Appendix I - NetCDF and ASCII File Processing Tools

The collection of CGCMII scenario data is accompanied by a collection of small programs that will assist the user in manipulating data in the NetCDF files, and extracting them to ASCII format files. Documentation on each of these programs are provided in this section.

Common features and problems

All of these C/C++ programs have been tested for compilation and operation on Solaris 2.6 and Linux. They should be easily portable to other Unix environments. All the executables recognize command line switches '-h' and '--help' and display a complete usage text in response (see detailed descriptions of individual programs below).

A subset of the input file may be processed by providing a range of years, or time steps, depending on the program.

ANUSPLIN format files do not include date information. Programs using ANUSPLIN format files for input, require that the year range of the data be encoded in the file name, as eight or nine characters, immediately left of the extension. Examples include:

pcp_1975-1984.anu
tmin_19002100.anu

Some grid arrays are dynamically allocated after determining the required dimensions, but most are defined at compile time by the macros XMAX and YMAX, which can be found in **functions.c** and each program that requires them.

Output files overwrite any existing file of the same name, without warning.

Almost all the programs include the **functions.c** module, and all such programs must be compiled with the '-lm' switch to link the math library because some of the functions in **functions.c** require that library.

All programs that process NetCDF format files must be compiled with the '-lnetcdf' switch to link the NetCDF library. All such programs include the **nc_functions.c** module.

To build the executable use the following command line:

```
%gcc foo.c -o foo [-g] -lm
```

'-g' adds debugging information (optional). '-lm' links the math library. '-o foo' names the executable as 'foo'.

or to compile and link a program which uses the NetCDF library:

```
%g++ nc_foo.c -o nc_foo [-g] -lnetcdf
```

where '-lnetcdf' links the NetCDF library.

Description of Executables

The following symbols are used to identify command line options:

[...]	Enclosed argument is optional.
|	Exclusive "or"(i.e., choose only one of the alternative arguments)

Other common terms:

Note that in all NetCDF data files, the first timestep is numbered "1" not "0".

I.1. ANUSPLIN To CCCma Format Converter (anu2ccc)

Description:

Reads an ANUSPLIN format file, and writes it in the format used for distributing files from the CCCma website (referred to from hereon as 'CCCma format), or the similar format used by University of Quebec à Montreal, (hence forth referred to as 'UQAM format'). It is assumed that the data in the ANUSPLIN input file start at the north-west corner of the grid and end at the south-east corner. The output data are written starting with the south-west corner of the grid, and ending with the north-east corner.

Usage:

./anu2ccc [--ccc | -c | --uqam | -u] <data file> [<start year> <end year>]

or:

./anu2ccc [-h | --help]

Switches: at most one of:

--ccc	To write output in CCCma format (1P6e12.5). This is the default.
--uqam	To write output in a format very similar to the CCCma format. The major difference is the increased precision (1P6E22.15).
-c	Same as '--ccc'.
-u	Same as '--uqam'.
--help	Display detailed program usage information.
-h	Same as '--help'.

Additional Notes:

<data file> is the set of ANUSPLIN format data to be converted to CCCma or UQAM format.

Output data are flipped vertically relative to the organization of the input data.

<start year> and <end year> are optional arguments to set the range for which output will be written. If the range is not specified, it defaults to 0 to 9999. Any data in <data file> falling outside this range will not be converted.

It is assumed that the name of the input ANUSPLIN file is of the general format '[c..._var_c...]yyyy[-]yyyy[.c...]', where the last 8 or 9 characters, before the extension, represent the year range of the data. Additionally, the name of the climate variable (FSS, PCP, STMN, STMX, or U) must be present in the input file name, and preceded and/or succeeded by an underscore '_'.

Output files are named after the input file, except the year range may be updated, and a file extension may be added, or replaced.

Author: Fred Woslyng, November, 2000

Possible Improvements:
(1) Only about half a dozen CCCma climate variable names are known to the function get_climate_var(), so until the function is updated with additional variable names, only those half dozen climate variables can be processed. An alternative solution would be to pass the climate variable name as a command line argument.

I.2. ANUSPLIN Ratio Cleaner (anu_clean)

Description:
Reads an ANUSPLIN format file of ratio data, replaces any instances of negative values in the data with zeroes (0.0), reports the number of changes, and writes the output in ANUSPLIN format. It optionally processes a subset of the input file if a year range is included on the command line.

Usage:

./anu_clean [-x] <data file> [<start year> <end year>]

or

./anu_clean [-h | --help]

Switch:

-x	Write the ANUSPLIN output in 12.5E format rather than the default 8.2F.
--help	Display detailed program usage information.
-h	Same as '--help'.

Additional Notes:
<data file> is the set of ANUSPLIN data to be cleaned of negative values.
<start year> and <end year> are optional arguments to set the range for which output will be written. If the range is not specified, it defaults to 0 to 9999. Any data in <data file> falling outside this range will not be processed.
It is assumed that the name of the input file is '[c...]yyyy[-]yyyy[.c...]', where the last 8 or 9 characters, before the extension, represent the year range of the data. The output file is named after the input file, except the year range may be updated, and a file extension may be added or replaced.

Author: Fred Woslyng, February, 2001.

Possible Improvements:
(1) Pass the replacement value as an optional command line argument, with a default value of zero, rather than having the replacement value fixed.

I.3. ANUSPLIN Mask (anu_mask)

Description: Reads an ANUSPLIN format data file and a mask file, and applies the mask to the ANUSPLIN data, across the specified subset of years. The output is written in ANUSPLIN format. The mask file may be either a comma separated value (csv) file, or an ANUSPLIN format file. If the latter, then the elevations, latitudes, and longitudes are read from both input files and compared to determine if they represent the same geographic points. The results of the comparison are reported, but the program continues the merge process regardless of the outcome of the comparison. Optionally, the latitudes, longitudes and elevations from ANUSPLIN format input may be output as individual csv format files. The mask elevation values are written as 0s and 1s representing masked and unmasked data cells, respectively.

