NCAR/TN-261+IA
NCAR TECHNICAL NOTE

January 1986

CCM Modular Processor Users' Guide (Version PROC01)

Richard J. Wolski

ATMOSPHERIC ANALYSIS AND PREDICTION DIVISION

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH
BOULDER, COLORADO

PREFACE

This Users' Guide is a printed version of the online documentation for the CCM Modular Processor, which is maintained on the IBM 4341 computers at NCAR. This printed version is the same as the online version as of the date shown on each page, except for minor formatting changes, and the addition of an Appendix containing sample plots. Page numbers have been added to the Table of Contents (Part 1), and separate indices have been added for the Topic Discussion Keys (Part 2) and the Input Control Parameter (ICP) Keywords (Part 3).

For routine referencing, most users will probably find the online file easier to use than this printed copy, since a string search macro has been provided to simplify cross-referencing. See the HELP Topic Discussion for details on accessing and using the online file. Before attempting to use either form of this documentation, please read the INTR Topic Discussion; it describes the organization of the documentation and the symbolic notations used.

I would like to thank all of the CCM users and programmers who have contributed to the development of the Modular Processor over the years since its inception. I would especially like to thank Mike Dias and Tom Mayer, who have both contributed a significant amount of code to the present version, and Bob Chervin for his painstaking review of this documentation.

Rick Wolski
January 1986

<-> Part 1. List of topics and keys

->LTOP - List of topics and keys

Dates of last modification (if any) are in parentheses.

USING THE DOCUMENTATION

USING THE PROCESSOR

INPUT AND OUTPUT OF DATA

GRAPHICS OUTPUT

<-> end of Part 1. List of topics and keys

The following are appended to the printed version only:

Appendix A - Sample Plots (see list on following page)

Index for Topic Discussions (Part 2)

Index for ICP Keywords (Part 3)

List of Sample Plots in Appendix A

<-> Part 2. Descriptions of Processing Options

->HELP - Online use of the documentation

In order to use the documentation online on the IBM 4341's at NCAR, it is necessary to link to the CSMLIB 191 disk. This can be done by issuing the command:

```
LINKTO CSMLIB
```

To automatically link to this disk at each logon, insert the following lines into your PROFILE EXEC:

```
CP LINK CSMLIB 191 vda RR
ACCESS vda m
```

where vda is an unused virtual disk address, and
m is an unused file mode letter.

This virtual disk also contains a few utilities which make examining the documentation online more convenient. The recommended way of using the documentation is to start by issuing the command:

```
PROCDOC
```

PROCDOC is a CMS EXEC which initiates XEDITing of the documentation file. Its single, optional argument is the file name of the documentation file, which defaults to the most recent version.

Since PROCDOC XEDITs the documentation file (which is rather large), this command may fail if your virtual memory allocation is too small. If you get an error message indicating that the file is too large, you can increase your allocation temporarily by issuing the commands:

```
DEFINE STOR 1216K
IPL CMS
```

1216K is currently the largest amount of virtual memory that can be requested without special permission. Note that any temporary disks will have to be re-accessed after this sequence. To increase your memory allocation permanently, issue the command:

```
DIRM STOR 1216K
```

PROCDOC also causes XPROCDOC to be used as the XEDIT profile, which configures XEDIT for use with the documentation file. This profile sets screen display characteristics (full-screen edit mode only), limits string searches to the first 15 columns (for efficiency), makes string searches circular across the end-of-file, and causes case (upper vs. lower) to be ignored in string searches.

XEDIT Note: Since XEDIT allows the editing of more than one file at a time, the PROCDOC command can be issued from within XEDIT, or conversely, editing of another file can be initiated while the documentation file is being examined. This might be useful, for example, for checking the syntax of a Processor data card image while editing the run deck. If XEDIT is used in full-screen mode, the subcommand "SET SCREEN 2" can be used to display both files simultaneously. Alternatively, the command "XEDIT" (or simply "X") without an argument can be used to move back and forth between the documentation file and the run deck being prepared. (Editing a number of large files at the same time may require more virtual memory than is available, since a copy of all files must exist in memory simultaneously.)

Once in XEDIT, positioning, paging, and typing commands function as usual. There are also a few XEDIT macros which make using the documentation online considerably more convenient. These macros are stored on the same disk as the documentation file; they are described below.

The XEDIT macro GO (which may be abbreviated G) can be used to "go" to a particular point in the documentation file. Its single argument is the search string used to locate the desired documentation. (These search strings are described in more detail in the next topic discussion.) The line from which GO was (successfully) invoked is "remembered", and a return to this line can be executed by issuing the command GO without an argument. Up to ten locations can be backtracked in this way; after ten (or the total number of successful GO searches, whichever is less), the backtracking becomes circular. For example, the command:

```
GO INTR
```

can be used to move to the beginning of the introduction topic. The command:

```
GO
```

will then cause a return jump to the location from which the "GO INTR" was issued. Movement caused by standard XEDIT subcommands (such as paging forward or typing lines) will not interfere with the operation of GO; backtracked locations are always the locations from which GO commands were issued.

Implementation Note: The macro GO uses line numbers to keep track of locations in the file. Since insertion and deletion of lines changes line numbers, these operations will cause GO to malfunction.

The macro KEY provides an easy way to determine the search key for a Topic Discussion (e.g., the key INTR used in the above GO example). Each Topic Discussion has such a key defined by Part 1 - see the Introduction for a description of the organization of this documentation. KEY (which may be abbreviated K) lists all lines in Part 1 containing a specified string. For example, the command:

KEY interp

produces a list of all of the Topic Discussions which deal with interpolation. The desired Topic can then be chosen, and the key used in the GO command to locate the discussion. Do not use string delimiters in specifying the argument for the KEY command, even if the search string contains imbedded blanks (i.e., multiple words).

There is also an XEDIT macro which can be used to relate the online file to a printed version which has been divided into 60-line pages. The approximate page number for the current (top of screen) line can be displayed by issuing the command PAGE, which can be abbreviated P. PF01 can also be used to issue this command. The page number cannot be determined exactly because of blank space at the bottom of some pages in the printed version, but the approximation is usually correct to within about 2 pages. In addition to the page number, the creation date for the online file is also displayed. If this date is later than the date on the printed version, then the two versions are different, and the displayed page number may be incorrect.

Another XEDIT macro has been provided for those who use XEDIT in line mode. This macro, named TP (no abbreviation), will instruct XEDIT to TYPE to the end of the topic, or a maximum of 20 lines, whichever comes first. If TYPEing is cut off at 20 lines, then TP prints a message giving the number of lines remaining in the topic. At this point, the TP command may be re-issued to type the next group of lines. If XEDIT is accessed in line mode, TP is automatically invoked by GO after moving to the requested location. When backtracking in line mode, GO positions the line pointer so that the backtracked line is the current line after TP is executed (TYPE moves the line pointer).

Implementation Note: In order to avoid conflict with user EXECS or XEDIT macros, the names GO, KEY, PAGE and TP are actually XEDIT synonyms (defined by XPROCDOC). The real names of these macros are GOPDOC, KEYDOC, PAGEDOC, and TPPDOC, respectively.

The terminal used to examine this documentation online should be able to display 80 columns of upper and lower case alpha characters, and it should also be able to display square [] and pointed <> brackets in addition to the more common special characters such as arithmetic and punctuation characters. A full-screen capability is not necessary. All of the normal modes of access to the IBM 4341 (including UNINET and SIM3278) provide sufficient capabilities.

->INTR - Introduction to the documentation

This Users' Guide contains a complete description of the capabilities of the CCM Modular Processor, including all of the information needed to run the code on the Cray-1 at NCAR. All users should be familiar with the introductory material, but the main purpose of this documentation is to

provide a quick reference for specific Processor capabilities. The extensive use of cross-references is intended to avoid the need for sequential reading of the whole Users' Guide.

Some aspects of Processor design are discussed where appropriate, but there is no attempt to document the details of the code. This is a rewrite of Part 1 of the NCAR Tech Note TN-216+IA "Users' Guide for the NCAR CCM Modular Processor". Part 2 of this Tech Note (description of the code) is still for the most part correct, but it is now incomplete.

DOCUMENTATION ORGANIZATION

There are three parts to this documentation: 1) a list of Topic Discussion keys, including a one-line description of each topic, 2) descriptions of the processing capabilities (Topic Discussions), and 3) an alphabetical listing of Input Control Parameter (ICP) descriptions. Part 1 is just a table of contents and list of keys for Part 2. Each 4-character key, when prefixed with ">", can be used as a search string to locate the topic discussion in Part 2, since the first line of each Topic Discussion in Part 2 contains such a string. (The macro GO will automatically add the ">" if it is omitted.) The number of dashes preceding the ">" in the located line indicates the depth of nesting in the topic outline. These Topic Discussions describe the available processing operations, and how they are performed. Part 3 shows how to request the various processing options by describing the Input Control Parameter (ICP) keywords and the values which may be assigned to them on input data card images.

XEDIT Note: A hardcopy of Part 1 (the list of topics and keys) can be very useful when using this documentation online. Once in XEDIT, the following sequence of commands can be used to generate a printout of the topic list, which is only a few pages long.

```
GO LTOP
PUT />/ fn ft fm
PRINT fn ft fm
```

where "fn ft fm" is the fileid for a new file on a write-accessable disk. The IBM 4341 printers have a very limited character set (no lower case characters or square brackets), but this is not a problem in the topic list printout.

The XEDIT macro KEY can also be used to determine the appropriate key from an arbitrary search string; this is actually a crude but general index function. See (*HELP) for details.

CROSS REFERENCING

Both Parts 2 and 3 contain cross-references to Topic Discussions and ICP keywords. Topic Discussion references consist of the four-character key, prefixed with a "*" and enclosed in parentheses. For example, (*INTR) is a reference to this topic. References to ICP keywords (see following paragraph) consist of the keyword enclosed in brackets: square brackets for primary keywords, and pointed brackets for secondary keywords. For example, [TAPESc] is a reference to the primary keyword TAPESc, while <TYPEc> is a reference to the secondary keyword TYPEc. The XEDIT macro GO simplifies cross-referencing when examining this documentation online. For example, pursuing the reference (*VERT) might result in the following command sequence:

```
G VERT
G PRESS
G
G
```

The first command causes a move to the topic discussion on vertical interpolation, which references the ICP [PRESSLE]. The second command causes a move to the description of this ICP, and the last two commands cause a return to the original (*VERT) reference.

INPUT CONTROL PARAMETER (ICP) KEYWORDS

Primary ICP keywords are used to activate options and control major processing characteristics; familiarity with all of an option's primary keywords is recommended before that option is used. Secondary ICP keywords control relatively minor aspects of an option and usually default to the simplest or most commonly used values. Familiarity with these keywords is not always necessary. The distinction between primary and secondary keywords is relative to the discussion at hand; a primary keyword for one option may be secondary for another. The ICP keywords are categorized in this way solely to minimize the amount of documentation read by new or casual users.

Most Topic Discussions describe a processing option which is controlled by a single ICP switch. This kind of ICP switch is listed as the first primary keyword, and is usually the minimal ICP specification required to activate the option being described. Primary ICPs, however, are NOT limited to minimal ICPs; it is usually necessary to understand the remaining primary ICPs in order to get the desired result.

A referenced keyword can be located (online) with a string search in the same way that discussion topics are located: prefix the keyword with a ">" to form the search string. (The prefix may be omitted if GO is used.) The first five characters of a keyword are usually sufficient for uniqueness.

Parentheses, asterisks, and brackets should NOT be used as part of the search strings.

All keywords listed in this documentation are symbolically compressed if (and only if) they are "Case" or "field pass" dependent (*DEFS) (i.e., a separate keyword exists for each possible Case and/or field pass). A lowercase "c" suffix indicates that "A", "B", or "C" should be used in its place, and a lowercase "n" suffix indicates that the field pass number should be used in its place. Both suffixes may be present, in which case "n" is always last. Some Case and field pass dependent keywords may not be defined for all possible Cases or field passes; exceptions are noted in the keyword descriptions. The keyword descriptions also include the minimum and maximum number of values which may assigned to a particular keyword, and the FORTRAN data type(s) of the value(s). Examples of input cards are also included for all but the simplest ICPs.

TOPIC DISCUSSIONS

Most Topic Discussions in Part 2 deal with options individually; for information concerning the relationships between different options, consult the topic on ordering (*ORDR). Since the result of one operation is usually the starting point for the next, the order of the operations can significantly affect the final result. Unless otherwise noted, any combination of Processor options may be specified in the same jobstep.

The Topic Discussions (*DESC) (*DEFS) (*NICP) (*SICP) (*ARCH) (*CJDK) briefly describe the Processor, and the mechanics of running it on the Crays at NCAR. It is recommended that new users read these discussions before attempting to run the Processor.

DOCUMENTATION MAINTENANCE

The online version of this documentation is maintained on the CSMLIB 191 disk on the IBM 4341 at NCAR, and is updated on an irregular basis; different versions may exist simultaneously. The file type is always PROCDOC; the file name indicates both the Processor version documented, and the date of the last update of the documentation file, in the same manner as the file name for Processor modification files. For example, the file P01JAN02 PROCDOC documents the PROC01 version, and the file was last updated on January 2, 1986.

These files are generated by running a basic, continuously updated file through a text formatter. A paginated version can also be produced in this way, automatically adding page numbers to Part 1 entries, and appending page indices for Parts 2 and 3.

When Part 2 topic discussions are updated or added, the corresponding line in Part 1 ends with the date of last update, enclosed in parentheses. In addition, all added or modified lines in Parts 2 and 3 are bracketed by "update lines" as follows:

```
+beg (mm/dd/yy)
...
... changed or added text ...
...
+end (mm/dd/yy)
```

where (mm/dd/yy) is the date of the change.

->DESC - Brief description of the Modular Processor

The NCAR CCM Modular Processor is the primary tool for the analysis of data output by the NCAR Community Climate Model (CCM). The general features of the Processor are listed below.

* Each major capability is isolated in an independent code Module with a standard interface for input and output of "units" of data.
* All Modules (and therefore all capabilities) exist in a single program. Modules which add to or modify the data being processed are usually strung together in series: the output of one Module becomes the input of the next Module. Other Modules, such as those which do spectral operations, may branch off and parallel the main data stream.
* User-controlled options determine which Modules are executed; most Modules can be skipped without interrupting the flow of data to other Modules. The options are controlled by values assigned to Input Control Parameters (ICPs) on a group of input card images (*NICP) (*SICP). Virtually all ICPs default to a predetermined setting if they are not specified by the user. All major processing operations, except for reading of input data, are skipped by default.
* All Model configuration parameters (e.g., space and time resolution) are handled automatically.
* All COS dataset handling for input and output of tapes (TBM VSNs) is handled by the code; no dataset JCL is required (*CJDK).
* Although the order in which Modules are executed is fixed in the standard Processor configuration, changing this order usually involves little more than rearranging subroutine calls within the code.
* All data manipulations are done in main memory, resulting in a minimal amount of I/O. Since memory is managed dynamically, a very large number of Modules can be strung together in the same run.
* For processing a large number of Model fields, or for very high spatial resolutions, a single run may make multiple passes over the input data tapes (*DEFS).
* A single Cray job may make several related Processor runs by jobstepping (*DEFS).

* The code is highly optimized on the Cray-1 for the most common processing tasks; for other tasks, processing efficiency can often be improved with simple changes in operational procedures and/or code (*COST) (*CMEM).

->DEFS - Definitions of some Processor terms

Definitions of some important Processor terms are given below, in alphabetical order. A particular definition may be located by prefixing the term with a ">" and using it as a search string. (No prefix is necessary if GO is used).

blocked point	latitude
Case	level
day	longitude
field	Processor job
field pass	Processor run
history tape	Save tape
Input Control Parameter (ICP)	

>blocked point - a grid point at which the value of some field is undefined. For example, a field may be undefined on a pressure surface which is below ground elevation. Blocked points are flagged by assigning to them a special value (1.E36) which is well outside the normal range of values for any field. Most of the computational algorithms in the Processor must check for this special value and take appropriate action when it is encountered. The most visible result of blocked points is their special treatment in contoured graphics - they are ignored, and appear as "holes" in the pattern of contour lines.

>Case - a set of Model data grouped in order to allow comparing it to another Case. The primary or control Case is designated "Case A"; Case B may be quantitatively compared to Case A, and the result is designated Case C. If no comparisons are requested, the data being processed should normally be designated Case A. (*COMP)

>day - a particular Model time at which instantaneous values of fields are defined. The unit of measurement is days (i.e., number of 24 hr. periods, including time of day as a fraction) relative to some initial time defined by the Model. Note that requesting a particular "day" does NOT imply the processing of all Model output for some 24-hour period.

>field - a set of values of some particular quantity associated with the Model, defined on some set of points in time and (three-dimensional) space. A field may be either output explicitly by the Model or derivable from other output fields (*DFLD). Requesting a particular field results (by default) in the processing of all points in space at which that field is defined. Each field has associated with it a name consisting of up to eight characters. Since field names may

vary from one Model version to another, the Model documentation should be consulted for a list of these names. See the "Users' Guide to NCAR CCM" for a list of fields on the history tapes output by CCMOB. Requesting processing for a history tape field which does not exist will result in a printed list of fields on the history tape being processed, and the job will abort.

>field pass - one pass through all input data. The number of fields which can be processed by reading the input tapes in one pass is limited by the amount of computer memory available. It is possible, however, to process a large number of fields in a single Processor run because the code contains provisions for re-reading the input data and processing additional groups of fields. This is referred to as making multiple "field passes". Also see (*MLIM).

>history tape - a volume of data, usually output by the Model, consisting of a sequence of logical records containing the values of Model variables for a series of Model days. The use of the word "tape" does not imply a physical, magnetic tape; virtually all history tapes are archived as data volumes on the Ampex Terabit Memory System (TMS-4, also known as the TBM) at NCAR. All such volumes have a unique 6-character name associated with them. The size of a typical volume is about 5 million Cray words. When a history tape is copied to the Cray disks for processing, it may be referred to as a Cray "dataset".

>Input Control Parameter (ICP) - input parameters which are assigned values by the user in order to control processing operations. Each ICP set is associated with a unique keyword that is used to identify the specified value(s).

>latitude - latitude is defined as increasing northward, positive in the northern hemisphere, negative in the southern. Values are always specified in degrees.

>level - a point in the Model vertical dimension which has a single value of the vertical coordinate associated with it. This should not be confused with a Model "layer", which has thickness. For some purposes, the Model treats fields as describing, in some average sense, the property of an entire layer; for other purposes, including most processing, field values must be associated with a specific level within the layer. There are generally only two possibilities: the layer midpoint, also referred to as a full level, and the interface between two layers, also called a half-level. The Model defines the location of these levels for all fields on the history tape. The value identifying a pressure level (which is indicated by a 'P' suffix), has the units of millibars. The value identifying a sigma level (which is indicated by an 'S' suffix), is 1000. times the actual sigma value.

>longitude - longitude is defined as increasing eastward, positive east of Greenwich, negative west of Greenwich. Values are always specified in degrees between minus 180. and the last grid point before (west of) plus 180. The dateline is always referred to as minus 180.

>Processor job - the execution of an entire batch submittal (Cray job deck). This may include more than one Processor run because it is possible to re-execute the Processor code in the same Cray job. This is referred to as "jobstepping". (*CJDK)

>Processor run - a single execution of the Processor code. This is not necessarily the same as a "Processor job". (*CJDK)

>Save tape - a volume of data output by the Processor and saved so that additional processing can be done at a later time. Save tapes are usually TMS-4 volumes, but may also exist only on the Cray disks.

->NICP - Naming conventions for Input Control Parameter keywords

Each Input Control Parameter is associated with a unique keyword - an upper case alphanumeric string consisting of up to 8 characters (only the first 7 characters are significant). Each keyword has up to 3 parts: 1) an alpha mnemonic for the associated option, 2) a single alpha character designating the Processor Case (Case-dependent options only), and 3) a single numeric character designating the field pass number (field pass dependent options only). See definitions of "Case" and "field pass" (*DEFS). The Case character is always A, B, or C; when Cases A and B are compared, the result is Case C. If only one set of Model data is to be processed, then it should be designated Case A. If the last alpha character of a keyword is A, B, or C, it always implies a Case dependent option. Case or field pass dependent options have separate, independent keywords for each Case or field pass, with two exceptions: sometimes there is no Case C counterpart to Cases A and B, and some field pass dependent options are implemented only for field pass 1. These exceptions are noted in the keyword descriptions. Input Control Parameters whose keywords do not contain Case letter suffixes apply to all Cases, and those without field pass numeric suffixes apply to all field passes. All references to keywords in this documentation symbolically compress the Case and field pass suffixes to lowercase "c" and "n", respectively. If no Case comparisons are being made, "A" should be used in place of the "c". If only one field pass is being made, "1" should be used in place of "n". These two substitutions will always result in a properly defined keyword.

->SICP - Syntax of Input Control Parameter data card images

The following is a summary of the syntactical rules for the data card images which control Processor options.

* The general format is keyword, equals sign, value(s). All keywords and values must be contained in columns 2 through 72. Multiple values following a single keyword must be separated by commas. The last value specified for a keyword should NOT be followed by a comma. Blanks between values are not significant. The number of values to be specified is variable for many keywords; this number is automatically determined when the values are read, and may be used as implicit input (e.g., the number of time filter weights supplied is used to determine the time filter width.)

