlf	mincp5
		sumone	sumhlf	sumcp5
5.6	httime	aveone	avehlf	avecp5
		mon2yearone	mon2yearhlf	mon2yearcp5
5.7	htmean	meanone	meanhlf	meancp5
5.8	htid	idone	ldhlf	idcp5
5.9	htedit	editone	dedithlf	editcp5
5.10	htpoint	pointone	pointhlf	pointcp5
5.11	htpointts	punchone	punchhlf	punchcp5
5.12	htmask	maskone	maskhlf	maskcp5
5.13	htmaskrplc	maskrplcone	maskrplchlf	maskrplccp5