Usage:

./anu_mask [--mask<mask>] [--nodata<no_data>] [-a] [-x] <mask file> <data file> [<start year> <end year>]

or:

./anu_mask [-h | --help]

Switch: some combination of:

--mask<mask> Indicates the value in the <mask file> that corresponds with points that are to be changed to <no_data> in the <data file>. The default is 0.0.

--nodata<no_data> Sets the NO_DATA value that will be written to cells that are masked. The default is -9999.

-a Write the elevations, longitudes and latitudes from the <data file> as individual csv files. The files are suitable as input to programs **ccc_norm**2anu and **ccc_ts2anu**. If the <mask file> is ANUSPLIN format, then like files are written, except that the elevation (mask) data is written as a series of 0s (masked) and 1s (unmasked).

-x Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than the default 8.2f format.

--help Display detailed program usage information.

-h Same as '--help'.

Additional Notes:

<mask file> is an ANUSPLIN format file (the monthly data is optional), with one grid, where the elevation field is used as a mask. Alternatively, a comma separated values (csv) file could be used. The csv file would contain only the mask values, and must have the extension '.csv'.

<data file> is the set of ANUSPLIN formatted data to be masked, to produce the ANUSPLIN formatted output data.

<start year> and <end year> are optional arguments to set the range for which output will be written. If the range is not specified, it defaults to 0 to 9999. Any data in <data file> that is outside this range will not be processed.

It is assumed that the name of the input file is '[c...]yyyy[-]yyyy[.c...]', where the last 8 or 9 characters, before the file extension, represent the year range of the data. The output files are named after the input files, except the year range may be updated, and a file extension is added or replaced.

Author: Fred Woslyng, December 2000

I.4. ANUSPLIN Merge (anu_merge)

Description:

Reads an ANUSPLIN format means file, and an ANUSPLIN format difference (or ratio) data file and adds (or multiplies) corresponding grid elements to produce absolute, rather than relative, climate data. The output is written in ANUSPLIN format. The elevation, latitudes, and longitudes are read from both input files and compared to determine if they represent the same geographic points. The results of the comparison are reported, but the program continues the merge process regardless of the outcome of the comparison. When merging (adding or multiplying) corresponding cells, if the value of either cell is the NO_DATA value, then the value of that cell in the output is also NO_DATA.

Usage:

./anu_merge [-[p | s] [x] [y]] <mean file> <data file> [<start year> <end year>]

or:

./anu_merge [-h | --help]

Switches: some combination of:

-p	Calculate the product of corresponding grid elements of <mean file> and <data file>.
-s	Calculate the sum of corresponding grid elements of <mean file> and <data file>.
-x	Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than the default 8.2f format.
-y	Merge corresponding years, when multiple years (grids) of data exist in the <mean file>. This switch has very limited application.
--help	Display detailed program usage information.
-h	Same as '--help'.

Additional Notes:

Multiple switches must be coded as '-sxy', rather than '-s -x -y'.

The 'p' and 's' switches are mutually exclusive. If neither switch 'p' nor 's' is specified, the program attempts to determine the operation to perform by examining file name <data file>.

The 'y' switch was created to merge mean sea level pressure data with specific humidity data to produce vapor pressure. (The ETA constant must be applied using program anu_units.) The two input files must have identical year ranges, and the same number of grids.

<mean file> is typically a single grid containing one year of mean data in ANUSPLIN format. However, it may contain multiple years (grids) of data to be merged with an identical number of grids in <data file>, by using '-y'.

<data file> is the set of ANUSPLIN data to be multiplied by the mean, or added to the mean, to produce the output data. The output value of an element is NO_DATA (-9999), if either input element is NO_DATA.

<start year> and <end year> are optional arguments to set the range for which output will be written. If the range is not specified, it defaults to 0 to 9999. Any data in <data file> that is outside this range will not be processed.

It is assumed that the name of the <data file> is '[c...]yyyy[-]yyyy[.c...]', where the last 8 or 9 characters, before the extension, represent the year range of the data. The same naming convention applies the the <mean file> if the '-y' switch is used. The output file is named after the <data file>, except the year range may be updated, and a file extension may be added or replaced.

Author: Fred Woslyng, December, 2000

Possible Improvements:

(1) Pass the NO_DATA value as a command line argument, so that different NO_DATA values can be easily supported.

I.5. Units Converter For ANUSPLIN Format Data (anu_units)

Description:

Reads an ANUSPLIN format file, performs a units conversion on the data values, and writes the output in ANUSPLIN format. The conversion factor may be supplied by the user, or a predefined conversion may be selected. If a year range is specified, then only that subset of the input file is converted and output.

Usage:

./anu_units -m<conversion factor> | --mj_per_day_to_watt | --per_month_to_per_second [-x] <data file> [<start year> <end year>]

or:

./anu_units [-h | --help]

Switches:

-m <conversion factor> Convert the units of the data set by multiplying by the value <conversion factor>.

--mj_per_day_to_watt Convert MJoules/day to Watts.

--per_month_to_per_second Convert time units from 'per month' to 'per second'.

-x Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than default 8.2f format.

--help Display detailed program usage information.

-h Same as '--help'.

Additional Notes:

Multiple switches must be entered as '-x -m3.14' rather than '-xm3.14'.

<data file> contains the set of ANUSPLIN format data to be converted.

<start year> and <end year> are optional arguments to set the range for which output will be written. If the range is not specified, it defaults to 0 to 9999. Any data in <data file> that is outside this range will not be processed.

It is assumed that the name of the input file is '[c...]yyyy[-]yyyy[.c...]', where the last 8 or 9 characters, before the extension, represent the year range of the data. The output file is named after the input file, except the year range may be updated, and a file extension may be added or replaced.