* The TYPE of values is significant - follow standard FORTRAN conventions for the specification of constants for each of the required types. CHARACTER values may be either enclosed by single quotes or prefixed with nH where n is the number of characters in the string. All CHARACTER values are padded on the right with blanks so that the total number of characters is a multiple of eight.

* Continuation cards are indicated by either a leading comma on the continuation card or a trailing comma on the previous card. Also, if a card contains only a keyword and equals sign, the following card is interpretted as a continuation. The keyword should not be repeated on continuation cards; if a keyword appears more than once, previously assigned values will be replaced by the new ones.

* The letter C in column one indicates a comment card; the entire card is ignored.

* The keywords may appear in any order. This ORDERING HAS NO EFFECT ON THE ORDER OF PROCESSOR OPERATIONS. If a keyword appears more than once, all but the last occurrence is ignored.

* Optional parameters whose keywords do not appear at all are assigned the default value(s), which are given in Part 3. The default is also used for unspecified values at the end of a list; single values imbedded in the list may be skipped (and therefore defaulted) by using successive commas.

* Multiple sets of keywords and values may appear on the same card if they are separated by a dollar sign ($).

* Repetition of a value may be indicated by prefixing the value with a repetition count, separated from the value with a colon. For example, DAYSA=5:-2. will result in the processing of the first five days on the specified tape(s). (See <DAYSc> in Part 3.)

* The complete set of card images for a Processor run should be terminated with a card image containing only the word ENDOFDATA.

* For multiple runs within a job (jobstepping) (*DEFS), each run should have its own, complete set of Input Control Parameter card images. The first card image of each set should immediately follow the ENDOFDATA in the

previous set; these sets should NOT be separated by end-of-file delimiters.

->ARCH - Archival and maintenance of the Processor code

Each version of the Processor is identified by a six-character name consisting of the root PROC followed by two numeric digits. This name is used as the VSN for a TBM volume containing the Processor code, which is FORTRAN source in the form of a Cray Update Program Library. See Cray documentation for a description of the Update utility. The most recent version of this Program Library is named PROCO1. The Update Decks contained in this Program Library are listed at the end of this Topic.

The most recent version of the Processor code, including all standard mods, is also maintained as two other files on PSTORE (see SCD documentation): an absolute binary (executable) file, and a relocatable binary (object) file containing compiled code. Users who do not need to modify the code should run from the executable file. Modifications to the code can be made by applying mods to selected Update Decks from the Program Library file on the TBM, and substituting the modified routines for those contained in the object file. There are standard run decks available for both ways of running the Processor. See (*CJDK). The current executable (program) PSTORE file is named /TB/CCMPROC/PROCO1.PRG, and the current object file is named /TB/CCMPROC/PROCO1.OBJ. There is also a third PSTORE file which is normally accessed when the executable file is used; it is named /TB/CCMPROC/PROCO1.DBG, and contains the symbol table used by DEBUG to produce a symbolic dump when a run is aborted. See (*CJDK).

The TBM volume containing the Update Program Library contains all of the source code needed to run the Processor, with the exception of the NCAR line graph package Autograph, and system library routines. These routines reside in libraries which are automatically searched by the Cray loader.

PROCO1 is the only version of the Processor code currently supported. This version is actually just a new name for the most recent update of BPROC7, which is documented by a draft version of this Users' Guide. Since PROCO1 requires different JCL in the run deck, this renaming allows existing BPROC7 decks to continue to execute. However, all BPROC7 users should immediately switch to the new PROCO1 run deck, since the BPROC7 files will be deleted in the near future.

When accessing the source code in Update Program Library form, it may be necessary to apply a set of mods to the Program Library source code. For formally-released versions of the Processor, these mod files are maintained on the CSMLIB 191 disk, using a naming convention which indicates the Program Library to which it applies, plus the creation date. For example, PO1JANO2 PRMODS is a Processor mod file for PROCO1 which was created on January 2, 1986. New mods are normally combined with the most recent set of mods, and a new, complete mod file is created. Previous mod files remain unchanged; users are expected to take appropriate action to incorporate the

new mods. User notification procedures are discussed below. The latest versions of all PSTORE files always contain the latest set of modifications.

For those users who are modifying the Processor code, a news item will be added to the CCM NEWS file on the CSMLIB 191 disk each time a new mod deck is created. All news items are also sent as IBM MAIL messages to the CCM users on the Core Group's distribution list. This news mechanism is also used to make other Processor-related announcements.

The Update Decks contained in PROCO1 are listed below. Decks CONREC, EZMAP, READLX, and VELVCT are NCAR packages which have been modified for use by the Processor. The modifications are contained in Mod Deck MPROCO1, which is part of the Program Library.

ADDERF - add derived fields for all computation types whose first digit is greater than 2 (*DFLD)
CFLD - horizontal projection graphics (*PHOR)
COMPDF - vertical slice computation subroutines for all derived fields in gridpoint space (*DFLD)
CONREC - contoured graphics; modified NCAR utility (*PCON)
CPYFLD - copies a subset of fields from one gridpoint space Data Unit to another
DEFALL - defines all derived fields (*DFLD)
DELFLD - deletes a subset of fields from a gridpoint space Data Unit
DELSPF - deletes a subset of fields from a spectral space Data Unit
DIFFLD - computes Case differences and ratios (*COMP)
DRIVER - main program and drivers; must be loaded first
DUM - Data Unit Manager and dynamic memory manager (Cray Heap) interface
EZMAP - map projections utility (*PHOR)
GD2SHC - gridpoint to spectral space transformation (*SPEC)
HSLICE - reorient gridpoint data from vertical to horizontal slices (not currently used)
ICP - definitions and input processing of ICPs (*SICP)
ILPOVP - computations for inverse Laplacian derived fields (*VDFL)
MASK - general surface type masking (all fields) (*MASK)
MSKAVG - horizontal (masked) area averaging (*HORA)
PRINTM - printing of field values to the output file (*PRNT)
RDHT - read input data from history tapes (*IHIS)
RDST - input and output of Time Average Save tapes (*STAV)
RDTSTP - input and output of Time Series Save tapes (*STSR)
READLX - ICP input; modified NCAR utility (*SICP)
SHCINT - horizontal interpolation in spectral space (*SINT)
SHC2GD - spectral to gridpoint space transformation (*SPEC)
SIG2P - sigma to pressure surface vertical interpolation (*VPRS)
SPBPAS - horizontal band pass filtering in spectral space (*SBND)
SPCUTL - spectral processing utility subroutines (*SPEC)
SPDERF - computation of linear derived fields in spectral space (*DFLD) (*SPEC)

SPDFNL - computation of nonlinear derived fields in spectral space (*DFLD) (*SPEC)
SPGRF - spectral graphics (*SPGR)
TIMAVG - gridpoint space time average statistics (*TAVG)
TIMFLT - time filtering (*TFIL)
TSERP - time series plotting (*PTIM)
UTIL - general utility subroutines
VELVCT - vector graphics; modified NCAR utility (*PVEC)
VERAVG - vertical averaging (*VERA)
WTHSTP - output of Horizontal Slice Save Tapes (*SHOR)
ZLAVG - zonal and meridional averaging and plotting (*ZONA) (*PMEX)
ZSTATS - computation of zonal eddy statistics (*ZEST)

->CJDK - The Cray job deck

For those who do not need to modify the Processor code, the recommended way of running the Processor is from the executable PSTORE file (*ARCH). The file CCMPROCA JOB (on the CSMLIB 191 disk on the IBM 4341) can be used as a starting point. A copy of this deck follows; it will always run the latest version of the Processor, with all standard modifications.

```
JOB,JN=nnnnnn,US=uuuupppppppp,T=120,OLM=300,*TB,*D1.
*
*    ********************************************************
*    * THIS DECK WILL EXECUTE AND RUN THE PROCESSOR VERSION *
*    *                    PROCO1                            *
*    *              FROM AN ABSOLUTE BINARY FILE            *
*    *          NO CODE MODIFICATIONS CAN BE MADE           *
*    ********************************************************
*
*  SETTING GO TO 'CHECK'H WILL CAUSE EACH JOBSTEP TO TERMINATE AFTER
*  INPUT CONTROL PARAMETERS ARE READ AND CHECKED FOR SYNTAX AND MOST
*  SIMPLE ERRORS.  SUCH A CHECKOUT MAY BE RUN IN EXPRESS CLASS.
*
*SET(GO='CHECK'H)
*
*  SET G1 TO THE NUMBER OF JOBSTEPS DESIRED (I.E., THE NUMBER OF
*  PROCESSOR EXECUTIONS).  THERE MUST BE A SEPARATE, COMPLETE SET
*  OF INPUT CONTROL PARAMETERS (DATA CARDS) FOR EACH JOBSTEP.
*
SET(G1=1)
*
* INITIALIZE G2 (JOBSTEP COUNTER)
SET(G2=1)
*
* INITIALIZE G3 (NUMBER OF PLOT FRAMES IN DATASET $PLT NOT YET DISPOSED)
SET(G3=0)
*
* INITIALIZE G4 (CURRENT PLOT DISPOSE GROUP NUMBER)
SET(G4=1)
*
* GET ABSOLUTE IMAGE FROM PSTORE
PCOPY ( FROM=/TB/CCMPROC/PROCO1.PRG TO=ABPROC )
*
* GET $DEBUG FROM PSTORE
PCOPY ( FROM=/TB/CCMPROC/PROCO1.DBG TO=$DEBUG)
*
* EXECUTE THE PROCESSOR
* THIS IS DONE ITERATIVELY FOR MULTIPLE JOBSTEPS
*
LOOP.
  EXITLOOP(G2.GT.G1)
  ABPROC.
  SET(G2=G2+1)
ENDLOOP.
*
* APPEND INDEX AND $PLT TO DPLT AND DISPOSE IT
* (BUT ONLY IF $PLT CONTAINS UNDISPOSED FRAMES)
IF(G3.GT.0)THEN
  PRSIM,I=PINDEX,O=PLTD.
  REWIND,DN=$PLT.
```

```
  COPYD,I=$PLT,O=PLTD.
  REWIND,DN=PLTD.
  NETDISP,DN=PLTD,NA,DF=BI,MF=D1,MDS,DC=PT,NOACK,^
          TITLE='PROCO1 TEST'.
ENDIF.
*
EXIT.
*
* PRODUCE A SYMBOLIC DUMP AFTER AN ERROR ABORT
DUMPJOB.
DEBUG,BLOCKS,PAGES=50.
*
* DISPOSE OF PLOT FILE(S) AFTER AN ERROR ABORT
*
* APPEND INDEX AND $PLT TO DPLT AND DISPOSE IT
* (BUT ONLY IF $PLT CONTAINS UNDISPOSED FRAMES)
IF(G3.GT.O)THEN
  PRSIM,I=PINDEX,O=PLTD.
  REWIND,DN=$PLT.
  COPYD,I=$PLT,O=PLTD.
  REWIND,DN=PLTD.
  NETDISP,DN=PLTD,NA,DF=BI,MF=D1,MDS,DC=PT,NOACK,^
          TITLE='PROCO1 TEST - ABORT'.
ENDIF.
*
EXIT.
*
\EOF
C
C INPUT CONTROL PARAMETERS - JOB STEP 1
C
C SPECIFICATION OF CASE A
C
 TITLEA='MODULAR PROCESSOR TEST RUN'
 TAPESA='TNEWOB'
 DAYSA=O.
 FIELDA1='T','Z'
C
 PRESSLEV=900.,850.,700.,600.,500.,400.,300.,200.,100.
 ZONAVG='YES'
 HPROJ='RECT'
C
 ENDOFDATA
\EOF
```

This job deck contains references to five Cray global pseudo registers: GO through G4. These registers provide a means of communication between different jobsteps (code executions) within a single job submittal. The SET statements for GO and G1 may be modified by the user (see next paragraphs)

and are not changed by the Processor code. G2, G3, and G4 are initialized by the JCL and maintained by the Processor code; their SET statements should not be modified by the user.

GO may be set so that the Processor run does nothing more than read and check the Input Control Parameters for all jobsteps. To activate this option, uncomment (delete the asterisk in column 1) the SET statement for GO. Most (but not all) ICP errors will be caught by this checkout. Such a checkout may be run in express class with a short time limit.

The Processor is executed from within a JCL loop over the number of jobsteps (*DEFS). The global pseudo-register G1 should be set to the number of jobsteps desired; for each jobstep, there must be a complete set of ICP data card images, terminated with an ENDOFDATA card (no intervening \EOF cards).

The microfilm output is conditionally disposed by the JCL in two places: before the EXIT card for a normal termination, and after the EXIT for aborted runs (some frames may have been produced before the run aborted). These disposes are executed only if $PLT contains plot frames not already disposed by the Processor code (*PDIS). The global pseudo-register G3 is maintained by the Processor code for use in this test. These NETDISPs may be changed to DISPOSEs (for running express class), or modified for specific needs (see Computing Division documentation).

For those who must modify the Processor code, the recommended way of running the Processor is from the relocatable binary PSTORE file (*ARCH). The basic procedure is to compile only the part of the code being modified, and substitute the new object code for the corresponding code in the relocatable binary file. One way to do this is illustrated by the sample job deck listed below (this job deck is available as the file CCMPROCR JOB on the CSMLIB 191 disk).

```
JOB,JN=nnnnnn,US=uuuupppppppp,T=120,OLM=300,*TB,*D1.
*
*     ********************************************************
*     * THIS DECK WILL EXECUTE AND RUN THE PROCESSOR VERSION *
*     *                    PROCO1                            *
*     *             FROM A RELOCATABLE FILE                  *
*     *      NEW SUBROUTINE VERSIONS MAY BE SUBSTITUTED      *
*     ********************************************************
*
*  SETTING GO TO 'CHECK'H WILL CAUSE EACH JOBSTEP TO TERMINATE AFTER
*  INPUT CONTROL PARAMETERS ARE READ AND CHECKED FOR SYNTAX AND MOST
*  SIMPLE ERRORS.
*
*SET(GO='CHECK'H)
*
*  SET G1 TO THE NUMBER OF JOBSTEPS DESIRED (I.E., THE NUMBER OF
*  PROCESSOR EXECUTIONS).  THERE MUST BE A SEPARATE, COMPLETE
*  SET OF INPUT CONTROL PARAMETERS (DATA CARDS) FOR EACH STEP.
*
SET(G1=1)
*
* INITIALIZE G2 (JOBSTEP COUNTER)
SET(G2=1)
*
* INITIALIZE G3 (NUMBER OF PLOT FRAMES IN DATASET $PLT NOT YET DISPOSED)
SET(G3=0)
*
* INITIALIZE G4 (CURRENT PLOT DISPOSE GROUP NUMBER)
SET(G4=1)
*
* GET UPDATE PROGRAM LIBRARY FROM TBM
ACQUIRE,PDN=PROCO1,DN=PROC,DF=TR,MF=TB.
*
* GET LATEST MOD FILE FROM CSMLIB MACHINE ON IBM
* SEE CCM NEWS FOR THE NAME OF THE LATEST FILE
* REPLACE $$$$$$ WITH A NAME LIKELY TO BE UNIQUE (TWO PLACES)
*
NETAQR,DN=$$$$$$,MF=IO,DF=CH,USER=CSMLIB,FLNM=filename,^
       FLTY=PRMODS,FORCE.
*
* RUN SELECTED UPDATED DECK(S) THROUGH UPDATE
UPDATE,I=$IN,P=PROC,N=0,S=0,C=$CPL,*=$,ID,ML=1,Q=deck1:deck2.
*
* COMPILE NEW CODE
CFT,I=$CPL,B=PRELOC,ON=AZ.
*
* GET RELOCATABLE BINARY FROM PSTORE
PCOPY ( FROM=/TB/CCMPROC/PROCO1.OBJ TO=RBPROC )
*
* CREATE EXECUTABLE IMAGE
```

```
LDR,DN=PRELOC:RBPROC,SET=INDEF,AB=ABPROC,NX.
*
* EXECUTE THE MODIFIED PROCESSOR CODE
* THIS IS DONE ITERATIVELY FOR MULTIPLE JOBSTEPS
*
LOOP.
  EXITLOOP(G2.GT.G1)
  ABPROC.
  SET(G2=G2+1)
ENDLOOP.
*
* APPEND INDEX AND $PLT TO DPLT AND DISPOSE IT IF $PLT CONTAINS
* UNDISPOSED FRAMES
IF(G3.GT.0)THEN
  PRSIM,I=PINDEX,O=PLTD.
  REWIND,DN=$PLT.
  COPYD,I=$PLT,O=PLTD.
  REWIND,DN=PLTD.
  NETDISP,DN=PLTD,NA,DF=BI,MF=D1,MDS,DC=PT,NOACK,^
          TITLE='PROCO1 PLOTS'.
ENDIF.
*
EXIT.
*
* PRODUCE A FORMATTED DUMP AFTER AN ABORT
DUMPJOB.
DEBUG,BLOCKS,PAGES=50.
*
* DISPOSE OF PLOT FILE(S) AFTER AN ERROR ABORT
*
* APPEND INDEX AND $PLT TO DPLT AND DISPOSE IT IF $PLT CONTAINS
* UNDISPOSED FRAMES
IF(G3.GT.0)THEN
  PRSIM,I=PINDEX,O=PLTD.
  REWIND,DN=$PLT.
  COPYD,I=$PLT,O=PLTD.
  REWIND,DN=PLTD.
  NETDISP,DN=PLTD,NA,DF=BI,MF=D1,MDS,DC=PT,NOACK,^
          TITLE='PROCO1 PLOTS - ABORT'.
ENDIF.
*
EXIT.
*
\EOF
$/
$/ READ NETAQRED MOD FILE
$READ $$$$$$
$/
$/ CRAY UPDATE MODS GO HERE (START WITH $ID)
$/
```

```
\EOF
C
C INPUT CONTROL PARAMETERS - JOB STEP 1
C
C SPECIFICATION OF CASE A
C
 TITLEA='MODULAR PROCESSOR TEST RUN'
 TAPESA='TNEWOB'
 DAYSA=0.
 FIELDA1='T','Z'
C
 PRESSLEV=900.,850.,700.,600.,500.,400.,300.,200.,100.
 ZONAVG='YES'
 HPROJ='RECT'
C
 ENDOFDATA
\EOF
```

This run deck is similar to the previous one, except that it has some additional JCL for handling the modifcations using Cray Update. First, the ACQUIRE gets the Update Program Library from the TBM. This Program Library contains all of the Processor code (*ARCH). Next, the NETAQR gets the latest mod file from the CSMLIB machine on the IBM. Note that the NETAQR prevents the use of express class. This mod file does not need to be applied, however, unless it contains mods to the Update deck(s) being modified by the user. Since NETAQR creates a Permanent Dataset on the Cray disk, it is best to use a unique name (replace $$$$$$ in two places) for this dataset in order to avoid conflicts. Update is then run, applying all mods to the deck or decks being modified (change the strings "deck1", "deck2", etc. on the UPDATE card). The modified code is then compiled, and a new executable image created. The loader will generate logfile messages for all replaced routines. All Update mods go between the first two \EOF cards.

An alternate method of modifying the Processor code is to DISPOSE the selected Update Deck(s) to a Network node, extract the individual subroutine(s) to be modified, and use an interactive editor to make the changes. The modified routine(s) can then be NETAQRed or added to the job deck, and compiled as in the sample deck above. The disadvantage of this method is that it is much harder to keep abreast of standard Processor modifications.

->ORDR - Order of Processor operations

This topic describes the relationships between various processing options by outlining the order of all possible operations within the major loops over time, field passes, and Comparison Cases (*DEFS). This loop structure (and therefore the order of operations) depends on whether gridpoint time averaging (*TAVG) and/or time filtering (*TFIL) have been requested; the status of these options may also limit the allowed combinations of some

other options. The four possibilities are therefore outlined separately on the following pages; they are:

1) no gridpoint time averaging, no time filtering (*ORNN),
2) gridpoint time averaging, no time filtering (*ORAN),
3) gridpoint time averaging, time filtering (*ORAF), and
4) no gridpoint time averaging, time filtering (*ORNF).

Each outline indicates the order of operations when all possible options are requested; if an option is not requested, that step in the sequence is simply skipped. The sequences of operations for spectral processing (*ORSP) and spatial averaging and related plotting (*ORSA) are basically the same for all four possibilities.

There are two columns of information on the left side of each outline. The column containing A's, B's and C's indicates which comparison Case(s) may be operated on in that step. The column containing the "+", "-", and "$" symbols indicates what effect (if any) that processing step has on the data processed by subsequent steps. Each step may add ("+"), delete ("-"), or change ("$") fields. The absence of such a symbol indicates that that processing step does not affect subsequent processing steps.

Although the order of operations is fixed by the Processor code, reordering them usually requires only simple modifications. With few exceptions, each line of this outline corresponds to a subroutine call in the Processor's Driver. The various option combinations for time averaging and time filtering are somewhat compressed in the code, but the sequences of spectral operations and spatial averaging are implemented as separate sub-drivers, just as they are outlined here.

-->ORNN - Order for no gridpoint time averaging, no time filtering

```
                begin time loop
                begin field pass loop
                begin AB Case loop

+   AB   1. read next day from input tape (*DATA)
+   AB   2. compute type 11, 12, 13, or 14 derived fields (*DFLD)
    AB   3. reorganize memory (*CMEM)
+$  AB   4. spectral operations (*ORSP)
$   AB   5. surface type masking (*MASK)
+   AB   6. compute type 31 derived fields (*DFLD)
$   AB   7. vertical interpolation (*VERT)
+   AB   8. compute type 41 derived fields (*DFLD)
+$  AB   9. spatial averaging and related plotting (*ORSA)
    AB  10. append this time to time series for plotting (*PTIM)
    AB  11. write this time to History Save Tape (*OHST)
    AB  12. write this time to Time Series Save Tape (*OTSR)

                end AB Case loop

+   C   13. compute Case differences, ratios (*COMP)
+$  C   14. spatial averaging and related plotting (*ORSA)
    C   15. append this time to time series for plotting (*PTIM)
    C   16. write this time to History Save Tape (*OHST)
    C   17. write this time to Time Series Save Tape (*OTSR)
    ABC 18. if last time in loop, plot all time series (*PTIM)
            and write Time Series Plot Save Tape (*OTSP)

                begin ABC Case loop

    ABC 19. write this time to Horizontal Slice Save Tape (*SHOR)
    ABC 20. print requested field values (*PRNT)

                end ABC Case loop
                end field pass loop
                end time loop
```

-->ORAN - Order for gridpoint time averaging, no time filtering

```
                begin field pass loop
                begin AB Case loop
                begin time loop

 +   AB    1. read next day from input tape (*DATA)
 +   AB    2. compute type 11, 12, 13, or 14 derived fields (*DFLD)
     AB    3. reorganize memory (*CMEM)
+$   AB    4. spectral operations (*ORSP)
 $   AB    5. surface type masking (*MASK)
 +   AB    6. compute type 31 derived fields (*DFLD)
 $   AB    7. vertical interpolation (*VERT)
 +   AB    8. compute type 41 derived fields (*DFLD)
     AB    9. append this time to time series for plotting (*PTIM)
     AB   10. write this time to History Save Tape (*OHST)
     AB   11. write this time to Time Series Save Tape (*OTSR)
 +   AB   12. accumulate zonal eddy statistics (*ZEST)
 +   AB   13. accumulate time average statistics (*TAVG)

                end time loop

+$   AB   14. spatial averaging and related plotting (*ORSA)
     AB   15. write Time Average Save Tape (*STAV)

                end AB Case loop

 +   C    16. compute Case differences, ratios (*COMP)
+$   C    17. spatial averaging and related plotting (*ORSA)
     C    18. write Time Average Save Tape (*STAV)
     AB   19. plot all time series (*PTIM)
              and write Time Series Plot Save Tape (*OTSP)

                begin ABC Case loop

     ABC  20. write this time to Horizontal Slice Save Tape (*SHOR)
     ABC  21. print requested field values (*PRNT)

                end ABC Case loop
                end field pass loop
```

-->ORNF - Order for no gridpoint time averaging, time filtering

```
                begin field pass loop
                begin AB Case loop
                begin time loop

 +  AB   1. read next day from input tape (*DATA)
 +  AB   2. compute type 11, 12, 13, or 14 derived fields (*DFLD)
    AB   3. reorganize memory (*CMEM)
+$  AB   4. spectral operations (*ORSP)
 $  AB   5. surface type masking (*MASK)
 +  AB   6. compute type 31 derived fields (*DFLD)
 $  AB   7. vertical interpolation (*VERT)
 +  AB   8. compute type 41 derived fields (*DFLD)

                steps 9-15 only done away from time series ends

 $  AB   9. filter this time (*TFIL)
 +  AB  10. compute type 51 derived fields (*DFLD)
    AB  11. append this time to time series for plotting (*PTIM)
    AB  12. write this time to History Save Tape (*OHST)
    AB  13. write this time to Time Series Save Tape (*OTSR)
+$  AB  14. spatial averaging and related plotting (*ORSA)
    AB  15. print requested field values (*PRNT)

                end time loop
                end AB Case loop

    AB  16. plot all time series (*PTIM)
            and write Time Series Plot Save Tape (*OTSP)

                end field pass loop
```

-->ORAF - Order for gridpoint time averaging, time filtering

```
                begin field pass loop
                begin AB Case loop
                begin time loop

 +  AB   1. read next day from input tape (*DATA)
 +  AB   2. compute type 11, 12, 13, or 14 derived fields (*DFLD)
    AB   3. reorganize memory (*CMEM)
 +$ AB   4. spectral operations (*ORSP)
 $  AB   5. surface type masking (*MASK)
 +  AB   6. compute type 31 derived fields (*DFLD)
 $  AB   7. vertical interpolation (*VERT)
 +  AB   8. compute type 41 derived fields (*DFLD)

                steps 9-14 only done away from time series ends

 $  AB   9. filter this time (*TFIL)
 +  AB  10. compute type 51 derived fields (*DFLD)
    AB  11. append this time to time series for plotting (*PTIM)
    AB  12. write this time to History Save Tape (*OHST)
    AB  13. write this time to Time Series Save Tape (*OTSR)
 +  AB  14. accumulate time average statistics (*TAVG)

                end time loop

 +$ AB  15. spatial averaging and related plotting (*ORSA)
    AB  16. write Time Average Save Tape (*STAV)

                end AB Case loop

 +  C   17. compute Case differences, ratios (*COMP)
 +$ C   18. spatial averaging and related plotting (*ORSA)
    C   19. write Time Average Save Tape (*STAV)
    AB  20. plot all time series (*PTIM)
            and write Time Series Plot Save Tape (*OTSP)

                begin ABC Case loop

    ABC 21. write this time to Horizontal Slice Save Tape (*SHOR)
    ABC 22. print requested field values (*PRNT)

                end ABC Case loop
                end field pass loop
```

-->ORSP - Order of spectral operations

```
 -  AB   1. exclude fields in requested exclusion list (*SPEC)
 $  AB   2. gridpoint to spectral transform (*SPEC)
 +  AB   3. compute type 21 derived fields (*DFLD)
 +  AB   4. compute type 211 derived fields (*VFLD)
 +  AB   5. compute type 22 derived fields (*DFLD)
 $  AB   6. spectral interpolation (*SINT)
 $  AB   7. spectral bandpass filtering (*SBND)
 +  AB   8. compute type 23 derived fields (*DFLD)
    AB   9. accumulate time averages for spectral graphing (*TAVG)
    AB  10. draw spectral graphics (*SPGR)
 -  AB  11. delete fields in requested delete list (*SPEC)
    AB  12. reorganize memory (*CMEM)
 $  AB  13. spectral to gridpoint transform (*SPEC)
 +  AB  14. compute type 24 derived fields (*DFLD)
 +  AB  15. add fields in requested exclusion list (*SPEC)
```

-->ORSA - Order of spatial averaging and related plotting

```
    AB   1. plotting of zonal eddy statistics (*ZEST)
    ABC  2. plotting of individual meridional cross sections (*PMEX)
    ABC  3. plotting of individual latitudinal cross sections (*PLAX)
 +  ABC  4. compute type 61 derived fields (*DFLD)
 $  AB   5. vertical averaging (*VAVG)
 +  ABC  6. compute type 62 derived fields (*DFLD)
    ABC  7. draw horizontal projection plots (*PHOR)
    ABC  8. horizontal (masked) area averaging (*HORA)
    ABC  9. zonal averaging (*ZONA)
 +  ABC 10. compute type 71 derived fields (*DFLD)
    ABC 11. plotting of zonal averages (*PMEX)
 -  ABC 12. delete zonal averages (*ZONA)
    ABC 13. meridional averaging (*MERA)
 +  ABC 14. compute type 81 derived fields (*DFLD)
    ABC 15. plotting of meridional averages (*PLAX)
 -  ABC 16. delete meridional averages (*MERA)
```

->SPCL - Special topics

The first few topics in this section are operational considerations; they describe the Processor's limitations on the amount of data which can be processed simultaneously (*MLIM), processing costs (*COST), conserving memory (*CMEM), interactions with the TBM and Cray disks (*MSDV), and error handling (*ERRS). The remaining topics deal with the Processor's handling of fields which have extra bottom levels (e.g., surface temperature) (*XBTL), and the use of a special set of contour levels for plotting vertical cross sections of mixing ratio (*MIXR).

-->MLIM - Memory limitations

Primary ICPs: [FIELDcn]

Secondary ICPs: <MEMCON>

The number of fields that can be processed in a single pass through the input tapes is limited by the amount of computer memory available. This restriction is imposed by the Processor's design, which requires that all spatial gridpoints of all fields being processed exist simultaneously in main memory. The use of memory in this way makes the Processor considerably more efficient because it minimizes the amount of disk I/O required.

One way around the memory limitation is to make multiple passes over the input tapes, and this can be done with a single execution of the Processor code. Since most fields may be processed independently of other fields, each pass can process a different set of fields, allowing a large number of fields to be processed, one set at a time.

Making multiple field passes is very similar to jobstepping (*DEFS). The main advantages of using multiple field passes are convenience, and the fact that time average Save tapes can contain more than one field pass (*STAV).

Dividing the fields to be processed into field pass subsets must be done by grouping the fields appropriately on the [FIELDcn] data card images. The optimum distribution is the one which minimizes the number of field passes without separating fields which must be processed together. The difficulty lies in determining how many fields can be processed in each pass without requiring too much memory. For most runs, a rough, conservative estimate of the number of "field-levels" which can be processed in a single pass can be obtained by dividing 220,000 by the number of grid points on a horizontal surface. A "field-level" is one level of one field; for example, a nine-layer Model defines temperature at nine levels in the free atmosphere, plus the surface, for a total of ten field-levels. For an R15 Model, which has 40 latitudes and 50 longitudes (including two wraparound points), there are 2,000 grid points on a horizontal surface. This implies that at least 110 field-levels can be handled in each field pass, for most processing operations. Remember, however, that this is a rough estimate and does not apply to all kinds of processing. Runs which do not compute time averages can often process more than twice this number, while time filtering operations may be more than an order of magnitude more restrictive. Also note that requesting derived fields (*DFLD) may cause additional fields to be carried through part of the processing. See (*CMEM) for a more complete discussion of memory usage.

The memory limitation problem is not solved by the field pass approach if the fields to be processed cannot be organized into independent groups that are small enough. Since this usually occurs only at very high spatial resolutions (usually higher than T63, which has 95 latitudes and 195 longitudes), it is not a common problem. The ICP <MEMCON>, which is

discussed in (*CMEM), may solve memory limitation problems which cannot be handled by the field pass approach.

-->COST - Processing costs

Primary ICPs: [FIELDcn]

Secondary ICPs: <SIGLEVc> <SURFLEV> <HPCINT> <HPVSCAL>
<MXCINT> <LXCINT>

Since most Processor runs dealing with short time series (less than 10 or 20 time samples) require less than a minute or two of Cray CPU time, it is usually not worth the effort of trying to estimate the running time before submitting a job. A major exception is runs which produce a large number of horizontal projection contour plots. Since drawing an R15, global contour plot requires roughly 1/3 second of CPU time, it may be necessary to estimate the plotting time, especially when requesting plots for all time samples in a series (i.e., if no time averaging is requested).

For runs which process long time series, the best way to estimate running time is to make a short run which processes 3 or 4 time samples without doing any graphics, then extrapolate to the full number of time samples and add a plot time estimate. Short test runs can also be very valuable in verifying that the desired processing steps have been properly requested. When extrapolating CPU times from short runs, be sure to account for startup overhead, especially when the executable file on PSTORE is not being used. The actual running time excluding overhead can be obtained from the Cray logfile column giving accumulated CPU time.

The remainder of this topic discusses a few simple operational procedures which can be used to reduce processing costs directly. The efficiencies of memory (*CMEM) and mass storage usage (*MSDV), which also affect cost, are discussed in separate topics.

Limiting processing to just those times and fields of interest is an obvious (but often overlooked) way to decrease the cost of processing. There are also a few options which may be used to limit the amount of data processed. When a field is requested, all existing vertical levels of that field are normally processed. It is possible, however, to request that processing be limited to a specified subset of Model sigma levels <SIGLEVc>; see (*LSIG) for a discussion. Different levels of different fields can be excluded by defining new derived fields which are single, specific levels of existing fields, and requesting only the derived fields (see (*UDFL) and the function .LEVELnn for <DERFLD>). It is also possible to inhibit the plotting of horizontal projections for specific levels of specific fields (*PHOR) <HPCINT> <HPVSCAL>, and specific fields can be excluded from vertical cross-section plotting in a similar way <MXCINT> <LXCINT>. Since drawing contour plots is one of the most CPU-intensive processing tasks, limiting the plotting in this way can sometimes result in a drastic cost reduction.

The use of Save tapes (*SAVT) can often significantly reduce the cost of a group of related processing jobs. Once partially-processed data are saved, subsequent runs can input the saved data and continue processing, without having to repeat the previous operations. For example, replotting the results of a long time average (perhaps with a different contour interval) is much more efficient if the time averages are saved by the run which first draws the plots (*STAV). If the full time series is needed for analysis, a partially-processed set of fields can be saved on a set of time series save tapes (*STSR) (*STSP).

Save tapes can also be used to reorder operations for efficiency. For example, there are two ways to compute time averages on pressure surfaces. Normally, each time sample is vertically interpolated, and averaging takes place on pressure surfaces. But if time averages are computed on Model sigma surfaces and saved, a subsequent run can interpolate to pressure surfaces using the time averaged locations of the sigma levels. Although these operations are not equivalent, the difference is not significant for many purposes.
Note: In order to interpolate time averaged data to pressure surfaces, surface pressure must be explicitly time averaged and saved (*VPRS).

-->CMEM - Conserving memory

Primary ICPs: [FIELDcn] [MEMCON] [MEMORY]

Secondary ICPs: <SIGLEVc> <SURFLEV>

Optimizing memory usage for specific Processor tasks may make possible some operations which would otherwise require too much memory, and it can significantly decrease the cost of processing. Conserving memory can lower costs directly because the Cray charging algorithm is based, in part, on the average memory used by a job, and it can also lower costs indirectly because it may allow the number of field passes to be reduced (*MLIM).

Reducing the amount of data processed is the easiest way to conserve memory. See (*COST) for a discussion of some ways to do this [FIELDcn] <SIGLEV> <SURFLEV>. There are several additional memory conservation techniques which can be switched on with the ICP [MEMCON]. This ICP can be used without completely understanding how it works, but it should be used with caution, because it always increases the amount of disk I/O performed. It is recommended that short test runs be made before using this option on costly runs, since it is possible that its use could increase the cost of the run. Note, however, that I/O costs under the current Cray charging algorithm are highly variable. In order to describe how [MEMCON] conserves memory, it is first necessary to describe how memory is managed in the Processor. The following paragraphs give a brief outline.

Each processing operation is performed by a different Processor code Module. For example, one Module reads the data from the input tapes, another Module vertically interpolates the data, another Module time

averages the data, etc. In general, each Module inputs data, transforms it, and outputs the results, using main memory to store both the "input" and "output" data. Most Modules process all fields at all spatial gridpoints with a single call, but only for one Model time; a series of Module calls is executed within a time loop in a "driver" routine.

If all Modules had to have memory statically allocated for their output data before execution began, the number of Modules which could be strung together would be very limited. Instead, memory is managed dynamically, i.e., each Module requests space for its output data only when that Module is executed. When the data are no longer needed, the space is returned to a pool for reuse by other Modules at a later time.

Sometimes all of a Module's input data must coexist with all of the output data, but more often it is possible to intersperse freeing of input data memory with allocation of output memory, one block at a time, effectively overwriting the input data with the results of the operation. The amount of memory needed by a Module depends not only on the amount of data being processed, but also on the type of operation being performed. The vertical interpolation Module, for example, needs only enough memory to hold one set of data at all spatial gridpoints, since it overwrites the input data latitude by latitude. Time averaging, however, requires simultaneous access to a time sample of input data, plus an accumulator - twice the memory needed for one "copy" of all fields at all spatial grid points. Spectral operations also comsume a lot of memory; large blocks of extra space are needed in order to vectorize the FFT transforms. Time filtering operations generally require the most amount of memory; applying a filter with a width of 31 points requires that 31 time samples be stored concurrently. In addition, the results of some operations may be needed for other processing later, and memory requirements may be cumulative over several processing steps. One example of this effect is the computation of derived fields (*DFLD); in order to compute some derived fields, additional derived fields must be computed at earlier stages of processing (they may be added automatically) and carried through to the point where they are used (they may then be deleted automatically).

The point at which a Processor job requires the maximum amount of memory is usually during the single operation which requires the most memory, but there are some complicating factors, and this is not always the case. The alternate allocation and deallocation of blocks of memory may result in significant fragmentation (i.e., the creation of many small, free blocks between allocated blocks). This is particularly likely if the blocks being allocated are a different size than those being freed (e.g., when vertically interpolating to a number of pressure levels different than the number of original sigma levels). Fragmentation may get worse with successive iterations of a fixed series of operations. For example, it is possible to run out of memory in the middle of a time average run, even though the same sequence of operations has been completed successfully many times. The fragmentation can be eliminated by writing the data to a temporary disk dataset, freeing all allocated blocks, then reading the data back into

memory. The [MEMCON] option performs this memory "reorganization" in the following places (*ORDR): 1) immediately after reading the Input Control Parameters (space for ICPs is also dynamically managed), 2) immediately after reading data from the input tape, and 3) preceding transformation of spectral data back into gridpoint space (a few large blocks of memory are needed here). A message is printed in the output file each time memory is reorganized.

In addition to the memory reorganization, [MEMCON] controls the temporary storage of units of data on disk while other, independent computations are made. This is currently done with two data units: 1) the time average statistics accumulator, and 2) data which is explicitly being excluded from spectral operations <SPCEFc>. The amount of memory conserved by the [MEMCON] option is particularly significant when time averaging spectrally processed fields.

In general, it is impractical to attempt to determine beforehand exactly how much memory is required for a given set of operations (but see (*MLIM) for a rule-of-thumb). It is usually necessary to make short trial-and-error test runs when trying to approach the memory limit. In some cases it may be desirable to specify a value for the ICP <MEMORY>. This is the size of the initial memory allocation, which defaults to zero. If a lot of memory will be needed, specifying a large value will reduce system overhead by reducing the number of times the job's field length must be increased. This will improve system efficiency and might decrease turnaround time, but cost benefits are probably not significant. Since the field length is always increased as needed, use of this ICP is never necessary, and using it is NOT likely to cure memory limitation problems.

The Cray Heap Manager is used for allocating memory dynamically (see the Cray Library Reference Manual). Several Heap parameters, including the initial size and allocation increment, can be controlled by the Cray loader (see LDR options in the COS Manual). This kind of control is not possible when running from the executable PSTORE file, which was created using the loader defaults (*ARCH).

-->MSDV - Mass storage data volumes

Primary ICPs: none

Secondary ICPs: <DELREL> <DELPDN>

The movement of data between the TBM and the Cray disks is handled automatically by the Processor code, but at times it may be advantageous to assume some degree of control. There are basically three ways in which this can be accomplished: 1) through the use of Input Control Parameters (ICPs), 2) manipulating datasets directly (adding Cray JCL to the run deck, for example), and 3) modifying the Processor code. The management of input datasets for a typical run is described below. A basic understanding of both Cray and TBM data management is assumed.

When execution reaches the point at which a particular volume is to be read, there is a check to see if the volume is currently accessed by the job. If not, ACQUIRE is called, initiating a TBM stage-up if the volume is not already on the Cray disk. After a successful return from ACQUIRE, the volume exists on the disk as a Cray "dataset", and is "accessed" by the job. The dataset is then ASSIGNed to a FORTRAN logical unit number. This unit number is usually the same for all datasets for a particular comparison Case (*COMP). If another dataset is already ASSIGNed to the unit, that dataset is CLOSEd. This means that a maximum of two primary input datasets (one from Case A and one from Case B) can be open simultaneously. No other dataset operations are performed (other than reading and positioning) until it is determined that the next volume in the series (for that Case) is to be read. If Comparison Cases (*COMP) A and B share the same input volume, the dataset management is a bit more complex, since both Cases may read from the same dataset.

When a dataset is CLOSEd so that its associated unit number can be reassigned, it may also be RELEASEd, and may be DELETEd as well, depending on the value of the ICP <DELREL>. If the dataset is RELEASEd, it is subject to purging from the disk by the scrubber before the job terminates. This is not very likely, however, and RELEASEing datasets helps to prevent problems associated with a shortage of disk space. The advantage of DELETEing the dataset is that the disk space is immediately freed, and so is available for a subsequent volume stageup from the TBM (and, of course, the need for a later purge by the Cray disk scrubber is eliminated). The disadvantage is that if the volume is needed again later, it will have to be staged up from the TBM.

In general, ACQUIREing volumes one at a time minimizes the number of TBM stage-ups required. There are times, however, when this procedure can cause wasted CPU time and unnecessary stage-ups. If a large number of volumes must be read before any Processor output is generated (e.g., when computing a long time average), stage-up problems encountered near the end of the time series usually mean the entire job must be rerun. This problem can be avoided by staging up all of the needed volumes before the Processor begins execution. The necessary ACQUIREs can be added to the Processor run deck, or they can be submitted as a separate job. In either case, use of the multiple ACQUIRE facility is recommended, especially if the volumes are on a dedicated TBM reel (see Computing Division documentation for MAQR). There is still a danger, however, that the volumes will be purged from the Cray disks before the processing job terminates, especially if the job is run at low priority. There is also the possibility that the Processor run will abort due to an incorrect deck setup, and the volumes purged before the job can be rerun.