Author: Fred Woslyng, February 2001

I.6. CCCma To CSV Converter (ccc2csv)

Description:

Reads a CCCma format file (from the CCCma web site) and optionally flips the input grid vertically, before it writes comma separated values (csv) format output. The program is useful for converting CCCma orography data to the elevs.csv elevation data file used by the **ccc_norm2anu** and **ccc_ts2anu** programs.

Usage:

./ccc2csv [-f] <data file>

or:

./ccc2csv [-h | --help]

Switch:

-f Flip each input data grid vertically before any other processing.

--help Display detailed program usage information.

-h Same as '--help'.

Additional Notes:

<data file> is the file of data in CCCma format to be converted.

Author: Fred Woslyng, 22 January, 2001.

I.7. CCCma Model Data Normalizer (ccc_norm2anu)

Description: Reads a CCCma model data file, calculates means across the specified subset of years, and either calculates differences or ratios between the whole file, (or the subset of years requested), and the mean values. It optionally calculates the standard deviations for the years the means are based on, and writes the means, standard deviations, and the input file, (or subset), to disk. All output is in ANUSPLIN format. Elevation, latitude, and longitude grids are required to produce the ANUSPLIN output, with names "elevs.csv", "lats.csv" and "lons.csv", respectively.

Usage:

./ccc_norm2anu [-[d | r | w] [f] [i] [m] [s] [x]] <data file> <mean start year> <mean end year> [<start year> <end year>]]

or:

./ccc_norm2anu [-h | --help]

Switches: some combination of:

-d	Calculate the difference of the values, and the means, of <data file>, and write the output in ANUSPLIN format.
-r	Calculate the ratio of the values, and the means, of <data file>, and write the output in ANUSPLIN format.
-w	Convert wind from 2m to 10m height, and then perform '-r'.
-f	Do NOT flip each input data grid vertically before any other processing.
-i	Write input data to disk, in ANUSPLIN format.
-m	Write the means to disk, in ANUSPLIN format.
-s	Write the standard deviations of the means to disk, in ANUSPLIN format.
-x	Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than the default 8.2f format.
--help	Display detailed program usage information.
-h	Same as '--help'.

Additional Notes:

Multiple switches must be coded as '-rms', rather than '-r -m -s'.

The 'd', 'r' and 'w' switches are mutually exclusive.

<data file> is the CCCma data set to be processed.

<mean start year> and <mean end year> are optional arguments to set the range over which the means are calculated. If the range is not specified, it defaults to 1961 to 1990.

<start year> and <end year> are optional arguments to set the range for which output will be written. If the range is not specified, it defaults to 0 to 9999. Any data in <data file> that fall outside this range will not be processed.

Output files are named after <data file>. A year range may be appended to the file name, or updated if it already exists, and a file extension may be added or replaced.

Authors: Erik Johnson, August 2000 Fred Woslyng, November 2000

Possible Improvements:

(1) The header data that start each grid could be used to verify the month value calculated by the program.

I.8. CCCma Model Data Time Slice Normalizer (ccc_ts2anu)

Description: Reads two CCCma data files, for specific time slices (such as 1975-1984 and 2040-2049) and outputs pairs of files of either ratios or differences, based on the means of the first input file. It optionally calculates means and standard deviations for both files. All output is written in ANUSPLIN format. By default, the input data is flipped vertically before processing it. Separate files containing grids of elevation, latitude, and longitude data are required to produce the ANUSPLIN output. The program is similar to **ccc_norm2anu**.

Usage:

./ccc_ts2anu [-[d | r | w] [f] [i] [m] [s] [x]] <data file 1>

or:

./ccc_ts2anu [-h | --help]

Switches: some combination of:

-d	Calculate the difference of the values of both input files, and the means of <data file 1>, and write the output in ANUSPLIN format.
-r	Calculate the ratio of the values of both input files, and the means of <data file 1>, and write the output in ANUSPLIN format.
-w	Convert wind from 2m to 10m height, and then perform '-r'.
-f	Do NOT flip each input data grid vertically before any other processing.
-i	Write input data to disk, in ANUSPLIN format.
-m	Write the means to disk, in ANUSPLIN format.
-s	Write the standard deviations of the means to disk, in ANUSPLIN format.
-x	Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than the default 8.2f format.
--help	Display detailed program usage information.
-h	Same as '--help'.

Additional Notes:
Multiple switches must be coded as '-rms', rather than '-r -m -s'.
The 'd', 'r' and 'w' switches are mutually exclusive.
<data file 1> is the first CCCma data set to be processed. The second input data set must have the same file name as <data file 1>, except the first character is replaced by a '2'.
Output files are named after the input files, but with a file extension added or replaced.

Authors: Erik Johnson, August 2000; Fred Woslyng, November 2000

Possible Improvements:
(1) The header data that start each grid could be used to verify the month value calculated by the program.

I.9. Catenate Monthly Data Files (cat_files)

Description:
Concatenates a set of monthly files, in chronological order, into a single file. The file names must conform to a prescribed <format>. If a <header size> is specified, that number of lines is discarded from the header of each file, starting with the second file. Discarded header lines are compared to ensure that corresponding lines are identical.

Usage:

./cat_files <base file name> <start year> <end year> [<header size> [<format>]]

or

./cat_files [-h | --help]

Switch:

--help Display detailed program usage information.
-h Same as '--help'.

Additional Notes:

<base file name> is the root name of all the monthly input files. See the explanation of the <format> argument for additional information.

<start year> and <end year> define the range of file names that will be concatenated. There is no default range. Both <start year> and <end year> must be non-negative integers, and <start year> must not be greater than <end year>.

<header size> is an optional argument denoting the number of lines to discard from the header of each input file, starting with the second file. The number must be in the range 0 to 1000. The default is 0 lines, which results in the entire contents of each file being copied to output. If a non-zero <header size> is specified, that number of lines is discarded from the start of each file, starting with the second input file. Thus, the concatenated output file may be created with, or without, embedded headers. Discarded header lines are compared to ensure that corresponding lines, from different files, are identical. If they are not, a warning is displayed, but the concatenation continues.

<format> is an optional argument to be used as a template to form the name of each file in the set. The default format is '%s%i_%i.grd', where '%s' is replaced by <base file name>, the first '%i' is replaced by the year, and the second '%i' is replaced by the month (1 to 12). The order of the <base file name>, year and month components cannot be changed, but otherwise, the template may be altered by adding or deleting characters to match the names of the existing set of input files. Add or delete '%' characters at your own risk because the <format> is interpreted by the C 'sprintf' command.

Author: Fred Woslyng, January 2001

Possible Improvements:

(1)Modify the program to concatenate yearly data files.

I.10. ANUSPLIN Data Normalizer (anu_norm)

Description:

Reads an ANUSPLIN format file, calculates means across the specified subset of years, and either calculates differences or ratios between the whole file, (or the subset of years requested), and the mean values. Optionally, calculates the standard deviations for the years the means are based on, and writes the means and standard deviations to disk. All input and output is in ANUSPLIN format. The elevation, latitudes, and longitudes are read from the input data file. The program is very similar to ccc_norm2anu.

Usage:

./anu_norm [-[d | r | w] [m] [s] [x]] <data file> [<mean start year> <mean end year> [<start year> <end year>]]

or:

./anu_norm [-h | --help]

Switches: some combination of:

-d	Calculate the difference of the values, and the means, of <data file>, and write the output in ANUSPLIN format.
-r	Calculate the ratio of the values, and the means, of <data file>, and write the output in ANUSPLIN format.
-w	Convert wind from 2m to 10m height, and then perform '-r'.
-m	Write the means to disk, in ANUSPLIN format.
-s	Write the standard deviations of the means to disk, in ANUSPLIN format.
-x	Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than the default 8.2f format.
--help	Display detailed program usage information.
-h	Same as '--help'.

Additional Notes:
Multiple switches must be coded as '-rms', rather than '-r -m -s'.
The 'd', 'r' and 'w' switches are mutually exclusive.
<data file> is the set of ANUSPLIN data to be processed.
<mean start year> and <mean end year> are optional arguments to set the range over which the means are calculated. If the range is not specified, it defaults to 1961 to 1990.
<start year> and <end year> are optional arguments to set the range for which output will be written. If the range is not specified, it defaults to 0 to 9999. Any data values in <data file> outside this range will not be processed.
It is assumed that the name of the input file is '[c...]yyyy[-]yyyy[.c...]', where the last 8 or 9 characters, before the extension, represent the year range of the data. The output files are named after <data file>, except that the year range may be updated, and a file extension is appended.

Author: Fred Woslyng, December 2000.

Possible Improvements:
(1) Dynamically allocate the grid arrays based on the number of rows and columns in the grid, as specified in the header record of each block of data.

I.11. ASCII-Grid To NetCDF Conversion (asg2nc)

Description:
Converts an ASCII-grid (.asg) format file to NetCDF (.nc) format. The data type of the output may be specified, along with optional scale and offset translation, which, depending on the input, may produce a smaller NetCDF output file, without any loss of precision. The input data are expected to be complete years (12 grids per year) of monthly data.
The NetCDF variable that is created is a function of time, level, latitude, and longitude, and NetCDF dimensions and variables, with those names, are created and populated with positive integer values.

Usage:

./asg2nc [-i<size> | -f<size>] [[[-s<scale>] [-o<offset>]] | [-n]] [-r] [-h<header file>] <input file> [<variable name>] [<year start> <year end>]

or:

./asg2nc [-h | --help]

Switches: some combination of:

-i<size> Convert the asg data to NetCDF signed integer format, where <size> is one of '1', '2' or '4', corresponding to 1, 2 or 4 bytes per data element.

-f<size> Convert the asg data to NetCDF decimal format, where <size> is '4' or '8', corresponding to 4 or 8 bytes per data element.

-s<scale> Scale the asg data by factor <scale>, before storing as NetCDF format. The asg data is divided by the scale factor, after any <offset> is subtracted.

-o<offset> Offset the asg data by amount <offset>, before storing as NetCDF format. The offset is subtracted from the asg data before any <scale> is applied.

-n Disable the integer storage optimization that is performed by default. Equivalent to '-s1.0 -o0.0'

-r Replace any negative ratio values with zero (0.0) before storing as NetCDF format.

-h<header> The name of a file containing asg header records, or NetCDF header (attribute) records. The default is <input file>, but with an '.hdr' extension. Values in the <header> file take precedence over values in the <input file> header.

--help Display detailed program usage information.

-h Same as '--help'.

Additional Notes:

In all NetCDF data files, the first timestep is numbered "1" not "0".

By default, decimal data (double and float) are converted to 4-byte decimal values (-f4 -o0.0 -s1.0), and integer data are converted to the smallest integer data type that does not cause a loss of precision, by applying only an <offset>, if necessary.

Multiple switches must be represented as '-i2 -s0.01 -o0.0 -r' rather than as '-i2s0.01o0.0r'. The order of switches is irrelevant. Switches 'i' and 'f' are mutually exclusive, as are 'n' and either of 's' or 'o'.

<input file> is the name of the ASCII grid (asg) input file of monthly data, starting with January, with exactly 12 grids per year.

Optional argument <variable name> is the name of the variable to be read by IBIS (subroutine readit). If <variable name> is not specified, it defaults to 'grid'.

Optional arguments <year start> and <year end> set the range for which data will be written to the output file. If the range is not specified, the default is 0 to 9999.

Authors: Ryan Sherrington, 1999-2000; rewritten by Fred Woslyng, December 2000.