The Processor also has the capability to force a stage-up from the TBM for any or all volumes read. As specified by the ICP <DELPDN>, selected "permanent" datasets are deleted from the Cray disk before processing begins. This is useful if the integrity of a Cray disk dataset has been compromised (provided you still have some degree of confidence in the copy

on the TBM), or if a TBM volume has been updated, and the old version still exists on the Cray disk.

When a new dataset is ASSIGNed to a FORTRAN logical unit, that dataset is opened and a memory buffer is set aside for all its I/O. In order to minimize I/O wait time, the size of this buffer should be at least several times larger than the average logical record size. The Processor sets this buffer size appropriately for history tapes output by Models with resolutions up to about R30, 9 vertical levels. Users who frequently process significantly higher resolutions (vertical as well as horizontal) may want to increase the size of this buffer (variable IBS, set in a DATA statement in subroutine POSITD in Module RDHT).

The Processor's handling of input datasets is designed primarily for efficient I/O (and minimal use of memory) when long time averages are computed for individual Model runs. Although it is possible to compute ensemble means of multiple Model runs (*ENSB) without modifying the code, this may involve a lot of unnecessary dataset positioning because the same FORTRAN logical unit is used for each Model run in the ensemble. (Each input tape must be positioned from the beginning for each time sample read from it). The Processor code can be modified (without too much difficulty) so that a separate unit is used for each case in the ensemble, allowing each set of Model tapes to be positioned independently. This has the disadvantage, however, of requiring a separate memory buffer for each case in the ensemble.

Except for the graphics metacode (*PDIS), all output datasets are DISPOSEd by the Processor code without waiting to make sure the DISPOSE was successful.

-->ERRS - Error handling

Each successful execution of the Processor code results in a printed message beginning with "*** NORMAL TERMINATION FOR JOBSTEP ..." followed by plot frame statistics. Successful completion of all requested jobsteps results in the printing of the message "*** NORMAL RUN TERMINATION ***". If this message is not printed, then something has gone wrong. Note however, that the printing of this message does not guarantee that output datasets (including plot files) have been successfully disposed.

Most execution time errors are detected by the Processor code and result in: 1) an error message printed in the output file ($OUT on the Cray), 2) an entry in the logfile indicating a fatal error was detected by the Processor, 3) an attempt to dispose the plot file(s) (*PDIS), and 4) a call to ABORT. On the Cray, a call to ABORT results in a subroutine traceback in the logfile, and causes a jump to the next EXIT in the JCL file. Following the EXIT, all standard Cray Processor decks contain JCL which generates a symbolic dump of all routines in the traceback, plus the common blocks in these routines. Any error messages printed by the Processor or its utilities will immediately precede this dump. When examining the output

file online, the beginning of the dump can be located by searching forward for the string "SYMBOLIC". If there is no error message preceding the dump, then the error was almost certainly detected by the Cray Operating System rather than the Processor code, and the logfile should contain a system error message. The most common errors in this category involve problems associated with staging up TBM volumes (e.g., "DATASET NOT AVAILABLE FROM STATION", checksum errors, etc.), and the subroutine ACQUIRE will appear in the traceback. Another common error is "FLOATING POINT ERROR". This can be caused by an improper processing request or bad data on an input tape (possibly related to TBM or Cray disk problems), but it can also be caused by an error in the Processor code (please save the dump!). The subroutine traceback, plus the printout preceding the dump, will usually indicate roughly where the error occurred; this may be enough to locate some simple problems.

Most fatal errors recognized by the Processor code or its utilities are the result of improper processing requests, and the resulting error messages are usually self-explanatory. A frequent source of such error messages is the input utility READLX which reads all Input Control Parameters. ICP syntax errors are usually flagged immediately, with a pointer attempting to indicate the exact column of the error. The most common error detected by READLX is "UNDEFINED KEYWORD", which usually means a keyword has been mispelled, a keyword suffix has been used improperly, or a character string value is not enclosed in single quotes or preceded by nH (*SICP). After all ICP's have been read, there are additional checks on the values input. This is done by the main Processor code, and results in messages that are usually self-explanatory. When these messages refer to the number of values input, note that strings containing more than eight characters are treated as multiple values, eight characters per value.

Since memory is managed dynamically, it is possible to get partway through processing before requiring more memory than is available. See (*MLIM) if the message "NOT ENOUGH MEMORY AVAILABLE" is printed. This message is followed by memory management statistics and maps which may help in formulating a solution to the problem, but a good understanding of the Processor's internal data structure is required in order to make use of this information. (See (*CMEM) and documentation of the Cray Heap in the Library Reference Manual). The easiest way around this problem is to simply decrease the amount of data processed in each field pass (*DEFS). A few other error messages relating to memory management may be printed; these are almost always due to code errors, and are so flagged.

A large class of error messages is produced by the Data Unit Manager, which is the interface between the main Processor code and the dynamic memory manager. These error messages are characterized by "Unit" and "Record" "Names" and "Numbers". Such errors are almost always caused by problems in the code (please save the dump!).

-->XBTL - Extra bottom (surface) levels

The Processor has the capability to treat certain Model fields, defined at only one level, as extra bottom levels of other, multilevel fields. The association is made automatically, based on the following assumptions about Model field naming conventions. If the name of a single level field ends in an "S" or "G", and without the "S" or "G" matches the name of a multilevel field, then that single level field is assumed to be a "surface" or "ground" level of the multilevel field. The value of the ICP <SURFLEV> determines whether the extra bottom levels are to be searched for in this way. If found, these extra bottom levels become part of the multilevel field. The "ground" level is considered to be lower than the "surface" level, which is considered to be lower than all free atmosphere levels. Both extra bottom levels are assigned the Model vertical coordinate value SIGMA=1. These levels are never vertically interpolated, and the level character string associated with them is always 1000.S or 1000.G. Extra bottom levels may also be processed separately (as single level fields) by requesting them explicitly with their full names. Extra bottom levels are not included in vertical cross-section plots if the vertical coordinate is pressure.

-->MIXR - Special contour levels for mixing ratio

The NCAR Plot Package routine CONREC has been modified to allow the specification of unequally spaced contour levels. This capability is currently rather crude, and its use in the Processor is limited to a fixed set of contour levels used in plotting vertical cross sections of the mixing ratio (field names MIXRAT or Q). These contour levels are linearly spaced for values larger than 1.E-3, and logarithmically spaced for smaller values. Unless a large scale factor is specified using the ICP <MXSCAL>, all contour lines for small values will be labelled 0. (1.E8 is usually adequate).

->DATA - Specification of data to process

This section describes the input of Model history tapes (*IHST), and the input and output of Processor Save tapes (*SAVT).

-->IHST - Input of History tapes

Primary ICPs:	[TAPESc]	[TYPEc]	[DAYSc]	[FIELDcn]	
Secondary ICPs:	<TITLEc>	<ENSMBLc>	<SIGLEVc>	<SURFLEV>	<REORDER>
	<DELPDN>	<DELREL>			

* Either CCMOB or CCMOA (grid point format only) history tapes may be input. The Model type is not determined automatically; it must be specified or allowed to default [TAPESc] [TYPEc]. History Save Tapes written by the Processor (*OHST) are always in CCMOB format.

* Certain tapes which contain data on pressure surfaces, and having basically the same format as history tapes output by the Model, can also be input [TYPEc].
* Both packed and unpacked tapes are handled automatically.
* Different space and time resolutions are handled automatically. Model resolutions higher than about T63 (95 latitudes and 190 longitudes) impose some option limitations and/or require special handling techniques due to memory limitations on the Cray-1 (*MLIM).
* Any set of Model output times can be processed from any set of tapes; tape positioning is automatic both forwards and backwards, including across tape boundaries.
* Model times to process may be requested by Model day value(s), and/or any of the following: 1) first day on the first tape, 2) last day on the last tape, 3) the next day following the day just processed [DAYSc].
* Any subset of Model fields may be processed [FIELDcn].
* Any subset of Model sigma levels may be processed (same set for all fields) (*LSIG).
* Surface levels are (optionally) automatically associated and processed with the corresponding multilevel fields <SURFLEV>.

--->LSIG - Limiting input sigma levels

Primary ICPs: [SIGLEVc]

Secondary ICPs: <TYPEc>

It is possible to process only those Model sigma levels contained in an arbitrary list of layers [SIGLEVc] (same set for all fields). Only fields read from history tapes <TYPEc>, (and fields derived from them (*DFLD)) are limited. This option is not available for input from Save tapes.

If a field is defined at full levels (layer midpoints), the levels processed are those located at the midpoints of the requested layers (*DEFS). For fields defined at half-levels (layer interfaces), the levels processed are those at the bottom of the requested layers. For example, layer 1 for geopotential height refers to the geopotential height at ground elevation. All derived fields computed during history tape input (*DFLD) are computed for all Model layers before being limited. This simplifies the computation of these fields, and means that the values of derived fields which require vertical integration (such as geopotential height) are not affected by the limiting process.

Since the limiting occurs latitude by latitude as the data are being read from the input history tape(s), the use of this option can significantly increase the number of fields which can be processed in one field pass (but not the total number of field-levels) (*MLIM).

All other options are available for subsequent processing of the sigma level subset. Note, however, that if the subset is vertically interpolated, the results may be be different than interpolation based on the full set of

Model levels. Processing of extra bottom levels (*XBTL) is not affected by this option. The surface level of fields located at half-levels (such as geopotential height) is NOT considered to be an extra bottom level when counting levels for this ICP.

-->ENSB - Case ensemble processing

Primary ICPs: [ENSMBLc] [TAPESc] [DAYSc]

Secondary ICPs: <TIMAVGc>

In some experiments, the Model is run a number of times over the same time period - with different initial conditions, for example. It may be desirable to process such a set of runs as a group. This is referred to as "case ensemble processing", with each Model run being a separate case, and all cases together forming the ensemble.

When case ensemble processing is requested [ENSMBLc], the relationship between the requested days list [DAYSc] and the requested tapes list [TAPESc] is modified. Instead of treating the list of tapes as a progression in time (where each day is assumed unique), each tape is assumed to represent a different Model case, and each requested day is processed from each requested tape. Note that the use of a simple, one-dimensional list of input tapes restricts case ensemble processing to one input tape per case.

When time averaging is also requested (*TAVG) <TIMAVGc>, the "time average" statistics computed are actually case ensemble statistics, where each requested day is grouped (averaged) with the same day from each of the specified input tapes.

An ensemble case should not be confused with a comparison Case (*COMP), which is totally independent. In fact, it is possible to define one ensemble average as comparison Case A <TAPESA> <DAYSA>, another ensemble average as Case B <TAPESB> <DAYSB>, and compute the difference between the two ensemble averages. Case ensemble processing is implemented in a way which minimizes the amount of memory needed, but this results in inefficient dataset positioning. See (*MSDV) for a discussion.

-->PRNT - Printout of field values

Primary ICPs: [PRINTc]

Secondary ICPs: <PRFNMc> <PRLATc> <PRLONc> <PRLEVc> <PRLIMR>

Field values may be printed (i.e., written to the output file) [PRINTc] for a subset of fields <PRFNMc> at a subset of grid points <PRLATc> <PRLONc> <PRLEVc>. The printing is done ONLY for field pass 1. Printout may be further limited by a simple test on the field values <PRLIMR>. Also see (*COMP) for a description of the capability to print only significant

differences between two Cases. CAUTION: This option may produce a large amount of output since there are at most two field values per line. A new line is started whenever a specification parameter (field name or spatial location parameter) changes, provided that parameter has been varying. Specification parameters are varied in the following order: field name, latitude, longitude, level. The actual location parameters are printed along with each field value (values are printed for the grid point closest to the requested locations). Field values are printed with a resolution of seven decimal digits; the location is printed in terms of degrees latitude and longitude, plus the level word (the same character string used on plots).

The printout will reflect the results of all previous processing operations (*ORDR). If the data have been averaged in one or more spatial dimensions, the location parameter(s) for the averaged dimension(s) will be the four-character string "AVRG".

-->SAVT - Save tapes

There are currently six different types of Processor Save tapes - Time Average (*STAV), Time Series (*STSR), Time Series Plot (*STSP), Horizontal Slice (*SHOR), History (*OHST), and Surface Type (*SFCT). Each of these Save tapes is described individually. Only Horizontal Slice Save Tapes (*SHOR) and History Save Tapes (*OHST) (*IHST) are suitable for input by programs other than the Processor.

--->STAV - Time Average Save Tapes

Time Average Save Tapes contain a single time average, plus associated time average statistics, for an arbitrary set of fields, at any spatial resolution. Time Average Save Tapes may be both Input (*ITAV) and Output (*OTAV) by the Processor. Multiple field passes are handled automatically, and the field pass groupings may be different in the input and output runs.

---->ITAV - Input of Time Average Save Tapes

Primary ICPs: [TAPESc] [TYPEc] [FIELDcn]

Secondary ICPs: <TITLEc> <DELPDN> <DELREL>

* An arbitrary number of Time Average Save Tapes may be input and processed as a time series. This series of time average statistics may itself be time averaged, producing "second order" (or higher) time average statistics [TAPESc] [TYPEc].
* Any subset of fields from the save tape may be processed [FIELDcn]. This subset may contain fields originally written in different field passes.
* Input levels cannot be limited (*LSIG).
* Input data can be vertically interpolated to pressure surfaces only if surface pressure is present on the save tape and is requested for processing [FIELDcn].

---->OTAV - Output of Time Average Save Tapes

Primary ICPs: [SAVTAVc] [TIMAVGc]

Secondary ICPs: <CVFLDcn> <SDFLDcn> <TCFLDcn> <PRFLDcn> <ZSTFLcn> <ZCVFLcn>

* A time average Save tape can be output only if time averages are requested [TIMAVGc].
* Each Case is requested separately and saved on a separate tape [SAVTAVc].
* All field passes are written to the same tape.
* All fields that are being processed at the time the tape is written are output, inluding all time average statistics <CVFLDcn> <SDFLDcn> <TCFLDcn> <PRFLDcn> <ZSTFLcn> <ZCVFLcn>.
* Processing may continue after the save tape(s) are written (*ORDR).

--->STSR - Time Series Save Tapes

Time series Save tapes contain a series of days for some set of processed fields. The time series may span multiple tapes, but each series can contain only one field pass (*DEFS). These tapes may be both input (*ITSR) and output (*OTSR) by the Processor. Time series Save tapes are especially useful when used in conjunction with time filtering (*TFIL).

---->ITSR - Input of Time Series Save Tapes

Primary ICPs: [TAPESc] [TYPEc] [FIELDcn] [DAYSc]

Secondary ICPs: <TITLEc> <DELPDN> <DELREL>

* Any subset of fields from the Save tape(s) may be processed [FIELDcn].
* Any set of times can be processed; tape positioning is automatic, including across tape boundaries [DAYSc] [TAPESc].
* Input levels cannot be limited (*LSIG).
* Input data can be vertically interpolated to pressure surfaces only if surface pressure is present on the save tape and is requested for processing.

---->OTSR - Output of Time Series Save Tapes

Primary ICPs: [SAVTSRc]

Secondary ICPs: <NDYTSRc> <PWDTSRc>

* All fields being processed in field pass number 1 (and only those) are output. All processed days are output.
* Each Case is requested separately and saved on a separate series of tapes [SAVTSRc]. Time Series Save Tapes cannot be produced for Case C in the same run as time averaging (*TAVG) or time filtering (*TFIL). There are no such restrictions on Cases A and B.

* The output time series may be of arbitrary length, on an arbitrary number of tapes. The maximum number of times to be written to each tape in the series is specified by <NDYTSRc>.
* Processing may continue after the Save tape(s) are written.

--->STSP - Time Series Plot Save Tapes

These tapes contain data generated specifically for plotting time series. The use of these tapes allows plot characteristics to be changed without reprocessing all of the original data. Time Series Plot Save Tapes may be both input (*ITSP) and output (*OTSP) by the Processor. The use of these tapes is currently restricted to field pass 1, but any or all comparison Cases (*COMP) may share the same tape.

---->ITSP - Input of Time Series Plot Save Tapes

Primary ICPs: [SAVTSPR]

Secondary ICPs: <TSPPLcn> <TSPZLcn> <TSPALcn> <TSPZCcn> <TSPMCcn>
<TSPFNPn> <TSPFPH> <TSLPASP> <TSLPSIZ> <TSZCASP>
<TSZCSIZ> <TSMCASP> <TSMCSIZ> <SAVTSPW> <SAVTSPR>
<CLTSLPc> <CLCTRGE> <CLCTRLT> <CLLABEL> <CLHIGHS>
<CLLOWS>

* A single Time Series Plot Save Tape may be input in addition to or instead of other input tapes. The data on this tape can be used only for producing time series plots, and are completely independent of any other input data or processing. If a Time Series Plot Save Tape is input [SAVTSPR], then time series plots cannot be produced for other input data.
* Although the data read from a Time Series Plot Save Tape cannot be modified in any way, the plot characteristics can be changed. These characteristics include: ordinate ranges and Case pairings for line plots <TSPPLcn> <TSPZLcn> <TSPALcn> <TSPFNPn> <TSPFPH>; contour intervals, scale factors, and dividing values for contour plots <TSPZCcn> <TSPMCcn>; aspect ratios, sizes, and colors for all plots <TSLPASP> <TSLPSIZ> <TSZCASP> <TSZCSIZ> <TSMCASP> <TSMCSIZ> <CLTSLPc> <CLCTRGE> <CLCTRLT> <CLLABEL> <CLHIGHS> <CLLOWS>.
* When a Time Series Plot Save Tape is input, the ICP data card images specifying the time series data <TSPPLcn> <TSPZLcn> <TSPALcn> <TSPZCcn> <TSPMCcn> should be the same (and in the same order) as the cards used in the run which wrote the save tape, except for the plot characteristics mentioned above.

---->OTSP - Output of Time Series Plot Save Tapes

Primary ICPs: [SAVTSPW]

Secondary ICPs: <TSPPLcn> <TSPZLcn> <TSPALcn> <TSPZCcn> <TSPMCcn>

* A Time Series Plot Save Tape can be written only in a run which produces time series plots <TSPPLcn> <TSPZLcn> <TSPALcn> <TSPZCcn> <TSPMCcn>. Save these ICP data card images; they will be needed again in the run which inputs the Save tape (*ISTP).
* All times being processed are written to the save tape [SAVTSPW]. Note that time filtering (*TFIL) reduces the number of days being processed (and therefore written to the tape).

--->SFCT - Surface Type Save Tapes

These tapes contain data defining the location of land, ocean, and sea ice with respect to Model grid points. Their purpose is to provide the Processor with this information in runs in which it is needed but not otherwise available (*HORA) (*PTAL). The surface type information is never needed when data from history tapes are being processed as Case A, or if no masking is requested for horizontal area averages. The surface type is assumed to be independent of Model time. Each Surface Type Save Tape can be used only in conjuction with data that has the same horizontal resolution; input Save Tapes are checked for consistency with the data being processed.

There are two Surface Type Save Tapes archived on the TBM for use with CCMOA or CCMOB R15 data. They are SR15JA and SR15JL, which contain sea ice distributions for January and July, respectively.

Surface Type Save Tapes are identical in format to Time Average Save Tapes (*STAV), so it is possible to process them as primary input in order to plot the surface type distribution. SFCT is the only field on these tapes; it has a value of -1. over open ocean, 0. over sea ice, and 1. over land.

---->ISFT - Input of Surface Type Save Tapes

Primary ICPs: [SFCTTAP]

Secondary ICPs: <TYPEc> <MSKFLcn> <TSPALcn>

An input Surface Type Save Tape must be specified if (and only if) surface type masking is requested for horizontal area averages (*HORA) (*PTIM) and the input data for Case A are not being read from a history tape <MSKFLcn> <TSPALcn> <TYPEc>.

---->OSFT - Output of Surface Type Save Tapes

Primary ICPs: [SFCTCRT]

Secondary ICPs: <TYPEc>

A Surface Type Save Tape can be created [SFCTCRT] only if a history tape is used as input <TYPEc>.

--->OHOR - Output of Horizontal Slice Save Tapes

Primary ICPs: [SAVHSLc]

Secondary ICPs: <SURFLEV> <SIGLEV> <PRESSLEV> <FIELDcn> <DAYSc> <ENSMBLc> <SHSLZAV>

The purpose of Horizontal Slice Save Tapes is to provide processed data, in a simple format, to other processing programs. Each record on the tape contains a horizontal slice of field values. There is no information describing the data, and these tapes cannot be processed further by the Modular Processor. All levels of all fields being processed in field pass 1 (and only those) are output (*DEFS). All data to be output must fit on a single tape.

All records contain only Cray floating point words, written with a BUFFER OUT statement. Field values at blocked points (*DEFS) are set to 1.E36. Each record is a horizontal slice of field values at a single level, in the following format:

* Longitude varies first, starting at the dateline (180 deg. E) and progressing to the east. The value at the first longitude is repeated as the last longitude value.
* Latitude varies next from south to north.

All levels for a field are output (one record each, bottom to top) before the next field is output. Surface levels may be present as the lowest level <SURFLEV>. The field-level ordering is indicated in the printout when the tape is written. All field-levels are output for the first time being processed, then all field-levels for the next time, etc. <DAYSc> <ENSMBLc>.

All records are the same length, and may be read into a two dimensional array with the first dimension equal to the number of longitudes plus one, and the second dimension equal to the number of latitudes.

The data may be zonally averaged before being written <SHSLZAV>, in which case there is only one longitude (the zonal average).

Since complete descriptive information is available in the subroutine which writes the data, any or all of this information can be added to the save tape by making simple coding modifications. Most aspects of data ordering can also be easily changed.

--->OHST - Output of History Save Tapes

Primary ICPs: [SAVHSTc]

Secondary ICPs: <NDYHSTc> <PWDHSTc> <BPHSTc> <PKHSTc>

A time series of processed data may be written to a set of Save tapes which have the same form as CCMOB History tapes. These tapes may be subsequently input to the Modular Processor (*IHST), other processing codes, or possibly the Model, if all assumptions made by the Model input code are met.

* The data are written on sigma surfaces unless they have been interpolated to pressure surfaces, in which case a pressure level history tape is written (*IHST).
* All fields being processed in field pass number 1 (and only those) are output. Input fields are written in the order in which they are requested; other fields are written in the order in which they are computed. The field ordering is summarized in the printed output. All processed days are output.
* Each Case is requested separately and saved on a separate series of tapes [SAVHSTc]. History Save Tapes cannot be produced for Case C in the same run as time averaging (*TAVG) or time filtering (*TFIL). There are no such restrictions on Cases A and B.
* The output time series may be of arbitrary length, on an arbitrary number of tapes. The maximum number of times to be written to each tape in the series is specified by <NDYHSTc>.
* The data written to the Save Tapes may be packed in the same manner as with history tapes output by the Model, except that all fields must be packed at the same density <PKHSTc>.
* Data containing blocked points (*DEFS) cannot normally be written to History Save Tapes, since such data are not properly handled on input by the Model, and may not be properly handled when input by the Modular Processor. (There is no check for blocked points when computing derived fields of types 11, 12, and 13). By default, data are checked for blocked points as the tapes are written, and an error abort results if the special value (1.E36) is encountered. The ICP <BPHST> may be used to override this check.
* Processing may continue after the Save tape(s) are written.

->DFLD - Derived fields

This section is divided into four parts: a general description of derived fields (*GDFL), user-defined derived fields (*UDFL), code-defined derived fields (*CDFL), and vector-pair derived fields (*VDFL).

-->GDFL - General description of derived fields

Primary ICPs: [FIELDcn]

Secondary ICPs: <TYPEc> <SPCcn> <DIFFLDn> <RATFLDn> <FLDSRCc> <SPCVP>

A derived field is a specific field which is computed from other specific fields at some particular stage of processing. Derived field computation is very different than a generic transformation (e.g., vertical interpolation),

which applies basically the same transformation to all fields being processed, replacing the original fields. Each derived field is defined with a specific name, which must be different than any of the fields from which it is computed. Source fields are never modified during the computations for the new field.

All derived fields are requested in the same way as fields to be read from the input tapes [FIELDcn]. Although the request to compute a derived field is Case-dependent (*COMP), the definitions of derived fields are Case-independent, and all derived field computations must be based on fields from the same Case (*COMP). If a derived field has the same name as a field on the input tape, the ICP <FLDSRCc> is used to resolve the ambiguity. The fields needed to compute derived fields need not be explicitly requested; they are automatically processed along with the requested fields until they are no longer needed, at which point they are deleted. Once computed, the requested derived fields are processed along with all other fields.

All derived fields are assigned a computation type code, based on when (with respect to other processing steps) they are computed (*ORDR). The relative order of computation for the various computation types can be obtained by placing a decimal point to the left of the type code, then sorting the resulting fractions in ascending order. For example, the following type codes are in computational order: 21,211,212,2121,31,41. The currently available types are listed below in the order in which they are computed. For those computation types described as being computed "after" some processing step, it is not necessary to request that step in order to compute the derived field, with three exceptions: computation types 51, 71, and 81 can be computed only when time filtering, zonal averaging, or meridional averaging, respectively, are requested. This means that different computation types may actually be the same if an intervening operation is not requested. For example, types 31 and 41 are the same if there is no vertical interpolation. Any type may be computed for Cases A and/or B, but only types 61, 62, 71, and 81 can be computed for Case C, since Case C does not exist during the computations for any other types. New computation types can be defined with simple changes to the Processor code. See (*ARCH) for a list of Update Decks.

11 - during input from a CCMOB history tape
12 - during input from a CCMOA history tape
13 - during input from a pressure level history tape
14 - during input from a Time Average or Time Series Save tape
21 - during gridpoint to spectral transformation
211 - in spectral space, inverse Laplacians only
22 - in spectral space (linear)
23 - in spectral space, for spectral graphics only (nonlinear)
24 - during spectral to gridpoint transformation
31 - after spectral operations
41 - after vertical interpolation
51 - after time filtering, inside the time loop
61 - before vertical averaging (after time averaging)
62 - after vertical averaging
71 - after zonal averaging, operating on the averages
81 - after meridional averaging, operating on the averages

Computation types 11 to 14 are special; since each of these types is computed during the reading of input data, and there can be only one input source <TYPEc> for each Case (*DEFS), derived fields can be computed for only one of these four types, for a given Case. For this reason, a derived field can be defined for more than one computation type, if all the types are in the range 11 through 14. Furthermore, the computational algorithm may be different for each type, since it is really the name and computation type pair which defines the derived field in this case. For all other computation types, derived field names must be unique in order to prevent ambiguity when requesting them.

Computation types with a first digit of 2 are also special. Since these computations involve operations in spectral space, it is necessary to request spectral transformation in order to compute any of these derived fields <SPCcn>. Note that this request is Case and field pass dependent - spectral transformation must be explicitly requested for ALL Cases and field passes which request these derived fields.

When derived fields are requested for Case C (Case Comparison differences and ratios (*COMP)), it is not sufficient to request the derived fields with the ICP <FIELDCn>; the fields needed to compute the requested derived field must also be explicitly requested with the <DIFFLDn> and/or <RATFLDn> ICPs, since this is the only way to create Case C fields.

A derived field can be defined in several ways. Code-defined derived fields are defined by the Processor code (*CDFL), while user-defined derived fields (*UDFL) are defined by the user with arithmetic expressions on input data card images. Vector-pair derived fields (*VDFL) are in a sense a mixture: the computational algorithm is fixed by the code, but the field names are specified by the user <SPCVP>. Derived fields are usually code-defined if the computations are complex, since the set of functions available for user-defined derived fields is limited to a predefined set.

All derived fields computed in gridpoint space, however, share the same limitations on data availability: derived fields computed in gridpoint space must be based on longitude-height slices (different latitudes are not available), and derived fields computed in spectral space must be based on longitude-latitude slices. Different Model times are not available for any derived field computations.
Note: The time filtering option (*TFIL) is very general, and can be used to perform simple computations involving different time levels. For example, a time filter with weights -1. and 1. results in a time difference. Such a filtered field can become the basis for a derived field.

Code-defined (*CDFL), user-defined (*UDFL), and vector-pair (*VDFL) derived fields are described in more detail in the following topics.

-->CDFL - Code-defined derived fields

Primary ICPs: [FIELDcn]

Secondary ICPs: <TYPEc> <SPCcn> <DIFFLDn> <RATFLD>

Many common and/or complex derived fields are defined by the Processor code, and may be requested in the same manner as fields to be read from the input tape(s). There are provisions which make it easy to add new code-defined derived fields; with the exception of the spectral computation types (first digit of 2), modifications are simple enough that familiarity with the Processor code is not required. Start by obtaining a listing of subroutine DEFALL in the Module (Deck) of the same name. Comments in this routine describe the procedure.

All code-defined derived fields are alphabetically listed below. Complete definitions follow; the field name may be used as a search string (for GO (*HELP)) to locate the definition.

name	type(s)	description
CHI	211	velocity potential
CONDH	11	condensational heating rate
DELPRES	41	pressure layer thickness
DIV	21	horizontal wind divergence
DKE	61	kinetic energy of differences
DLNPSX	24	x-derivative of natural log of surface pressure
DLNPSY	24	y-derivative of natural log of surface pressure
DVMAG	61	velocity magnitude of differences
ENET	11	net total atmospheric energy flux
ENETS	11,12	net surface energy flux
ENETT	11,12	net top of Model energy flux
HTO	11	geopotential height, full levels, CCMOB version
HT1	11	geopotential height, full levels, CCM1 version
KE	11-14	kinetic energy

KTOOPVO	31	kappa*temperature*omega/pressure, CCMOB version
KTOOPV1	31	kappa*temperature*omega/pressure, CCM1 version
LNPS	11	natural log of surface pressure
MQ	11	water vapor mass (3-dimensional)
NRADS	11,12	net radiative surface flux
PRECT	11	total precipitation (accumulated)
PRES	11,12	pressure on sigma surfaces
PSI	211	stream function
PSL	11,12	sea level pressure
QSRC	11	moisture source term
RADD	11	net radiative heating rate
RELHUM	11-13	relative humidity
TCLD	11,12	total cloud fraction
TEST	11-13	programmable test field (used for debugging)
THETA	11,12	potential temperature on sigma surfaces
TMQ	11	water vapor mass in a column (2-dimensional)
UD	24	u-velocity computed from divergence alone
UVSQ	23	total kinetic energy spectra
UVSQD	23	kinetic energy spectra (divergent part)
UVSQR	23	kinetic energy spectra (rotational part)
UZ	24	u-velocity computed from vorticity alone
VADVTVO	31	vertical advection of temperature, CCMOB version
VADVTV1	31	vertical advection of temperature, CCM1 version
VARWV	61	kinetic energy variance
VD	24	v-velocity computed from divergence alone
VIKE	11	vertical integral of kinetic energy
VIPE	11	vertical integral of potential energy
VITE	11	vertical integral of total energy
VMAG	11-14	horizontal wind speed
VOR	21	horizontal wind vorticity
VZ	24	v-velocity computed from vorticity alone
Z	11,12	geopotential height, half-levels (see Note 1)

Notes:

1) Z should be requested only when working with CCMOA, or if the values are to be compared with previously computed values of Z. In general, HTO or HT1 should be requested instead. The finite differencing used to vertically integrate the hydrostatic equation in the Z computation is not the same as that used in CCMOB. (It matches an earlier, developmental version of the CCM instead.) Also, because Z is defined at half-levels, a potential problem is introduced if Z is written to a Save tape on sigma surfaces, then interpolated to pressure surfaces in a subsequent Processor run. In general, the values for Z in the top Model layer computed this way are different than if Z is computed and interpolated in the same Processor run. The reason is that the highest sigma level of Z which is written to a Save tape is the bottom of the top Model layer (Z is infinite at the next highest half-level). When Z is read from a Save tape and interpolated to pressure surfaces, it is extrapolated above the highest level available, and the information

contained in the temperature of the highest Model layer is not used. This temperature IS taken into account, however, if interpolation is done in the same Processor run as the Z computation. HTO and HT1 avoid this problem because they are defined on full levels.

>CHI
Description: velocity potential
Computation Type(s): 211 (spectral processing required)
Fields Required: DIV (derived)
Units: M**2/S
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space;

$$CHI = - ((A*A) * DIV) / (N * (N+1))$$

where A is the radius of the earth, and
N is the degree of the Legendre polynomial

>CONDH
Description: heating rate due to condensation
Computation Type(s): 11
Fields Required: PS, QC
Units: degrees K per day
Vertical Location: layer midpoints (full levels)
Algorithm: condensational heating rate for layer k is computed as follows:

$$CONDH(k) = 8.64E4 * \frac{G * L}{Cp} * \frac{QC}{PS * (sig(k+1)-sig(k))}$$

where G is the acceleration due to gravity (9.80616 M/S**2),
L is the latent heat of condensation (2.5104E6 J/Kg),
Cp is the specific heat of dry air at constant pressure (1.00464E3 (J/KgK)),
QC is the condensation rate (KG/M**2/S)
PS is the surface pressure (Pascals), and
sig(k) and sig(k+1) are the values of sigma at the upper and lower layer interfaces, respectively, for the layer at which CONDH(k) is defined.

>DELPRES
Description: pressure layer thickness
Computation Type(s): 41
Fields Required: PS
Units: Pascals
Vertical Location: full pressure levels
Algorithm: pressure layer thickness for layer k is computed as follows:

for k=1,NLVLS-1, (k increasing upward, NLVLS>1)

DP(k) = 0.5 * (P(k-1)-P(k+1)) for P(k-1/2) < PS

DP(k) = 0.0 for P(k+1/2) > PS

DP(k) = PS - P(k+1/2) for P(k+1/2) < PS < P(k-1/2)

for k=NLVLS, (NLVLS>1)

DP(NLVLS) = P(NLVLS-1/2) - PTOP for specified PTOP

DP(NLVLS) = AMIN1(P(NLVLS-1/2),P(NLVLS-1)-P(NLVLS))

for PTOP undefined

for NLVLS=1

DP(1) = PS

where P(k) is pressure in Pascals at level k,
DP(k) is DELPRES at a given latitude at level k,
P(k+1/2) = 0.5 * (P(k) + P(k+1)),
P(1/2) > PS for all PS, and
PTOP may be specified via the ICP <PTOPc>

>DIV
Description: horizontal wind divergence
Computation Type(s): 21 (spectral processing required)
Fields Required: U, V
Units: 1/S
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see Model documentation

>DKE
Description: kinetic energy of wind differences
Computation Type(s): 61
Fields Required: DU, DV (both computed as differences)
Units: J/Kg
Vertical Location: layer midpoints (full levels)
Algorithm: DKE = 0.5 * (DU**2 + DV**2)

>DLNPSX
Description: x-derivative (partial) of natural log of surface pressure
Computation Type(s): 24 (spectral processing required)
Fields Required: LNPS (derived)
Units: P/M
Vertical Location: single level
Algorithm: computed in spectral space;

```
    DLNPSX = 1./(A*COS(PHI)*COS(PHI)) * D(LNPS)/D(LAMBDA)

 where A is the radius of the earth
       LAMDA is longitudinal angle
       PHI is latitudinal angle
 and   D( )/D( ) indicates a partial derivative

>DLNPSY
 Description: y-derivative (partial) of natural log of surface pressure
 Computation Type(s): 24 (spectral processing required)
 Fields Required: LNPS (derived)
 Units: P/M
 Vertical Location: single level
 Algorithm: computed in spectral space;

    DLNPSY = (1./A) * D(LNPS)/D(MU)

 where A is the radius of the earth
       MU is the sine of the latitude
 and   D( )/D( ) indicates a partial derivative

>DVMAG
 Description: velocity magnitude of wind differences
 Computation Type(s): 61
 Fields Required: DU, DV (both computed as differences)
 Units: M/S
 Vertical Location: layer midpoints (full levels)
 Algorithm: DVMAG = sqrt( DU**2 + DV**2 )

>ENET
 Description: net (total) energy flux into the atmosphere
 Computation Type(s): 11
 Fields Required: ENETT, ENETS (both derived)
 Units: W/M**2
 Vertical Location: single level
 Algorithm: ENET = ENETT - ENETS

>ENETS
 Description: net energy flux into the surface
 Computation Type(s): 11,12
 Fields Required: FRSA, FRLA, HFL,   QFL           (CCMOB)
                  FRSA, FRLA, LHFLX, SHFLX, TSTAR  (CCMOA)
 Units: W/M**2
 Vertical Location: single level
 Algorithm (CCMOB):

    ENETS = FRSA - FRLA - ( HFL + RH2O*HLAT*QFL ) / MAX(1,DT)

 where RH2O is the density of water (1.E3 Kg/M**3)
       HLAT is latent heat of condensation (2.5104E6 J/Kg)
```

DT is accumulation time (sec) for HFL and QFL

Algorithm (CCMOA):

ENETS = FRSA - FRLA - LHFLX - SHFLX

except over sea ice (TSTAR < 100.), where

ENETS = FRSA - FRLA - LHFLX - SHFLX - 1.046 * (TSTAR + 1.96)

>ENETT
Description: net energy flux into the top of the atmosphere
Computation Type(s): 11,12
Fields Required: SABS, FIRTP
Units: W/M**2
Vertical Location: single level
Algorithm: ENETT = SABS - FIRTP

>HTO
Description: geopotential height, full levels, CCMOB formulation
Computation Type(s): 11
Fields Required: T, PHIS
Units: M
Vertical Location: layer midpoints (full levels)
Algorithm: The geopotential height at level k is given by:

$$HTO(k) = PHIS/GO + R/GO * \sum_{j=1,K} (B(k,j) * T(j))$$

where k is the level index increasing downward from 1 at the highest layer to K at the lowest,
PHIS is the surface geopotential (Pascals),
T(j) is the Model temperature at level j (degrees K),
GO is 9.8 M/S**2,
R is the gas constant for air (287.04 J/(KgK)), and
B(k,j) is the matrix used by CCMOB for integrating the hydrostatic equation (see Model documentation).

>HT1
Description: geopotential height, full levels, CCM1 formulation
Computation Type(s): 11
Fields Required: T, PHIS
Units: M
Vertical Location: layer midpoints (full levels)
Algorithm: The geopotential height at level k is given by:

$$HTO(k) = PHIS/GO + R/GO * \sum_{j=1,K} (B(k,j) * T(j))$$

where k is the level index increasing downward from 1 at

```
      the highest layer to K at the lowest,
   PHIS is the surface geopotential (Pascals),
   T(j) is the Model temperature at level j (degrees K),
   GO is 9.8 M/S**2,
   R is the gas constant for air (287.04 J/(KgK) ), and
   B(k,j) is the matrix used by CCMOB for integrating the
      hydrostatic equation (see Model documentation),
      except for the following four elements:

      B(K  ,K  ) = -ln(sig(K))
      B(K  ,K-1) = 0
      B(K-1,K  ) = -0.5 * ( ln(sig(K)) + ln(sig(K-1)) )
      B(K-1,K-1) =  0.5 * ( ln(sig(K)) - ln(sig(K-1)) )

      where sig(k) is the full-level sigma value for layer k.

>KE
 Description: kinetic energy
 Computation Type(s): 11,12,13,14
 Fields Required: U, V
 Units: J/Kg
 Vertical Location: layer midpoints (full levels)
 Algorithm: KE = 0.5 * ( U**2 + V**2 )

>KTOOPVO
 Description: kappa*temperature*omega/pressure, CCMOB version
 Computation Type(s): 31
 Fields Required: T, U, V, DIV, DLNPSX, DLNPSY
   (DIV, DLNPSX, and DLNPSY are spectral derived fields)
 Units: degrees K per sec
 Vertical Location: full levels (layer midpoints)
 Algorithm: KTOOPVO at level k is given by:

    KTOOPVO(k) = Kappa * T(k) * OMEGA(k) / P(k)

  where k is the level index increasing downward from 1 at
      the highest layer to K at the lowest,

    Kappa = R/Cp = 287.04/1.00464E3, and

    OMEGA(k) / P(k) = U(k)*DLNPSX(k) + V(k)*DLNPSY(k) -

      SUM     ( C(k,j)*(DIV(j)+U(k)*DLNPSX(k)+V(k)*DLNPSY(k)) )
       j=1,k

    and C is the integration matrix used by CCMOB.
    See Eqs. 3.48 - 3.51, p. 27 of "Description of NCAR Community
    Climate Model (CCMOB)", NCAR/TN-210+STR, May 1983.
```

>KTOOPV1
Description: kappa*temperature*omega/pressure, CCM1 version
Computation Type(s): 31
Fields Required: T, U, V, DIV, DLNPSX, DLNPSY
(DIV, DLNPSX, and DLNPSY are spectral derived fields)
Units: degrees K per sec
Vertical Location: full levels (layer midpoints)
Algorithm: KTOOPV1 at level k is given by:

KTOOPV1(k) = Kappa * T(k) * OMEGA(k) / P(k)

where k is the level index increasing downward from 1 at the highest layer to K at the lowest,

Kappa = R/Cp = 287.04/1.00464E3, and

$$\frac{OMEGA(k)}{P(k)} = U(k)*DLNPSX(k) + V(k)*DLNPSY(k) - \sum_{j=1,k} \left(C(k,j)*(DIV(j)+U(k)*DLNPSX(k)+V(k)*DLNPSY(k)) \right)$$

and C is the integration matrix used by CCM1; it is given by:

C(k,j) = B(j,k) * dsig(j) / dsig(k)

where dsig(i) is delta sigma at full level i, and B(j,k) is the CCM1 hydrostatic integration matrix (see Algorithm for HT1). Also see Eqs. 3.48, 3.52, p. 27 of "Description of NCAR Community Climate Model (CCMOB)", NCAR/TN-210+STR, May 1983.