Work To Finish:

(1) The function set_nc_nodata_int() which sets the NODATA value for integer data, is not called from all locations where the NODATA value is set. Thus, depending on which arguments are supplied on the command line, the NODATA value may be set to the minimum valid value for the specified data type (for example -32767), rather than more acceptable values like -9999.

I.12. NetCDF To ASCII-Grid Conversion (nc2asg)

Description:
Converts a specific variable from a NetCDF input file to ASCII-grid (.asg) format. The NetCDF variable read in is expected to be a function of time, level, latitude, and longitude. The NetCDF dimensions and variables, with names "time", "level", "latitude" and "longitude", are expected to exist.

Usage:

./nc2asg [-s<start>] [-e<end>] [-h<header file>] [-v<variable>] [-l<level>] [--split] [--asg | -a | --big | -b] [--nosummary] <nc file>

or

./nc2asg --summary <nc file>

or

./nc2asg [-h | --help]

Switches: some combination of:

-s<start> The lower limit of the range of time steps to convert. The default is 1.

-e<end> The upper limit of the range of time steps to convert. The default is <start> if it was specified, otherwise it is 999999.

-h<header file> The name of a file containing ASCII-grid header records. The default is <nc file>, but with an '.hdr' extension. The keywords 'ncols', 'nrows', 'xllcorner', 'yllcorner', 'cellsize', and 'nodata_value' are recognized, and their values are (mostly) written to the output file. All other records are ignored. Values for 'ncols' and 'nrows' are set by the program. If a 'nodata_value' value is not provided in the header file, the default (-9999) is used.

-v<variable> The name of the 3-D or 4-D NetCDF variable to process. The default is any variable with a rank of 3 or more.

-l<level> The value of the NetCDF level variable, which must be an integer greater than or equal to 1. The default is 1.

--split Write each output ASCII-grid to a separate file. The default is to write all output grids to a single file, with a single header at the start of the file.

--asg Write the output in ASCII-grid format.

-a The same as '--asg'.

--big Write the output in binary-grid format. NOT IMPLEMENTED.

-b Same as '--big'.

--summary Display a summary of the NetCDF dataset contents. No aggregates are calculated, only the summary is displayed.

--nosummary Disable the default display of the NetCDF dataset summary.

--help Display detailed program usage information.

-h Same as '--help'.

Additional Notes:
<nc file> is the name of the NetCDF input file.
In all NetCDF data files, the first timestep is numbered "1" not "0".
Output file names are based on <nc file>, but with the last file extension removed, and '_####.asg' appended, where '####' is the number of the input time step for the first grid in the ASCII output file.

Authors: Ryan Sherrington, 1999-2000; Fred Woslyng, February-March 2001.

Possible Improvements:
(1) Put binary grid format output code back into program.

I.13. NetCDF Aggregate Grid Cells (nc_aggregate)

Description:
Calculate spatial arithmetic averages by aggregating grid cells from NetCDF format input data, producing ASCII-grid format output.
The NetCDF variable read in is expected to be a function of time, level, latitude, and longitude. The NetCDF dimensions and variables, with names "time", "level", "latitude" and "longitude", are expected to exist.
Usage:

./nc_aggregate [-s<start>] [-e<end>] [-h<header file>] [-v<variable>] [-l<level>] [--split] [--nosummary] <nc file> <cell size>

or

./nc_aggregate [-s<start>] [-e<end>] [-h<header file>] [-v<variable>] [-l<level>] [--split] [--nosummary] -a <nc file> or /nc_aggregate --summary <nc file>

or

./nc_aggregate [-h | --help]

Switches: some combination of:

-s<start> The lower limit of the range of time steps to aggregate. The default is 1.
-e<end> The upper limit of the range of time steps to aggregate. The default is 999999.
-h<header file> The name of a file containing ASCII-grid header records. The default is <nc file>, but with an '.hdr' extension. The keywords 'ncols', 'nrows', 'xllcorner', 'yllcorner', 'cellsize', and 'nodata_value' are recognized, and their values are (mostly) written to the output file. All other records are ignored. Values for 'ncols' and 'nrows' are set by the program. If a 'nodata_value' value is not provided in the header file, the default (-9999) is used.
-v<variable> The name of the 3-D or 4-D NetCDF variable to process. The default is any variable with a rank of 3 or more.
-l<level> The value of the NetCDF level variable, which must be an integer greater than or equal to 1. The default is 1.
--split Write each output ASCII-grid to a separate file. The default is to write all output grids to a single file, with a single header at the start of the file.
-a Aggregate all the cells in the grid to a single point. Switch '-a' and <cell size> are mutually exclusive.
--summary Display a summary of the NetCDF dataset contents. No aggregates are calculated, only the summary is displayed.
--nosummary Disable the default display of the NetCDF dataset summary.
--help Display detailed program usage information.
-h Same as '--help'.

Additional Notes:
<nc file> is the name of the NetCDF input file.
In all NetCDF data files, the first timestep is numbered "1" not "0".
<cell size> is the number of grid cells to aggregate into a single cell. A value of 5, for example, would aggregate an area 5 cells by 5 cells. The value must divide evenly into both

dimensions of the grid. <cell size> and switch '-a' are mutually exclusive, and one or the other is mandatory.

Output file names are based on <nc file>, but with the last file extension removed, and '_####.asg' appended, where '####' is the number of the input time step that is the first grid in the ASCII output file.

Author: Fred Woslyng, February 2001 (some code taken from Ryan Sherrington's nc2asg.c).

Possible Improvements:
(1) Check whether the output files already exist, before overwriting them.

I.14. NetCDF Temporal Average (nc_average)

Description:
Calculate temporal arithmetic averages from NetCDF format input data, producing ASCII-grid format output. Consecutive (moving average) and non-consecutive time steps (e.g., all Julys in a time series), or sets of time steps (e.g., all three winter months over a 30-year period) may be averaged, producing single, or multiple ASCII-grids, output to a single file, or separate files (one grid per file). This is a very flexible and powerful utility: please read all the instructions carefully! The NetCDF variable that is read is expected to be a function of time, level, latitude, and longitude, and NetCDF dimensions and variables, with those names, are expected to exist.