>LNPS
Description: natural log of surface pressure
Computation Type(s): 11
Fields Required: PS
Units: P
Vertical Location: single level
Algorithm: LNPS = ln (PS)

```
>MQ
 Description: water vapor mass (3-dimensional)
 Computation Type(s): 11
 Fields Required: Q, PS
 Units: Kg/M**2
 Vertical Location: Layer midpoints (full levels)
 Algorithm: MQ = PS * Q * DSIG(K) / G

 where  DSIG(K) = SIGMA(K+1/2) - SIGMA(K-1/2)
        K is vertical index, increasing downward
        G is gravitational acceleration (9.80616 M/S**2)

>NRADS
 Description: net radiative flux into the surface
 Computation Type(s): 11,12
 Fields Required: FRSA, FRLA
 Units: W/M**2
 Vertical Location: single level
 Algorithm: NRADS = FRSA - FRLA

>PRECT
 Description: total precipitation (accumulated)
 Computation Type(s): 11
 Fields Required: PRECL, PRECC
 Units: M (accumulated)
 Vertical Location: single level
 Algorithm: PRECT = PRECL + PRECC

>PRES
 Description: pressure on sigma surfaces
 Computation Type(s): 11,12
 Fields Required: PS
 Units: Pascals
 Vertical Location: full sigma levels (layer midpoints)
 Algorithm: PRES(K) = PS * SIGMA(K)

>PSI
 Description: stream function
 Computation Type(s): 211 (spectral processing required)
 Fields Required: VOR (derived)
 Units: M**2/S
 Vertical Location: layer midpoints (full levels)
 Algorithm: computed in spectral space;

        PSI = - ((A*A) * VOR) / (N * (N+1))

 where  A is the radius of the earth, and
        N is the degree of the Legendre polynomial
```

>PSL
Description: sea level pressure
Computation Type(s): 11,12
Fields Required: PHIS, PS, T
Units: Pascals
Vertical Location: single level
Algorithm: Using the two lowest levels of free atmosphere temperature, a lapse rate is defined which is linear in log sigma. This lapse rate is extrapolated from the lowest free atmosphere level to the surface to determine a "surface" temperature as follows:

$$TSURF = T(NLEV) + SF * (T(NLEV-1) - T(NLEV))$$

where T(NLEV) is the temperature at the lowest full level (degrees K) and SF is an extrapolation factor (independent of time) defined as:

$$SF = \frac{\ln(S(NLEV))}{\ln(S(NLEV) / S(NLEV-1))}$$

where S(NLEV) is the sigma value at the lowest full level.

The moist adiabatic lapse rate is then used between the surface and sea level to integrate the hydrostatic equation with the following result:

$$PSL=PS*(1. + GMOIST*PHIS/(TSURF*G))^{G/(R*GMOIST)}$$

where PS is the surface pressure (Pascals),
PHIS is the surface geopotential (M**2/S**2),
R is the gas constant for air (287.04 J/(KgK)),
G is the acceleration due to gravity (9.80616 M/S**2), and
GMOIST is the moist adiabatic lapse rate (6.5E-3 K/M).

>QSRC
Description: moisture source term (accumulated water vapor mass)
Computation Type(s): 11
Fields Required: QFL, PRECL, PRECC
Units: Kg/M**2
Vertical Location: single level
Algorithm:

$$QSRC = (QFL - (PRECL+PRECC)) * RHO$$

where QFL is evaporation (accumulated) (M),
PRECL is the large scale precipitation (accumulated) (M),
PRECC is the convective precipitation (accumulated) (M), and
RHO is the density of water (1.0E3 Kg/M**3)

>RADD
Description: net radiative heating rate
Computation Type(s): 11
Fields Required: QRS, QRL
Units: deg K per day
Vertical Location: layer midpoints (full levels)
Algorithm: RADD = 8.64E4 * (QRS + QRL)

>RELHUM
Description: relative humidity
Computation Type(s): 11,12,13
Fields Required: PS, T, MIXRAT (CCMOA), Q (CCMOB)
Units: per cent; T must be in degrees Kelvin, PS in Pascals
Vertical Location: layer midpoints (full levels)
Algorithm: Relative humidity is computed from the following equation:

$$RELHUM = 100. * MIXRAT * \frac{PS * SIGMA - ES}{0.622 * ES}$$

where MIXRAT is the mixing ratio (Kg/Kg),
PS is the surface pressure (pascals),
SIGMA is the sigma value at the appropriate level, and
ES is the saturation vapor pressure determined from a table lookup routine identical to those in the Models (Pascals).
If the relative humidity is negative, it is set to -1.E-36. This value was chosen because it results in a zero value contour line being drawn around areas of negative RELHUM, and avoids bunching of negative contours around meaningless negative values.

>TCLD
Description: total cloud fraction (all layers combined)
Computation Type(s): 11,12
Fields Required: CLD for CCMOA, CLOUD for CCMOB
Units: none (cloud fraction)
Vertical Location: single level
Algorithm: Using the random overlap assumption, the total clear sky fraction is computed as the product of the clear sky fractions for all levels. (clear sky fraction = 1 - cloud fraction)

>TEST
Description: programmable test field (used for debugging)
Computation Type(s): 11,12,13
Fields Required: none
Units: none
Vertical Location: layer midpoints
Algorithm: A field with a sinusoidal variation in each of the three spatial dimensions is generated, using the grid point indices as independent variables for the sine waves. The mean value, amplitude, period, and phase may all be set independently for each of the three dimensions.

Many different kinds of processing operations can be done analytically for such a field, providing a means of verifing the Processor's point-by-point computations. It is normally necessary to reset these values by modifying the code. See the internal documentation in subroutine CTEST in Deck COMPDF for further details. Other analytical functions can be used instead of the sine waves, including dependencies on time and other history tape fields.

>THETA
Description: potential temperature on sigma surfaces
Computation Type(s): 11,12
Fields Required: T, PRES (derived)
Units: degrees Kelvin
Vertical Location: full sigma levels (layer midpoints)
Algorithm:

THETA(k) = T(k) * (1.E5/PRES(k)) ** (R/CP)

where R = 287.04 J/(KgK)
and CP = 1.00464E3 J/(KgK)

>TMQ
Description: water vapor mass in a column (2-dimensional)
Computation Type(s): 11
Fields Required: MQ (derived)
Units: Kg/M**2
Vertical Location: single level
Algorithm: TMQ = sum of all levels of MQ

>UD
Description: U-velocity computed from divergence alone
Computation Type(s): 24 (spectral processing required)
Fields Required: VOR, DIV (derived)
Units: M/S
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see Model documentation

>UVSQ
Description: Total kinetic energy spectra
Computation Type(s): 23 (spectral processing required)
Fields Required: VOR, DIV (derived)
Units: J/Kg
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space;

$$UVSQ = 1/4 * N*(N+1)/(A*A) * (\overline{PSI} * PSI + \overline{CHI} * CHI)$$

where PSI = - ((A*A) * VOR) / (N * (N+1))
CHI = - ((A*A) * DIV) / (N * (N+1))
A is the radius of the earth
N is the degree of the Legendre polynomial
and the overbar indicates the complex conjugate.

>UVSQD
Description: Kinetic energy spectra (divergent part)
Computation Type(s): 23 (spectral processing required)
Fields Required: VOR, DIV (derived)
Units: J/Kg
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space;

$$UVSQD = 1/4 * N*(N+1)/(A*A) * (CHI * \overline{CHI})$$

where CHI = - ((A*A) * DIV) / (N * (N+1))
A is the radius of the earth
N is the degree of the Legendre polynomial
and the overbar indicates the complex conjugate.

>UVSQR
Description: Kinetic energy spectra (rotational part)
Computation Type(s): 23 (spectral processing required)
Fields Required: VOR, DIV (derived)
Units: J/Kg
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space;

$$UVSQR = 1/4 * N*(N+1)/(A*A) * (PSI * \overline{PSI})$$

where PSI = - ((A*A) * VOR) / (N * (N+1)),
A is the radius of the earth
N is the degree of the Legendre polynomial
and the overbar indicates the complex conjugate.

>UZ
Description: U-velocity computed from vorticity alone
Computation Type(s): 24 (spectral processing required)
Fields Required: VOR, DIV (derived)
Units: M/S
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see Model documentation

>VADVTV0
Description: vertical advection of temperature, CCMOB version
Computation Type(s): 31
Fields Required: T, U, V, DIV, DLNPSX, DLNPSY
(DIV, DLNPSX, and DLNPSY are spectral derived fields)
Units: degrees K per sec
Vertical Location: layer midpoints (full levels)
Algorithm: VADVTV0 at level k is given by:

for $1 \le k \le K-1$,

$$VADVTV0(k) = sigdot(k) * \left(a_{k+} * (dT/ds)_{k+1/2} + a_{k-} * (dT/ds)_{k-1/2} \right)$$

and

$$VADVTV0(K) = sigdot(K) * \frac{T(K) - T(K-1)}{sig(K) * \ln(sig(K)/sig(K-1))}$$

where k is the level index increasing downward from 1 at the highest layer to K at the lowest,
sig(k) is sigma at full level k,
sigdot(k) is sigma dot at full level k,
(dT/ds) is the partial derivative of temperature with respect to sigma, and
a_{k+} and a_{k-} are layer weights based on sigma.

See Eqs. 3.19 - 3.26, 3.32, pp. 22-24 of "Description of NCAR Community Climate Model (CCMOB)", NCAR/TN-210+STR, May 1983.

>VADVTV1
Description: vertical advection of temperature, CCM1 version
Computation Type(s): 31
Fields Required: T, U, V, DIV, DLNPSX, DLNPSY
(DIV, DLNPSX, and DLNPSY are spectral derived fields)
Units: degrees K per sec
Vertical Location: layer midpoints (full levels)
Algorithm: VADVTV1 at level k is given by:

$$VADVTV1(k) = \frac{sig(k+1)-sig(k)}{2 * delsig(k)} * sigdot(k+1/2) * (dT/ds)_{k+1/2} + \frac{sig(k)-sig(k-1)}{2 * delsig(k)} * sigdot(k-1/2) * (dT/ds)_{k-1/2}$$

where k is the level index increasing downward from 1 at the highest layer to K at the lowest,

sig(k) is sigma at full level k,
sigdot(k+1/2) is sigma dot at half level k+1/2,
sigdot(1/2) = sigdot(K+1/2) = 0,
delsig(k) is delta sigma at full level k, and

$$\left(\frac{dT}{ds}\right)_{k+1/2} = \frac{T(k+1)-T(k)}{sig(k+1)-sig(k)}$$

delsig(k) is computed from Eq. 3.29, p. 24 of "Description of NCAR Community Climate Model (CCMOB)", NCAR/TN-210+STR, May 1983. sigdot(k+1/2) is computed from Eq. 3.27, p. 23 of the same document, except that the last term (with the 1/4 factor) is omitted.

>VARWV
Description: variance of wind vector
Computation Type(s): 61

Fields Required: $\overline{U}$, $\overline{V}$, $\overline{KE}$ (KE derived, all must be time averaged)
Units: J/Kg
Vertical Location: same as U, V

Algorithm: $VARWV = 2.0 * \overline{KE} - \overline{U}*\overline{U} - \overline{V}*\overline{V}$

>VD
Description: V-velocity computed from divergence alone
Computation Type(s): 24 (spectral processing required)
Fields Required: VOR, DIV (derived)
Units: M/S
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see Model documentation

>VIKE
Description: vertical integral of kinetic energy
Computation Type(s): 11
Fields Required: U, V, PS
Units: J/M**2
Vertical Location: single level
Algorithm: VIKE = sum of all levels of KE(K)

where KE(K) = 0.5 * (U*U + V*V) * DSIG(K) / G
and
DSIG(K) = SIGMA(K+1/2) - SIGMA(K-1/2)
K is vertical index, increasing downward
G is gravitational acceleration (9.80616 M/S**2)

>VIPE
Description: vertical integral of potential energy
Computation Type(s): 11
Fields Required: T, PHIS, PS

```
 Units: J/M**2
 Vertical Location: single level
 Algorithm: VIPE = sum of all levels of PE(k)

 where  PE(k) = ( CP*T + PHIS ) * DSIG(k) / G
 and
        DSIG(k) = SIGMA(k+1/2) - SIGMA(k-1/2)
        k is vertical index, increasing downward
        CP is specific heat of dry air at constant pressure
           (1.00464E3 J/Kg/deg K)
        G is gravitational acceleration (9.80616 M/S**2)

>VITE
 Description: vertical integral of total energy
 Computation Type(s): 11
 Fields Required: VIKE, VIPE (both derived)
 Units: J/M**2
 Vertical Location: single level
 Algorithm: VITE = VIKE + VIPE

>VMAG
 Description: velocity magnitude
 Computation Type(s): 11-14
 Fields Required: U, V
 Units: M/S
 Vertical Location: layer midpoints (full levels)
 Algorithm: VMAG = sqrt( U**2 + V**2 )

>VOR
 Description: horizontal wind vorticity
 Computation Type(s): 21
 Fields Required: U, V
 Units: 1/S
 Vertical Location: layer midpoints (full levels)
 Algorithm: computed in spectral space; see Model documentation

>VZ
 Description: V-velocity computed from vorticity alone
 Computation Type(s): 24
 Fields Required: VOR, DIV (derived)
 Units: M/S
 Vertical Location: layer midpoints (full levels)
 Algorithm: computed in spectral space; see Model documentation

>Z
 Description: Geopotential height (also see HT0 and HT1)
 Computation Type(s): 11,12
 Fields Required: PHIS, T, Q (CCMOB only)
 Units: M
```

Vertical Location: layer interfaces (half-levels). An extra half-level may be added above the highest full level defined by the Model in order to incorporate the information contained in the temperature at the highest Model level, which would otherwise be lost. (This information may be needed later for vertical interpolation.) The extra half-level is placed by assuming a geometric progression of sigma values, but the location is actually immaterial for the purpose of vertical interpolation because of the extrapolation algorithm at the top of the Model. This extra level is added only if vertical interpolation is to be performed.

Algorithm: The surface geopotential height is computed as follows:

$$Z(NLEV) = PHIS/G$$

where PHIS is the surface geopotential (Pascals), and
G is the acceleration due to gravity (9.80616 M/S**2),

The rest of the levels are computed by working upward, from one half-level to the next, using the equation:

$$Z(k) = Z(k+1) + R*B(k)*TV(k)/G$$

where R is the gas constant for air (287.04 J/(KgK)),
B(k) = ln(SIGMA(k) / SIGMA(k+1)),
SIGMA(k) is the Model sigma value at level k, and
TV(k) is the virtual temperature given by:

$$TV(k) = T(k) * \frac{0.622 + Q}{0.622*(1.+Q)} \qquad \text{(for CCMOB)}$$

$$TV(k) = T(k) \qquad \text{(for CCMOA)}$$

where T(k) is the Model temperature at the midpoint of layer k,
Q is the mixing ratio (Kg/Kg), and
k is the vertical index, increasing downward.

-->UDFL - User-defined derived fields

Primary ICPs: [DERFLD] [FIELDcn]

Secondary ICPs: None

It is possible to define derived fields by specifying the computational algorithm as an arithmetic expression on an ICP data card image [DERFLD]. The computations are performed point-by-point on longitude-height vertical slices of fields in gridpoint space. Once defined, these derived fields may be requested in the same way as code-defined derived fields [FIELDcn].

Defining expressions are specified in Reverse Polish Notation (RPN). Each item in the expression is either an operand (field or scalar constant) or an operator (function). Functions are either unary (one operand) or binary (two operands), and must be selected from a given list [DERFLD] (the list can be expanded with minor code modifications). Expression evaluation is performed according to standard RPN rules using a four-register, vector stack; this is similar to the method used by many hand calculators, except that most operands are vectors (longitude-height slices). Evaluation proceeds as follows. The expression is scanned from left to right, looking at one item at a time. If an operand is encountered, it is copied to the top register on the stack (or the logical equivalent for scalar operands). If a unary function is encountered, the operation is performed directly on the top register. If a binary function is encountered, the operation is performed using the top two registers, overwriting operand 1 (second from the top of the stack) with the result; operand 2 is popped (deleted) from the stack. This procedure continues until the .END operator is encountered; this causes the top register in the stack to be moved to the location set aside for the derived field. This will leave the stack empty if the number of operands in the expression is exactly one more than the number of binary functions; otherwise the expression is considered invalid.

Although user-defined derived fields cannot be computed in spectral space, the vector-pair derived field capability (*VDFL) allows some degree of control in defining new derived fields to be computed in spectral space.

-->VDFL - Vector-pair derived fields

Primary ICPs: [SPCVP] [FIELDcn] [SPCcn]

Secondary ICPs: None

Most of the derived fields (*CDFL) computed in spectral space (*SPEC) are based on the orthogonal vector fields U and V, which require special handling as a pair. The eight fields which may be derived from the (U,V) pair are: DIV, VOR, CHI, PSI, UD, UZ, VD, and VZ. The algorithms used to compute these fields may also be applied to an arbitrary vector pair, thus defining an analogous set of eight new derived fields. These new derived fields are defined by the ICP [SPCVP], which is used to specify one or more additional vector pairs, plus the names of the associated set of derived fields for each pair. Each specified set of ten field names completely defines a set of eight vector-pair derived fields. The vector-pair basis fields are not limited to input fields; either or both may be previously-computed derived fields. For example, the divergence of moisture flux may be computed by defining U*Q and V*Q as user-defined derived fields (*UDFL), and then defining U*Q and V*Q as a vector pair.

The ICP [SPCVP] just defines vector-pair derived fields; these fields are computed only if they are requested with the [FIELDcn] ICP. Also, since all vector-pair derived fields require spectral processing, they can only be computed if [SPCcn] is set to 'YES'.

->VERT - Vertical interpolation

Interpolation to pressure surfaces (*VPRS) is the only form of vertical interpolation currently supported.

-->VPRS - Vertical interpolation to pressure surfaces

Primary ICPs: [PRESSLE] [FIELDcn]

Secondary ICPs: <INTDP> <LBTDP> <NLCDP> <PINTXL>

The fields being processed may be vertically interpolated from the Model's sigma (P/PO) vertical coordinate to an arbitrary set of pressure levels [PRESSLE] (same levels for all fields). When interpolating to pressure surfaces, surface pressure must be explictly requested for processing [FIELDcn] if (and only if) the processed data are NOT read from a history tape containing PS (*IHST).

There are three adjustable interpolation parameters: 1) the interpolation curve type <INTDP>, 2) the treatment of the lower Model boundary <LBTDP>, and 3) the number of Planetary Boundary Layer levels to leave on sigma surfaces <NLCDP>. A single value for each of these parameters may be specified as a default for any field not named in an exception list <PINTXL>, and each parameter may also be specified independently for each field in the exception list. A table summarizing the parameter settings is printed in the output file when vertical interpolation is performed. The parameters are described in detail below.

Vertical interpolation is implemented by first applying a transform function to the input vertical coordinate, and then interpolating linearly with respect to the transformed coordinate. For any individual field, any one of three different transform functions may be specified, or interpolation can be skipped for that field. Thus there are four different interpolation types available:

0 - no interpolation
1 - linear in sigma
2 - linear in natural log of sigma
3 - linear in natural log of natural log of (1000*sigma + 2.72)

The significance of the factor of 1000 in the fourth type is simply that it makes the transformation similar to the natural log of the natural log of pressure. The sigma form, however, results in more efficient processing because it does not vary with time on the input surfaces. The 2.72 additive factor prevents an attempt to compute the log of a negative number in cases where sigma becomes very small; it has a negligible effect for large sigma. The ICPs <INTDP> and <PINTXL> control the setting of the interpolation type from a fixed set of predefined interpolation types. Additional interpolation types may be defined by a simple code modification, as long as

the transformation of the vertical coordinate for the input surfaces is independent of time; otherwise the code modification is considerably more complex. The internal documentation in Module SIG2P contains instructions for adding new interpolation types.

Depending on the pressure levels requested (and the surface pressure output by the Model), there may not be data available both above and below a given interpolation point. When this happens above the top of the Model, a vertical extrapolation is always performed. When it happens at the Model's lower boundary, either an extrapolation is performed, or the point is "blocked" (*DEFS). No attempt is made to perform a horizontal interpolation. Whether lower boundary points are blocked or extrapolated is determined by the values assigned to the ICPs <LBTDP> and <PINTXL>. The same parameter also determines (independently) whether the "surface" level, if available, is to be used in the interpolation (or extrapolation). The "ground" level is never used (*XBTL).

Another parameter, controlled by the ICPs <NLCFP> and <PINTXL>, specifies the number of free atmospheric levels (counting from the bottom), which are to remain on sigma surfaces. These levels are placed below the requested pressure levels, mixing sigma and pressure surfaces together. The purpose of this feature is to handle very low "free atmosphere" levels which are within the Planetary Boundary Layer, and are best interpreted on sigma rather than pressure surfaces. Since mixing sigma and pressure coordinates on a vertical cross-section plot introduces a contour discontinuity, this option is most useful when only horizontal projection plots are being produced.

Each field is independently assigned a value for each of these three interpolation parameters. If the field is not named in the exceptions list <PINTXL>, then the values of <INTDP>, <LBTDP>, and <NLCDP> are used (each of these ICPs has a default value). If the field is in the exceptions list, then all three values are specified by that list. A default exceptions list is used if, and only if, all three ICPs <INTDP>, <LBTDP>, and <NLCDP> are defaulted. If a value is input for any of these ICPs, the exceptions list is considered to be null (i.e., it has no entries) unless it is explicitly specified with the <PINTXL> keyword.

->SPEC - Spectral operations

Primary ICPs: [SPCcn]

Secondary ICPs: <SPCINTc> <SPBPcn> <SPCDFcn> <SPCEFcn> <TIMAVGc>

The data being processed may be transformed into Spherical Harmonic Coefficients (SHC) [SPCcn], and in this form, may be interpolated (*SINT) and/or bandpass filtered (*SBND). In addition, derived fields may be computed in spectral space (*DFLD), and the SHC for all fields may be graphed (*SPGR). By default, all fields are then transformed back into grid point space for further processing. Fields which are not needed for further

processing (or those which do not properly transform to grid point space) may (and should) be deleted before this transformation <SPCDFcn>. Note that it is possible to transform into spectral space and back to grid point space without performing any operations on the SHC. For some fields output by the Model, this processing step has no effect. For many fields, however, the two transforms will have the effect of filtering the data. If this filtering is undesirable, or if some fields are simply not needed in spectral space, then those fields may be excluded from the spectral transformations <SPCEFcn>. When spectral processing is requested [SPCcn], all other fields being processed, except for horizontal wind (U and V), are transformed into SHC; The transformation algorithm is the same as in the Model (CCMOB), and is defined by the same pentagonal spectral truncation parameters (M, N, and K).