Usage:

./nc_average [-s<start>] [-e<end>] [-c<consecutive>] [-i<increment>] [-r<repeat>] [-n<next incr>] [-g<grids out>] [-h<header file>] [-v<variable>] [-l<level>] [--test] [--split] [--nosummary] <nc file>

or

./nc_average --summary <nc file>

or

./nc_average [-h | --help]

Switches: some combination of:

-s<start> The lower limit of the range of time steps to average. The first time step of the first average. It must be an integer. The default is 1. A negative setting is permitted even though no negative time steps exist.

-e<end> The upper limit of the range of time steps to average. The default is 999999. The '-g' switch is usually easier to use.

-c<consecutive> The number of consecutive time steps in a 'set'. It must be a positive integer. The default is 1.

-i<increment> The number of steps between the start of two consecutive 'sets'. <Increment> is applied to <start>. It must be a positive integer. The default is 1.

-r<repeat> The number of 'sets' in a single, averaged, output grid. This is also the number of times for which the increment must be carried out. <repeat> must be a positive integer. The default is 1.

-n<next incr> The increment applied to <start> to begin the next average. <next incr> must be an integer, equal to, or greater than 0. The default is 1. A setting of 0 may have some application.

-g<grids out> The number of averaged output grids to produce. It must be a positive integer. Default is a function of other values.

-h<header file> The name of a an Arc/Info header file containing ASCII-grid header records. The default is <nc file>, but with an '.hdr' extension. The keywords 'ncols', 'nrows', 'xllcorner', 'yllcorner', 'cellsize', and 'nodata_value' are recognized, and their values are (mostly) written to the output file. All other records are ignored. Values for 'ncols' and 'nrows' are set by the program. If a 'nodata_value' value is not provided in the header file, the default (-9999) is used. Note, if no header file exists, or if it is not specified, then nc_average will insert only three lines at the top of each ASCII output file (containing values for 'ncols', 'nrows' and 'nodata_value'), because these are the only items it can determine from the NetCDF file info.

-v<variable> The name of the 3-D or 4-D NetCDF variable to process. The default is any variable with a rank of 3 or more.

-l<level> The value of the NetCDF level variable, which must be an integer greater than or equal to 1. The default is 1.

--test Display the number of each time step included in each averaged output grid. No averages are calculated. Try it.

--split Write each output ASCII-grid to a separate file. The default is to write all output grids to a single file, with a single header at the start of the file. If <next incr> is set to 0 when using '--split', all output file names will be identical (since they have the same starting time step), and will overwrite each other.

--summary Display a summary of the NetCDF dataset contents. No averages are calculated, only the summary is displayed.

--nosummary Disable the default display of the NetCDF dataset summary.

--help Display detailed program usage information.

-h Same as '--help'.

Additional Notes:

<nc file> is the name of the NetCDF input file.

In all NetCDF data files, the first timestep is numbered "1" not "0".

Output file names are based on <nc file>, but with the last file extension removed, and '_####.asg' appended, where '####' is the number of the input time step that is the first grid in the ASCII output file.

It is possible to create output grids that are filled completely with the NODATA value. If all the time steps for a given average are outside the range of valid time steps, then a grid initialized to NODATA will be produced. This is not necessarily a problem. Such a grid would act as a 'place holder' if multiple grids are being written to a single output file.

When the program runs it shows which months (specified by number) are included in the average (or in each average if more than one is being generated). If a particular month does not exist, a warning message is generated. The months the program does *not* complain about are included in the average; obviously, those months that do not exist in the file are *not* included in the average. Use the '-test' switch to experiment, without actually writing any data.

A particular issue surrounds the handling of "missing" months when calculating averages of sequences of months that cross year-ends. For example, suppose you want to generate winter averages using data from December, January and February, from a file containing 10 complete years of monthly data. The program will generate *nine* DJF averages by starting with months 12, 13, and 14 as the first average (i.e., assuming you use '-s12' in the command line). Alternatively, it can be made to generate *ten* averages from the same data by starting with month 0, using '-s0' (to indicate December of the year *before* the start of the data set). Because month 0 does not exist, the program will issue the warning and

compute the first winter average from months 1 and 2 only, with all remaining averages calculated properly using sets of three months.

Author: Fred Woslyng, February 2001 (some code taken from Ryan Sherrington's nc2asg.c).

Possible Improvements:
(1) Check whether the output files already exist, before overwriting them.

I.15. NetCDF Modify Attribute (nc_mod_attr)

Description:
Modify the value of an existing attribute of a NetCDF variable, or of a global attribute, or create a new attribute and value. The program is limited to writing a single text string, or a single numeric value for a given attribute. Multiple values for a single attribute are not supported. Use this program in combination with the '--summary' switch of nc_average, or one of the other NetCDF programs which supports the summary option, to determine variable and attribute names, and to verify the change to the attribute.

Usage:

./nc_mod_attr <variable name> | NC_GLOBAL <attr name> <attr value> <nc file> [NC_CHAR | NC_BYTE | NC_SHORT | NC_INT | NC_FLOAT | NC_DOUBLE]

or

./nc_mod_attr [-h | --help]

Switches:

--help	Display detailed program usage information.
-h	Same as '--help'.

Additional Notes:
<variable name> is the name of an existing variable in <nc file>. The name is case sensitive. 'NC_GLOBAL' accesses the global attributes.
<attr name> is the name of an existing attribute of variable <variable name> (or NC_GLOBAL), or a new attribute to create for the variable (or globally). The name is case sensitive.
<attr value> is the new value of the existing attribute <attr name>, or the value of a new attribute. The value must either be a string, or a single numeric value. Multiple values are not supported. Strings containing blanks (white space) must be enclosed in quotes.
<nc file> is the name of the NetCDF input file.
The optional NetCDF data type ('NC_CHAR', 'NC_BYTE', 'NC_SHORT', 'NC_INT', 'NC_FLOAT' or 'NC_DOUBLE') argument must be specified if the attribute does not already exist. If the attribute does exist, the argument is ignored.