Note: The Model is coded for a pentagonal truncation of the series of spherical harmonic functions used to represent prognostic variables. The parameter M is the highest Fourier wavenumber included in the east-west representation; K is the highest degree of the Legendre polynomials used for the north-south representation, and N is the highest degree of these polynomials at a Fourier wavenumber of 0. For additional details, see pp. 28 and 29 of the NCAR Tech Note "Description of NCAR Community Climate Model (CCMOB)".

Since U and V are not properly defined by the spectral transformation, they are replaced in spectral space by vorticity and divergence. When requesting spectral operations, U and V should only be requested as a pair; if only one or the other is requested, an error termination results.

The spectral coefficients may also be used as the basis for computing derived fields (*DFLD), including some non-linear fields. All fields may also be time averaged in spectral space (*TAVG) and/or graphed (*SPGR).

-->SINT - Spectral interpolation

Primary ICPs: [SPCINTc] [SPCcn] [FIELDcn]

Secondary ICPs: none

If spectral operations have been requested (*SPEC) [SPCcn], then the spectral resolution of the data can be altered by specifying new values for the spectral truncation parameters M, N, and K [SPCINTc]. These new values may be either larger (interpolation) or smaller (smoothing) than the old ones. The resolution change is accomplished by either adding additional SHC (set to zero), or by truncating existing SHC. For proper transformation back into grid point space, consistent values for the horizontal grid resolution must also be specified [SPCINTc].

-->SBND - Spectral bandpass filtering

Primary ICPs: [SPCcn] [SPBPcn] [FIELDcn]

Secondary ICPs: none

If spectral operations have been requested (*SPEC) [SPCcn], then the data may be bandpass filtered in spectral space [SPBPcn]. This is accomplished by setting the spectral truncation parameters m and n to zero except within a specified range:

$$m=0 \text{ for } m < M_1 \text{ or } m > M_2$$
$$n=0 \text{ for } n < N_1 \text{ or } n > N_2$$

where m is the Fourier wavenumber, n is the order of the Legendre polynomial, and M_1, M_2, N_1, and N_2 are input parameters [SPBPcn].

The bandpassing rectangle defined by these input parameters must intersect some portion of the spectral pentagon defined by the current values of M, N, and K or a fatal error will result. If necessary, the requested spectral band is modified so that it is wholly contained in the current K by M rectangle. See pp. 28 and 29 of the NCAR Tech Note "Description of NCAR Community Climate Model (CCMOB)".

->TAVG - Time average statistics

Primary ICPs: [TIMAVGc] [TAPESc] [DAYSc] [FIELDcn]

Secondary ICPs: <SDFLDcn> <CVFLDcn> <TCFLDcn> <PRFLDcn>
<ZSTFLcn> <ZCVFLcn> <DERFLD>

All fields being processed [FIELDcn] may be averaged over the days being processed [DAYSc] in gridpoint space. Either or both comparison Cases (A and/or B) may be time averaged [TIMAVGc] for all field passes. If (and only if) time averages are requested, then a number of other time average statistics may also be requested <SDFLDcn> <CVFLDcn> <TCFLDcn> <PRFLDcn> <ZSTFLcn> <ZCVFLcn>. These statistics (which are defined below) are computed for all levels of the specified fields. When computed, the zonal eddy statistics [ZSTFLcn] [ZCVFLcn] (*ZEST) are automatically plotted as meridional cross sections (*PMEX). All fields used to compute all additional statistics must be explicitly requested for processing <FIELDcn> in the appropriate Case and field pass. All time average statistics are available for further processing; names for these fields are specified by the user.

Note that combining time average statistics with Save tapes (*SAVT), user-defined derived fields (*UDFL), time filtering (*TFIL), and possibly Case comparison statistics (*COMP) in multistep Processor jobs allows the

computation of many additional complex statistics. When Time Average Save tapes are input (*STAV), the input tapes [TAPESc] define the time sample to be averaged, and the days list [DAYSc] is ignored. A time series consisting of only one time sample may be averaged.

The time average statistics which may be requested are listed below. In the defining expressions, "x" and "y" denote arbitrary fields, and computations are performed at matching points in three-dimensional space. Overbars and square brackets denote time and zonal averages, respectively. Primes and stars denote deviations from time and zonal averages, respectively.

$\overline{x}$: time average [TIMAVGcn]

$\mathrm{sqrt}(\overline{x'x'})$: time standard deviation <SDFLDcn>

$\overline{x'y'}$: time covariance <CVFLDcn>

$\overline{xy}$: time average of product <PRFLDcn>

$\overline{x*y*}$: time average of product of deviations from zonal mean <TCFLDcn>

$\overline{\mathrm{sqrt}[x*x*]}$: time average of zonal standard deviation <ZSTFLcn>

$\overline{[x*y*]}$: time average of zonal covariance <ZCVFLcn>

All correlation statistics are computed for corresponding points in space, between fields in the same Case (*COMP). It is possible, however, to extract any level of any field and, treating it as a single-level field, correlate it with any other single-level field. See (*UDFL) and the example for <DERFLD>.

All standard deviations and covariances are computed by dividing the time series sums by n, not n-1 (where n is the length of the time series). This means that these statistics describe only the time series being processed, and are NOT the best estimate of these statistics for a longer time series based on the analysis of a sample. The distinction is usually negligible except for a very short sample.

When time average statistics are computed, a complete list of the days in the analyzed series is stored with the data. For time average operations which have been applied more than once by making multiple runs (*SAVT), this

list is formed by using only the first day of the list for each time average being averaged.

There is also an option to time average all fields in spectral space for the purpose of graphing them [TIMAVGc] (*SPGR).

->COMP - Case comparison

Primary ICPs: [DIFFLDn] [TAPESc] [DAYSc] [FIELDcn] [TYPEc]

Secondary ICPs: <RATFLD> <ORIGFLD> <TITLEc> <NSDPRNT>

Two different sets of data (Cases A and B) can be requested for processing in the same run, and these sets may be compared within that run. The Cases to be compared may be input in different forms (e.g., history tapes can be compared to time average Save tapes) and many processing options may be controlled individually for each Case. Both Cases must have the same spatial resolution and vertical coordinate when they are compared (*ORDR), but resolutions can be changed during the run for either or both Cases (*SINT) (*VERT).

The Cases are quantitatively compared by computing differences and/or ratios for specific fields at corresponding points in space. The fields to be compared should be defined at the same levels in the vertical, with the possible exception of extra bottom levels (*XBTL); they are compared if (and only if) they exist for both fields. The fields to be compared must be requested explicitly within the same field pass [FIELDcn]. Field names for the differences and/or ratios [DIFFLDn] <RATFLDn> are specified independently for each Case. The resultant fields are always available for further processing as Case C, but continued processing of the original fields is optional <ORIGFLD>.

The time(s) compared [DAYSc] need not be the same for both Cases. If time averaging of the input data is requested for one Case and not the other, then the time average for that Case is compared to the first day of the other Case. If both Cases are time averaged, then the averages are compared. If neither Case is time averaged, then the first day in Case B is compared to the first day in Case A, the second day in Case B is compared to the second day in Case A, and so on; the total number of days compared is equal to the minimum of the number of days specified for either Case. Since the two Cases are processed independently prior to the comparison, they may share the same input data in whole or part. This means that a Case can be compared with itself, allowing the computation of differences and/or ratios of different fields at the same time, or the same field at different times.

When Case differences are requested [DIFFLDn], a printout of "significant" differences may also be requested <NSDPRNT>. A computed difference is "significant" if the Case A value differs from the Case B value by more than a specified number of decimal digits. All fields for which differences are requested are compared in this way, at all levels,

longitudes, and latitudes, in that order. For each significant difference (up to the first 10 per latitude or 50 per field), the location (latitude, longitude, and level), and the Case A and Case B values are printed. A summary of all significant differences is also printed, including the point of maximum (absolute value) difference, and the minimum, maximum, and average (absolute values) for Case A, Case B, and the difference.

Many of the Case-dependent keywords are defined for Case C in addition to Cases A and B, so it is possible to control processing of Case C individually. However, since Case C exists only after Case Comparison, there are some option limitations (*ORDR).

->MASK - Surface type masking

Primary ICPs: [MASKSc]

Secondary ICPs: <ZBKFR> <MBKFR> <VBKFR> <ZSBKFR>
<TYPEc> <SFCTTAP> <SFCTCRT>

The fields being processed may be "masked" by surface type. Any combination of the three surface types (land, ocean, sea ice) may be specified. When masking is requested, all levels of all fields are masked by setting the field values at masked points to the "blocked point" value 1.E36 (*DEFS), which excludes these points from subsequent computations. When performing spatial averages on masked fields, it may be desirable to explicitly set the blocking fraction values <ZBKFR> <MBKFR> <VBKFR> <ZSBKFR>.

This masking option is independent of other options which may perform a similar masking as part of another operation.

The distribution of surface types must be available whenever masking is requested. It is assumed to be independent of time. (But see (*UDFL) and the [DERFLD] example for a method of time-dependent masking.) If a history tape is used as input for Case A <TYPEc>, then the surface type information is automatically extracted from it and used for the masking. Otherwise, it is necessary to specify a surface type Save tape (*SFCT) <SFCTTAP> with the same horizontal resolution as the data being masked. A fatal error will result if the resolutions do not match.

There are two Surface Type Save Tapes archived on the TBM for use with CCMOA or CCMOB R15 data. They are SR15JA and SR15JL, which contain sea ice distributions for January and July, respectively. A surface type Save tape can be created in any run in which a history tape is input <SFCTCRT>.

->TFIL - Time filtering

Primary ICPs: [TIMFILc] [DAYSc] [FIELDcn]

Secondary ICPs: <TFWTSc>

The fields being processed may be filtered in time by applying a centered, weighted, "running mean" type of filter [TIMFILc]. All fields being processed [FIELDcn] are filtered. The filter is applied to the ith time of the requested series [DAYSc] as follows:

$$FF_i = \sum_{j=1,n} W_j * UF_{i-(n/2)+j-1}$$

where

FF is the filtered field at a given point in space,
UF is the unfiltered field at the same point in space,
n is the number of weights (width of the filter),
W_j is the jth filter weight, and
(n/2) is truncated to an integer.

Note that with the proper choice of weights, this "filter" can also be used to compute time differences and derivatives.

In order to accommodate the full width of the filter at each filtered time, this equation is applied only to those times sufficiently far from the end points of the time series. Therefore the resultant filtered time series is n-1 time samples shorter than the unfiltered series. Note that if the total number of weights is even, the filter's center falls between existing times; in this case the center is shifted one-half of a time increment towards a later time. Any one of the following sets of weights may be applied:

* 31-point low-pass filter (passes frequencies 0.0 to 0.1 inverse days for 0.5 day increment time series)
* 31-point band-pass filter (passes frequencies 0.17 to 0.45 inverse days for 0.5 day increment time series)
* 31-point high-pass filter (passes frequencies 0.5 to 1.0 inverse days for 0.5 day increment time series)
* any filter of arbitrary width (up to about 75 points) with an arbitrary set of weights. The weights may be specified in several different ways <TIMFILc> <TFWTSc>.

A list of the weights used is printed in the Processor output file. For detailed descriptions of the 3 predefined 31-point filters, see Blackmon, M.L., "A Climatological Spectral Study of the 500 mb Geopotential Height of the Northern Hemisphere", J. Atmos. Sci., Vol. 33, No. 8, August, 1976,

p.1609. Note that if all weights are equal and normalized <TIMFILc> <TFWTSc>, then running means are computed.

Although more than one Case can be time filtered in the same job step, a separate step is required to compare the filtered cases on a day by day basis. It is possible, however, to filter two Cases, compute time average statistics, and compare the time average statistics, all in one job step (*ORDR).

Although time filters with a width of up to about 75 time samples may be specified, the restrictions on the amount of data that may be filtered at once become very severe for filters approaching this limit. Applying a 31-point filter to output from a 9-level, R15 Model will allow about 10 field-levels to be filtered at once. This number decreases in proportion to an increase in filter width. The size restrictions apply to Cases individually, i.e., Case A is filtered completely before filtering of Case B begins. See (*MLIM) for a discussion of memory limitations.

Saving of the filtered data on Time Series Save Tapes (*STSR) is recommended for most time filtering applications.

->ZEST - Zonal eddy statistics

Primary ICPs: [TIMAVGc] [ZSTFLcn] [ZCVFLcn] [FIELDcn]

Secondary ICPs: <ZSBKFR>

Zonal eddy statistics can be computed only when time average statistics have been explicitly requested [TIMAVGc]. Furthermore, the requested fields [FIELDcn] should be limited to those involved in the zonal eddy statistic computations. When requested, these statistics [ZSTFLcn] [ZCVFLcn] are computed for all levels of the specified field(s), and are automatically plotted as meridional cross sections (*PMEX). See (*TAVG) for definitions of the statistics which may be requested. The zonal eddy statistics will be undefined (blocked) at any latitude where the number of blocked points exceeds a certain fraction <ZSBKFR>.

Implementation Note: Partial computations involving the longitudinal variations are implemented as an independent code Module; the zonally averaged data are then time averaged. This means that some aspects of the zonal eddy statistics computations can be changed by simply rearranging Module calls.

Also see (*UDFL) and <DERFLD> for another, more general way to compute statistics based on deviations from zonal averages.

->SPAV - Spatial averaging

Four different kinds of spatial averaging may be requested: 1) an arbitrary band of longitudes may be averaged along all latitude lines (*ZONA), 2) an arbitrary band of latitudes may be averaged along all meridians (*MERA), 3) an arbitrary subset of contiguous vertical levels may be averaged over the full horizontal domain (*VERA), and 4) all points above a specified surface type or types (land, ocean, sea ice) may be horizontally averaged, level by level, within an arbitrary latitude-longitude rectangle (*HORA).

All horizontal spatial averaging operations are done in parallel with each other. In other words, the results of one averaging operation are NOT used as the starting point for another averaging operation in the same jobstep; each average is performed independently starting with the same data (which may have been modified by other options). Vertical averaging, however, is performed in series with other processing operations, i.e., vertical averaging is performed, then processing (including other spatial averages) continues, operating on the vertical averages (*ORDR). All spatial averaging options are Case independent, i.e., they apply to all Cases being processed.

-->VERA - Vertical averaging

Primary ICPs: [VERAVG]

Secondary ICPs: <VAVRNG> <VBKFR> <FIELDcn> <HPROJ>

When vertical averaging is requested [VERAVG], all fields are averaged at each point in the horizontal domain. Processing then continues, with subsequent operations using the vertical averages as a starting point (*SPAV) (*ORDR). An arbitrary subset of contiguous vertical levels may be averaged <VAVRNG>, and the minimum fraction of unblocked points used to compute each average may be controlled <VBKFR>. There is a special provision for handling those plot characteristics which can be assigned individually to specific levels (e.g., contour intervals <HPCINT>). All vertical averages are assigned a level value of -1. Since this is a unique level value, plot characteristics for vertical averages can be assigned independently of the characteristics for individual levels, using the same ICPs. See the <HPCINT> keyword description for details of how this is done. The main purpose of this feature is to avoid having to change plot characteristic ICPs when switching vertical averaging on and off.

Each level used in the average is weighted by a layer thickness which depends on the location of the level in the vertical. For levels that are not adjacent to either the bottom or top boundaries, this thickness is computed as

$$d_k = ABS(h_{k+1} - h_{k-1}) / 2.$$

where d is the layer thickness,
h is the value of the vertical coordinate, and
k is a vertical index increasing upward.

When k is the highest level available, then

$$d_k = ABS(h_k - h_{k-1})$$

and when k is the lowest level available, then

$$d_k = (ABS(h_k - h_{k+1}) + ABS (h_k - h_s)) / 2.$$

where s refers to the surface (ground elevation). In this last case the level k is effectively assigned a layer which ranges from the surface to a point midway between levels k and k+1, allowing partial layers immediately above the surface to be included in the average. Field values at the surface are never included. The average is normalized by the sum of the weights.

If the vertical coordinate is pressure, then the surface pressure is required for averaging levels next to the surface, so surface pressure must be explicitly requested for processing when vertical averaging on pressure surfaces is requested.

Also see (*UDFL) and <DERFLD> for another way to compute vertical averages using user-defined derived fields.

-->ZONA - Zonal averaging

Primary ICPs: [ZONAVG]

Secondary ICPs: <ZAVRNG> <ZBKFR> <ZAVGPRN>

When zonal averaging is requested [ZONAVG], all levels of all fields are averaged along each latitude line. All longitudes are weighted equally. An arbitrary longitude range for the averaging may be specified <ZAVRNG>, and the minimum fraction of unblocked points used to compute each average may be controlled <ZBKFR>. The averages may also be printed in the output file <ZAVGPRN>. When computed, zonal averages are always plotted as meridional cross sections (*PMEX). Plot characteristics can be controlled through the use of the appropriate ICPs (*PLOT).

-->MERA - Meridional averaging

Primary ICPs: [MERAVG]

Secondary ICPs: <MAVRNG> <MBKFR> <MAVGPRN>

When meridional averaging is requested [MERAVG], all levels of all fields are averaged along all meridians. Gaussian weights are applied at each latitude to account for the variable area represented by each point contributing to the average. An arbitrary latitude range for the averaging may be specified <MAVRNG>, and the minimum fraction of unblocked points used to compute each average may be controlled <MBKFR>. The averages may also be printed in the output file <MAVGPRN>. When computed, meridional averages are always plotted as latitudinal cross sections (*PLXC). Plot characteristics can be controlled through the use of the appropriate ICPs (*PLOT).

-->HORA - Horizontal area averaging with surface type masking

Primary ICPs: [MSKFLcn]

Secondary ICPs: <SFCTTAP> <SFCTCRT> <TYPEc> <MSKAP> <MSKAPS> <MSKAZS>

If masked area averages are requested [MSKFLcn], then each level of each specified field is averaged over a specified latitude-longitude rectangle. A surface type mask may also be applied to the rectangle in order to limit the averaging to points above a particular surface type or types. Any combination of land, ocean, and sea ice masks may be specified; combining all three results in no masking, i.e., no exclusions due to surface type. Different masks and rectangles may be specified for each field to be averaged, and each field may be averaged more than once. All levels of each field are averaged individually, using the same mask. Gaussian weights are applied at each latitude to account for the variable area represented by each point contributing to the average. Masked area averages requested with the [MSKFLcn] keyword are always printed in the output file, but they cannot be plotted, except as time series (*PTAL).

The three ICPs <MSKAP>, <MSKAPS>, and <MSKAZS> may be used to modify the averaging process. The three options are: 1) averaging of point (field) values <MSKAP>, 2) averaging of the square of the point values <MSKAPS>, and 3) meridional averaging of the square of the zonal averages <MSKAZS>. Any or all of these averaging options may be selected, except for time series plotting (*PTIM), in which case only one option may be selected. Note, however, that the use of user-defined derived fields (*UDFL) provides another way of computing these statistics without this limitation.

The distribution of surface types must be available whenever masking is requested. It is assumed to be independent of time. If a history tape is used as input for Case A <TYPEc>, then the surface type information is automatically extracted from it and used for the masking. Otherwise, it is necessary to specify a Surface Type Save Tape (*SFCT) <SFCTTAP> with the same horizontal resolution as the data being averaged. A fatal error will result if the resolutions do not match.

There are two Surface Type Save Tapes archived on the TBM for use with CCMOA or CCMOB R15 data. They are SR15JA and SR15JL, which contain sea ice distributions for January and July, respectively. A Surface Type Save Tape can be created in any run in which a history tape is input <SFCTCRT>.

See (*UDFL) and the examples for <DERFLD> for an alternate method of masking using an arbitrary, time-dependent, input field.

->PLOT - Production of graphics

Primary ICPs: none

Secondary ICPs: <INDEX> <ICPECHO> <NUMPLT>

Execution of the Processor's graphics code does not result in the direct generation of plots, but rather in the production of a file containing graphics instructions (metacode). See (*PDIS) for a discussion of the disposition of this metafile.

In the following Topic Discussions, the plots produced by the Processor are categorized in two different ways. For some purposes, the plots are best grouped according to which physical dimensions (in space and time) are varied on the plot. The current plot dimension types are: 1) horizontal projections (*PHOR), 2) meridional cross sections (*PMEX), 3) latitudinal cross sections (*PLAX), 4) time series plots (*PTIM), and 5) spectral line graphs (*SPLG). For other purposes the plots are best grouped according to the means by which field values are represented. The current representation types are: 1) printed values (*PVAL), 2) line plots (*PLIN), 3) contour plots (*PCON), and 4) vector plots (*PVEC). Cross-references are used in all topics to indicate which display representations can be used with which dimensional configurations, and vice versa. There is also a general topic on controlling plot colors (*PCLR). In addition, the individual plot type topics describe which parts of each type of plot can be set to different colors.

Some Processor options (such as spatial averaging) will automatically produce a certain kind of plot (*ZONA) (*MERA), while other plots must always be requested explicitly (*PTIM). No matter how a plot is requested, its characteristics can be controlled with options that apply to all plots of the same dimension type. Except for some time series plot options (*PTIM), all plot characteristic options apply to all Cases and field passes.