Author: Fred Woslyng, April 2001.

Possible Improvements:
(1) Change the command line processing to allow multiple attribute values.

I.16. NetCDF Variable Rename (nc_rename_var)

Description:

Rename a variable in a NetCDF file. Use this program in combination with the '--summary' switch of nc_average, or one of the other NetCDF programs which supports the summary option, to determine variable names, and to verify the changes.

Usage:

./nc_rename_var <variable name> <new variable name> <nc file>

or

./nc_rename_var [-h | --help]

Switches:

--help Display detailed program usage information.
-h Same as '--help'.

Additional Notes:

<variable name> is the name of an existing variable in <nc file>. <new variable name> is the name to which variable <variable name> will be renamed. <nc file> is the name of the NetCDF input file. File names are case sensitive.

Author: Fred Woslyng, April 2001.

I.17. NetCDF Subregion (nc_subregion)

Description:

Select a subregion of a NetCDF grid by specifying the 'x' (longitude) and 'y' (latitude) coordinates of the lower left corner and the upper right corner of the subregion. The output is written in ASCII-grid format. The longitude and latitude are actually integer x-y coordinates. The origin (1, 1) is located in the lower left corner of the grid. The program nc_xy_grid may be useful in determining the x-y coordinates of the subregion.

The NetCDF variable that is read in is expected to be a function of time, level, latitude, and longitude, and NetCDF dimensions and variables, with those names, are expected to exist.

Usage:

./nc_subregion [-s<start>] [-e<end>] [-h<header file>] [-v<variable>] [-l<level>] [--split] [--nosummary] <nc file> <lower left 'longitude'> <lower left 'latitude'> [<upper right 'longitude'> <upper right 'latitude'>]

or

./nc_subregion --summary <nc file>

or

./nc_subregion [-h | --help]

Switches: some combination of:

-s<start> The lower limit of the range of time steps to select. The default is 1.
-e<end> The upper limit of the range of time steps to select. The default is 999999.
-h<header file> The name of a file containing ASCII-grid header records. The default is <nc file>, but with an '.hdr' extension. The keywords 'ncols', 'nrows', 'xllcorner', 'yllcorner', 'cellsize', and 'nodata_value' are recognized, and their values are (mostly) written to the output file. All other records are ignored. Values for 'ncols' and 'nrows' are set by the program. If a 'nodata_value' value is not provided in the header file, the default (-9999) is used.

-v<variable> The name of the 3-D or 4-D NetCDF variable to process. The default is any variable with a rank of 3 or more.
-l<level> The value of the NetCDF level variable, which must be an integer greater than or equal to 1. The default is 1.
--split Write each output ASCII-grid to a separate file. The default is to write all output grids to a single file, with a single header at the start of the file.
--summary Display a summary of the NetCDF dataset contents. No subregion is calculated, only the summary is displayed.
--nosummary Disable the default display of the NetCDF dataset summary.
--help Display detailed program usage information.
-h Same as '--help'.

Additional Notes:
<nc file> is the name of the NetCDF input file.
<lower left 'longitude'> and <lower left 'latitude'> are the x-y integer coordinates of the lower left corner of the subregion. The origin (1, 1) is located in the lower left corner of the grid. Both coordinate values must be greater than or equal to 1.
Optional arguments <upper right 'x'> and <upper right 'y'> are the x-y integer coordinates of the upper right corner of the subregion. Both coordinate values must be greater than or equal to the corresponding values of the lower left corner coordinates. They default to the coordinates of the *upper right* corner of the input grid.
Output file names are based on <nc file>, but with the last file extension removed, and '_####.asg' appended, where '####' is the input time step of the first grid in the ASCII output file.

Author: Fred Woslyng, March 2001 (some code taken from Ryan Sherrington's nc2asg.c).

Possible Improvements:
(1) Check whether the output files already exist, before overwriting them.

I.18. NetCDF X-Y Coordinate Grid (nc_xy_grid)

Description:
Generate an ASCII-grid file of x-y coordinate values to be used as an overlay. The dimensions of the grid, and the range of cell values are specified by the <columns> and <rows> arguments. The value of each cell is the concatenated x-y coordinates of the cell, padded with leading zeroes, if necessary. For example, the value of the cell at the grid origin (1,1) is 00010001. The file is useful for determining the x-y coordinates of a desired region for the NetCDF Subregion program (I.17).

Usage:

./nc_xy_grid <columns> <rows>

or

./nc_xy_grid [-h | --help]

Switches:

--help Display detailed program usage information.
-h Same as '--help'.

Additional Notes:

<columns> defines the number of columns in the grid. There is no default. The value of <columns> must be a positive integer.
<rows> defines the number of rows in the grid. There is no default. The value of <rows> must be a positive integer.

Author: Fred Woslyng, April 2001.

Possible Improvements:
(1)Check whether the output file already exists, before overwriting it.

I.19. NetCDF Mean (nc_mean)

Description:
Given two "parallel" NetCDF files containing variable A and variable B data used as input (such as maximum and minimum temperature), generates a single NetCDF file as output, containing the mean of A and B for each data point (i.e., mean = (A + B)/2). Works on any size of NetCDF grid. The most obvious use for this is to create maps of monthly mean daily temperature from monthly mean daily minimum and maximum temperatures.

Given input files with names such as:

setAd.danom.nc
setBd.danom.nc

The output file is named: temp.danom.nc. The mean value generated by *nc_mean* is named "mean". This can be changed to any other name using NetCDF Variable Rename (I.16).

Other names used inside the NetCDF files are consistent with the input files. (I.e., they should be the same in all respects, otherwise unpredictable results will occur).

Usage:

```
./nc_mean  [ -[v][x][h][a] ]  <setA file>  <setB file> [<start year>  <end year>]
                                        [--nosummary]
```

or

```
./nc_mean  --summary  <nc file>
./nc_mean  [-h | --help]
```

Switches: some combination of:

-x	Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than the default 8.2f format.
-y	Merge corresponding years, when multiple years (grids) of data exist in the <mean file>. Has very limited application.
--help	Display detailed program usage information.
-h	Same as '--help'.

Author: Ryan Sherrington, May 2001 (with code borrowed from Fred Woslyng's anu_merge and nc_aggregate).

Possible Improvements:
(1) Pass the NO_DATA value as a command line argument, so that different NO_DATA values can be supported.
(2) Dynamically allocate the grid arrays based on the number of rows and columns in the grid.

I.20. NetCDF Range (nc_range)

Description:
Given two "parallel" NetCDF files containing variable A and variable B data used as input (such as maximum and minimum temperature), generates a single NetCDF file as output, containing the range of A and B for each data point (i.e., range = abs(A - B)). Works on any size of NetCDF grid. Similar to nc_mean, the most obvious use for nc_range is to create maps of monthly mean daily temperature range from monthly mean daily minimum and maximum temperatures.

Given input files with names such as:

setAd.danom.nc
setBd.danom.nc

The output file is named: range.danom.nc. The value generated by *nc_mean* is named "range". This can be changed to any other name using NetCDF Variable Rename (I.16).

Other names used inside the NetCDF files are consistent with the input files. (I.e., they should be the same in all respects, otherwise unpredictable results will occur).

Usage:

```
./nc_mean  [ -[v][x][h][a] ]  <setA file>  <setB file> [<start year>  <end year>]
                                        [--nosummary]
```

or

```
./nc_mean  --summary  <nc file>
./nc_mean  [-h | --help]
```

Switches: some combination of:

-x	Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than the default 8.2f format.
-y	Merge corresponding years, when multiple years (grids) of data exist in the <mean file>. Has very limited application.
--help	Display detailed program usage information.
-h	Same as '--help'.

Author: Ryan Sherrington, May 2001 (with code borrowed from Fred Woslyng's anu_merge and nc_aggregate).

Possible Improvements:
(1) Pass the NO_DATA value as a command line argument, so that different NO_DATA values can be supported.
(2)Dynamically allocate the grid arrays based on the number of rows and columns in the grid.

I.21. NetCDF Add (nc_add)

Description:
Given two "parallel" NetCDF files containing variable A and variable B data used as input (such as maximum and minimum temperature), generates a single NetCDF file as output, containing the sum of A and B for each data point (i.e., add = A + B). Only the first 12 months of variable A are

used (normally the file will contain only 12 months of data). These monthly values are added repeatedly to the corresponding values for each month of a long time-series of variable B. Works on any size of NetCDF grid. The most obvious application is to combine a file of 12 monthly climate normals such as temperature with monthly differences derived from a GCM simulation.

Given input files with names such as:

setAd.danom.nc
setBd.danom.nc

The output file is named: add.danom.nc. The value generated by *nc_add* is named "add". This can be changed to any other name using NetCDF Variable Rename (I.16).

Other names used inside the NetCDF files are consistent with the input files. (I.e., they should be the same in all respects, otherwise unpredictable results will occur).

Usage:

./nc_add [-[v][x][h][a]] <setA file> <setB file> [<start year> <end year>] [--nosummary]

or

./nc_add --summary <nc file>
./nc_add [-h | --help]

Switches: some combination of:

-x	Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than the default 8.2f format.
-y	Merge corresponding years, when multiple years (grids) of data exist in the <mean file>. Has very limited application.
--help	Display detailed program usage information.
-h	Same as '--help'.

Author: Ryan Sherrington, July 2001 (with code borrowed from Fred Woslyng's anu_merge and nc_aggregate).

Possible Improvements:
(1) Pass the NO_DATA value as a command line argument, so that different NO_DATA values can be supported.
(2) Dynamically allocate the grid arrays based on the number of rows and columns in the grid.

I.22. NetCDF Multiply (nc_multiply)

Description:
Given two "parallel" NetCDF files containing variable A and variable B data used as input (such as maximum and minimum temperature), generates a single NetCDF file as output, containing the product of A and B for each data point (i.e., product = A * B). Only the first 12 months of variable A are used (normally the file will contain only 12 months of data). These monthly values are multiplied repeatedly by the corresponding values for each month of a long-term time series of variable B. Works on any size of NetCDF grid. The most obvious application is to combine a

file of 12 monthly climate normals such as precipitation with monthly ratios derived from a GCM simulation.

Given input files with names such as:

setAd.danom.nc
setBd.danom.nc

The output file is named: multiply.danom.nc. The value generated by *nc_multiply* is named "multiply". This can be changed to any other name using NetCDF Variable Rename (I.16).

Other names used inside the NetCDF files are consistent with the input files. (I.e., they should be the same in all respects, otherwise unpredictable results will occur).

Usage:

./nc_add [-[v][x][h][a]] <setA file> <setB file> [<start year> <end year>] [--nosummary]

or

./nc_add --summary <nc file>
./nc_add [-h | --help]

Switches: some combination of:

-x	Write the ANUSPLIN output with extra precision. Specifically, 12.5e rather than the default 8.2f format.
-y	Merge corresponding years, when multiple years (grids) of data exist in the <mean file>. Has very limited application.
--help	Display detailed program usage information.
-h	Same as '--help'.

Author: Ryan Sherrington, July 2001 (with code borrowed from Fred Woslyng's anu_merge and nc_aggregate).

Possible Improvements:
(1) Pass the NO_DATA value as a command line argument, so that different NO_DATA values can be supported.
(2) Dynamically allocate the grid arrays based on the number of rows and columns in the grid.