For all plots, grid points are assumed to be equally spaced in longitude (as in the Model). Latitude, the vertical coordinate, and time are always located exactly with respect to the plot axes, even if the grid points are not equally spaced. Vertical levels are indicated on the plots by a coded character string: sigma values are multiplied by 1000 and have a suffix of "S"; pressure levels are in millibars and have a suffix of "P"; ground and surface levels are indicated by "1000.G" and "1000.S", respectively

(*XBTL). Except for ground level fields, the level indicated for all single level fields is "1000.S" regardless of the actual level location in the Model.

When the data are averaged in space or time, the actual averaging ranges used are indicated on most plots (requested ranges are adjusted if necessary according to actual grid point locations) (*SPAV). Although a complete list of coordinate values for each averaged dimension is stored with the data, the list is compressed on plot labels by assuming a constant increment. Spectral (*SPEC) and time filtering operations (*TFIL) are not currently indicated on any plots. See (*TAVG) for for a discussion of ranges for multiple-step time averages.

An index of the plot frames produced may be written to the plot and/or print file <INDEX>, the plot frames may be numbered <NUMPLT>, and the input ICPs may be echoed to the plot file <ICPECHO>.

-->PDIS - Disposition of plot file

Primary ICPs: none

Secondary ICPs: <DPLTMF> <DPLTCA> <DPLTCT> <DPLTIT> <DPLTFN>
<DPLTFT> <MNFRMS> <MXFRMS> <INDEX> <ICPECHO>
<NUMPLT>

The options discussed in this topic are machine-dependent, and refer only to the Cray-1 version of the Modular Processor.

Under most circumstances, disposition of the plot file is handled automatically by the Processor code, using the FORTRAN-callable version of NETDISP (see Computing Division documentation). This is the case even in the event of a fatal error, as long as the error is detected by the Processor. In the event of a fatal error detected by the Cray Operating System (COS), plot file disposition is handled by the JCL in the run deck. Errors of this type include TBM or Network staging problems, I/O errors, exceeding the job time limit, and floating point errors. The standard Processor run decks (*CJDK) contain JCL which disposes (to the Dicomed fiche unit) any plots not previously disposed during execution.

The NOACK parameter is used in the NETDISP call so that the job does not need to wait for acknowledgement from the mainframe(s) to which the plot file is being disposed. The use of NETDISP, however, prevents a job from being run in express class. Express class jobs can still be run, but only if all execution-time disposes are disabled (see below), and the JCL dispose is changed from NETDISP to DISPOSE.

Execution-time plot file disposition is controlled by a number of ICPs. A brief description of these ICPs follows; see the keyword descriptions in Part 3 for the details of using them.

<DPLTMF> - one or more mainframes (Network nodes) to receive the plot file(s);
<DPLTCA> - camera type for Dicomed disposes;
<DPLTIT> - plot title for Dicomed disposes;
<DPLTCT> - Case title to be used as Dicomed plot title if <DPLTIT> is not specified;
<MNFRMS> - minimum number of plot frames produced before an attempt to dispose the plot file at a logical breakpoint;
<MXFRMS> - maximum number of plot frames produced before an attempt at an immediate plot file dispose.
<INDEX> - controls copying of plot index to print file and/or disposed plot file(s).
<ICPECHO> - controls copying of input ICP data card images to disposed plot file(s).

The printed information added to the plot file(s) with the ICPs <INDEX> and <ICPECHO> is not metacode, but specific instructions meaningful only to the Dicomed. (This text is added using the NCAR printer simulation package PRSIM). The metafile editors on both the AAP VAX Cluster and the IBM4341 ignore these non-metacode records. When disposing plot files to other Network nodes, however, it may be necessary to set these ICPs so that the text information is not copied to the plot file(s).

The plots produced by one or more jobsteps (*DEFS) may be broken up into two or more groups which are disposed separately. This is done automatically if the size of the plot file approaches the maximum size which can be handled on the Dicomed, or if more than <MXFRMS> plot frames are produced. There is also an attempt to dispose the plot file at a logical breakpoint (e.g., between vertical cross-sections and horizontal projections) when more than <MNFRMS> plot frames have been produced. This may be useful for enhancing Dicomed throughput, since large files are given low priority. Whenever a plot group is disposed, a message is printed giving the group number and the number of plot frames in the dispose group. A separate plot index <INDEX> is generated for each dispose group. There is no attempt at execution-time dispose under any circumstances if <DPLTMF> is specified as 'NO'. If for any reason there are any undisposed plot frames at run termination (either normal or error exit), they are disposed by the JCL in the standard run deck (*CJDK).

Since the plot file is normally disposed at the end of each jobstep, and each dispose generates a separate piece of film on the Dicomed (or a separate file for other mainframes), each jobstep may produce a separate piece of film (file). This can be avoided, however, by specifying 'NO' for the ICP <DPLTMF> for all jobsteps except the last. Since the file containing plotting instructions is rewound only when it is disposed, each jobstep's plot frames are appended to any undisposed frames from previous jobsteps. A single plot dispose (and its associated index <INDEX>) may therefore contain plots generated by more than one jobstep. Plot frame numbers <NUMPLT> are formed by using the jobstep number, followed by a period, followed by the frame number within the jobstep. Frames containing

only text (<ICPECHO> and <INDEX>) are not numbered, and are not included in any frame counts.

Setting <DPLTMF> to 'NO' for all jobsteps totally disables all execution-time disposes of plot files. This allows the user to control disposes through the JCL by manipulating the plot datasets directly. There may be as many as four datasets involved; they are used by the Processor code as follows. The ICP input data card images are always copied to the dataset ICPEC0. This dataset is rewound at the beginning of each jobstep, so it contains only the ICPs for the current jobstep. If requested by the ICP <ICPECHO>, dataset ICPEC0 is copied (immediately after being written) to dataset PLTD using the Dicomed printer simulator package PRSIM. It is also unconditionally copied to the print file $OUT. All of the vector plot instructions (metacode) are written directly to the dataset $PLT as they are generated. As each frame is completed, a plot description line is added to the dataset PINDEX, regardless of the value of the ICP <INDEX>. At plot dispose time, The ICP <DPLTMF> is checked; if no dispose is requested, nothing further is done. If at least one mainframe is requested, the dispose procedure begins by copying PINDEX to PLTD, if requested by the ICP <INDEX>. Next, $PLT is copied to PLTD, and PLTD is then disposed to all requested mainframes, in the order in which they are specified. PINDEX, $PLT, and PLTD are then rewound, completing the dispose procedure. No end-of-files are ever written to any of these datasets.

Proper handling of plot files requires communication between jobsteps. This is accomplished through the use of Cray JCL global pseudo-registers, which can be manipulated by both the JCL and the executing code (see the Cray COS and Library Routines manuals). The following registers are used by the Processor: (*CJDK)

G0 - used to request ICP check option
G1 - total number of jobsteps requested
G2 - current jobstep number
G3 - number of plot frames in $PLT not yet disposed
G4 - current plot dispose group number
(G4 minus 1 is the number of disposes in the run so far)

-->PHOR - Plotting horizontal projections

All horizontal projection plots (*PHOR) display field values on a longitude-latitude grid. The grid projection may be either rectangular (*PHRE) or polar (*PHPO).

--->PHRE - Plotting rectangular horizontal projections

Primary ICPs: [HPROJ]

Secondary ICPs: <HPPTVAL> <HPCINT> <HPSCAL> <HPCDIV> <DASHLIN>
<HPLFPVn> <HPVOPT> <HPVSCAL> <HPVDIN> <CLCNTNT>

When rectangular horizontal projection plots are requested [HPROJ], each level of each field being processed is plotted on a separate frame. In this type of horizontal projection, the entire globe is projected as a longitude-latitude rectangle using an equatorial cylindrical projection. Distances along the axes are proportional to latitude or longitude, resulting in exaggerated areas at high latitudes. Both the left and right edges of the plot correspond to 180 degrees East longitude, where grid point values are always defined and represented. The top and bottom edges of the plot correspond to the North and South Poles, respectively, where grid point values are never defined; representation of field values ends at the grid points closest to the poles. Continental outlines are always drawn, and their color can be controlled <CLCNTNT>. Field values may be printed on the plot (*PVAL) and/or contoured (*PCON) <HPPTVAL>, and any two fields may be combined as vector components represented as arrows (*PVEC) <HPLFPV>. Contours of vector magnitude may be superimposed over the vector arrows <HPVOPT>.

For contour plots (*PCON), the contour interval <HPCINT>, scale factor for numeric labels <HPSCAL>, and a dividing value for countour line attributes <HPCDIV> <DASHLIN> may all be controlled individually for each level of each field.

For vector plots, the vector length scale factor <HPVSCAL>, and vector plotting density <HPVDIN> may both be controlled individually for each level of each field.

For contour, vector, and point value plots, specific vertical levels of specific fields can be skipped individually <HPCINT>. See the topic which describes the field value representation type for a description of other color options (*PCLR) (*PCON) (*PVEC) (*PVAL).

--->PHPO - Plotting polar horizontal projections

Primary ICPs: [HPROJ]

Secondary ICPs: <HEMIS> <HPCINT> <HPCDIV> <HPSCAL> <DASHLIN>
<HPLFPVn> <HPVOPT> <HPVSCAL> <HPVDIN> <CLCNTNT>

When polar horizontal projection plots are requested [HPROJ], each level of each field is plotted. In this type of horizontal projection, each polar hemisphere is projected (on a separate frame) as a circle centered on the pole. A stereographic azimuthal projection is used, resulting in a radial scaling of latitude that is proportional to the tangent of the angular distance to the pole. Continental outlines are always drawn, their color can be controlled <CLCNTNT>, and either or both hemispheres may be plotted <HEMIS>. Field values may be contoured (*PCON), or any two fields may be combined as vector components represented as arrows (*PVEC) <HPLFPV>. Contours of vector magnitude may be superimposed over the vector arrows <HPVOPT>. Field values cannot be printed on the plot (this would not be useful due to crowding at the poles). See the topic which describes the

field value representation type for a description of other color options (*PCLR) (*PCON) (*PVEC).

For contour plots, the contour interval <HPCINT>, line label scale factor <HPSCAL>, and a dividing value for countour line attributes <HPCDIV> <DASHLIN> may all be controlled individually for each level of each field. Specific vertical levels of specific fields may be skipped <HPCINT>.

For vector plots, the vector length scale factor <HPVSCAL>, and vector plotting density <HPVDIN> may both be controlled individually for each level of each field. Specific levels of specific fields may be skipped <HPVSCAL>.

See the topic which describes the field value representation type for a description of color options (*PCLR) (*PVAL) (*PCON) (*PVEC).

An internal Processor parameter determines the cutoff latitude at the plot's perimeter; this is currently set at 70 degrees from either pole, so that the cutoff latitude is 20 degrees North or South, depending on the polar hemisphere being plotted. This cutoff latitude is exact regardless of the latitudes at which grid points are defined; an interpolation which is linear with respect to latitude is used to determine field values at the plot perimeter.

-->PMEX - Plotting meridional cross sections

Primary ICPs: none

Secondary ICPs: <ZONAVG> <MXPLOT> <MXLNSCL> <MXLATRV> <MXSIZE>
<MXASPRT> <LMLFPSL> <MXCINT> <MXSCAL> <MXCDIV>
<DASHLIN> <MXPTVAL> <MXLTRNG> <ZORD>

When zonal averages (*ZONA) are requested <ZONAVG>, the resultant fields are automatically plotted as meridional cross sections. If meridional cross sections are explicitly requested <MXPLOT>, then each field is plotted at each available longitude, before any requested zonal averaging is done (*ORDR). The range of latitudes to be plotted may also be specified <MXLTRNG>.

For multilevel fields, a meridional cross-section plot uses latitude as the abscissa and the height coordinate (either sigma or pressure) as the ordinate. The ordinate may be scaled logarithmically or linearly <MXLNSCL>, the North Pole may be plotted on either the left or right side of the frame <MXLATRV>, and the size <MXSIZE> and aspect ratio <MXASPRT> of the plot may be specified. The ordinate tick marks are placed at the levels of the data points, and are labelled with a character string indicating the level (*PLOT). The interior tick marks on the abscissa show the exact locations of the latitudes at which grid points are defined. Labelled tick marks giving the height (in km) may also be drawn <ZORD>. A 7 km scale height is used to relate height to the plot ordinate, and a surface pressure of 1013.2

millibars is assumed when the ordinate is pressure.

For fields which are defined at only one level, field value replaces the vertical coordinate as the ordinate, and the plot is a single curve (*PLIN). Multilevel fields may also be graphed this way (all levels, one level per plot) in addition to being plotted with vertical variation <LMLFPSL>.

On plots with vertical variation, the field values may be contoured (*PCON) and/or printed (*PVAL) <MXPTVAL>. Vectors cannot be drawn. For contour plots, the contour interval <MXCINT>, scale factor for numeric values <MXSCAL>, and a dividing value for countour line attributes <MXCDIV> <DASHLIN> may all be set individually for each field. For both contour and point value plots, specific fields can be skipped individually <MXCINT>. See the topic which describes the field value representation type for a description of other color options (*PCLR) (*PVAL) (*PCON).

-->PLAX - Plotting latitudinal cross sections

Primary ICPs: none

Secondary ICPs: <MERAVG> <LXPLOT> <LXLNSCL> <LXSIZE> <LXASPRT>
<LMLFPSL> <LXCINT> <LXSCAL> <LXCDIV> <DASHLIN>
<LXPTVAL> <ZORD>

When meridional averages are requested <MERAVG> (*MERA), each meridionally averaged field is automatically plotted as a latitudinal cross section. If latitudinal cross sections are explicitly requested <LXPLOT>, then each field is plotted at each available latitude, before any requested meridional averaging is done (*ORDR).

For multilevel fields, a latitudinal cross-section plot uses longitude as the abscissa and the height coordinate (either sigma or pressure) as the ordinate. The ordinate may be scaled logarithmically or linearly <LXLNSCL> and the size <LXSIZE> and aspect ratio <LXASPRT> of the plot may be specified. The ordinate tick marks are placed at the levels of the data points, and are labelled with a character string indicating the level (*PLOT). The interior tick marks on the abscissa show the exact locations of the longitudes at which grid points are defined. Labelled tick marks giving the height (in km) may also be drawn <ZORD>. A 7 km scale height is used to relate height to the plot ordinate, and a surface pressure of 1013.2 millibars is assumed when the ordinate is pressure.

For fields which are defined at only one level, field value replaces the vertical coordinate as the ordinate, and the plot is a single curve (*PLIN). Multilevel fields may also be plotted this way (all levels, one level per plot) in addition to being plotted with vertical variation <LMLFPSL>.

On plots with vertical variation, the field values may be contoured (*PCON) and/or printed (*PVAL) <LXPTVAL>. Vectors cannot be drawn. For contour plots, the contour interval <LXCINT>, scale factor for numeric values <LXSCAL>, and a dividing value for countour line attributes <LXCDIV> <DASHLIN> may all be set individually for each field. For both contour and point value plots, specific fields can be skipped individually <LXCINT>. See the topic which describes the field value representation type for a description of other color options (*PCLR) (*PVAL) (*PCON).

-->PTIM - Plotting time series

Primary ICPs: [FIELDcn]

Secondary ICPs: <DIFFLDn> <RATFLDN> <SAVTSPW> <SAVTSPR>

There are currently five different types of time series which can be generated and plotted:

1) Time vs. field value for a given field, level, latitude, and longitude (*PTFL)
2) Time vs. zonal average for a given field, level, and latitude (*PTZL)
3) Time vs. masked area average for a given field, level, latitude-longitude rectangle, and land-ocean-sea ice mask (*PTAL)
4) Zonal averages for a given field and level contoured on a time vs. latitude grid (*PTZC)
5) Meridional averages for a given field and level contoured on a longitude vs. time grid (Hovmoller plots) (*PTMC).

Any input or derived field may be plotted as any or all time series types, and may be plotted more than once within a single type (e.g., different levels, latitudes, areas, etc.). Each field-level to be plotted must be requested separately for each of the three Cases (A, B, and C). Line plots for Cases A and B may be paired on the same axes. All fields to be plotted as time series must be explicitly requested for processing [FIELDcn] <DIFFLDn> <RATFLDn> for the Case and/or field pass in which they are to be plotted. There is an additional restriction: time series plots can be produced ONLY FOR FIELD PASS NUMBER 1. All other processing options may be requested in the same run, with the following exception: time series plots cannot be produced for Case C (differences and ratios) unless the Model days are processed individually (i.e., no time averaging or time filtering).
Implementation Note: This restriction is necessitated by the ordering of processing operations (*ORDR), but it can be removed with simple coding modifications that change the order of Module calls.

There are no option combination restrictions for Cases A or B. If time filtering is also requested, the filtered time series is plotted (*TFIL). Any spatial averaging or masking done for the time series plots is independent of separately-requested averaging or masking options. The time series used for the plots is the same time series used for the other

processing options (time filtering shortens the series). The time coordinate on all time series plots is labelled with Model day values unless the time series being processed consists of a series of time averages (from a set of Time Average Save Tapes), or is a case ensemble <ENSEMBLc>. In both of these kinds of runs, the time coordinate is labelled from 1 to the number of time samples in the series. On plots containing time series for both Cases A and B, the first days for each case are aligned, and the time coordinate is determined by Case A.

For all types of time series plots, the complete time series may be written to a Save tape as well as being plotted (*STSP) <SAVTSPW>. On subsequent runs, this Save tape can be input <SAVTSPR>, and the time series can be replotted with different plot characteristics. Since each type of time series involves spatial compression in at least one dimension, the use of these Save tapes can result in considerable savings in processing time.

--->PTFL - Plotting time vs. field value

Primary ICPs: [TSPPLcn]

Secondary ICPs: <TSPFNP> <TSPFPH> <TSLPASP> <TSLPSIZ> <CLTSLPc> <DASHLIN>

This type of time series plot is a line plot (*PLIN) with time as the abscissa and field value at a particular point in three-dimensional space as the ordinate. Each plot is requested separately by specifying the field name, latitude, longitude, vertical coordinate value, and (optionally) an ordinate range [TSPPLcn]. The ordinate range is set at appropriate "nice" values if it is not specified. Fields from different Cases may be paired and plotted on the same axes <TSPFNP> in addition to or instead of being plotted individually <TSPFPH>. On such paired plots, the Case A time series is always a solid line. Dashing of the curve for Case B is controlled by <DASHLIN>. Time series colors are controlled by <CLTSLPc> for both paired and unpaired plots. If different ordinate ranges are specified for paired fields, then the paired plot will have the minimally expanded range which encompasses both individual ranges. The size <TSLPSIZ> and aspect ratio <TSLPASP> used for all time series line plots may be specified.

--->PTZL - Plotting time vs. zonal average

Primary ICPs: [TSPZLcn]

Secondary ICPs: <TSPFNP> <TSPFPH> <TSLPASP> <TSLPSIZ> <CLTSLPc> <DASHLIN> <ZAVRNG> <ZBKFR>

This type of time series plot is a line plot (*PLIN) with time as the abscissa and zonal average of a particular latitude and vertical level the ordinate. Each plot is requested separately by specifying the field name, latitude, vertical coordinate value, and (optionally) an ordinate range [TSPZLcn]. The ordinate range is set at appropriate "nice" values if it is

not specified. Fields from different Cases may be paired and plotted on the same axes <TSPFNP> in addition to or instead of being plotted individually <TSPFPH>. On such paired plots, the Case A time series is always a solid line. Dashing of the curve for Case B is controlled by <DASHLIN>. Time series colors are controlled by <CLTSLPc> for both paired and unpaired plots. If different ordinate ranges are specified for paired fields, then the paired plot will have the minimally expanded range which encompasses both individual ranges. The size <TSLPSIZ> and aspect ratio <TSLPASP> used for all time series line plots may be specified.

Although the zonal averaging for time series is independent of the general zonal averaging (*ZONA) requested with the ICP <ZONAVG>, both averaging procedures share the same ICPs for limiting the longitude range <ZAVRNG> and setting the minimum required fraction of unblocked points <ZBKFR>.

--->PTAL - Plotting time vs. masked area average

Primary ICPs: [TSPALcn]

Secondary ICPs: <TSPFNP> <TSPFPH> <TSLPASP> <TSLPSIZ> <CLTSLPc> <DASHLIN> <SFCTTAP> <MSKAVPR> <MSKAP> <MSKAPS> <MSKAZS>

This type of time series plot is a line plot (*PLIN) with time as the abscissa and masked (horizontal) area average the ordinate. A masked area average is a two-dimensional average over an arbitrary latitude-longitude rectangle, with a surface type mask applied to the rectangle in order to limit the averaging to points above a particular surface type or types. Any combination of land, ocean, and sea ice masks may be specified; combining all three results in no masking, i.e., no exclusions due to surface type. Different masks and rectangles may be specified for each field to be averaged, and each field may be averaged more than once. Gaussian weights are applied at each latitude to account for the variable area represented by each point contributing to the average.

Each plot is requested separately by specifying the field name, the latitude-longitude rectangle, the vertical coordinate value, the surface type mask, and (optionally) an ordinate range [TSPALcn]. The ordinate range is set at appropriate "nice" values if it is not specified. Fields from different Cases may be paired and plotted on the same axes <TSPFNP> in addition to (or instead of) being plotted individually <TSPFPH>. On such paired plots, the Case A time series is always a solid line. Dashing of the curve for Case B is controlled by <DASHLIN>. Time series colors are controlled by <CLTSLPc> for both paired and unpaired plots. If different ordinate ranges are specified for paired fields, then the paired plot will have the minimally expanded range which encompasses both individual ranges. The size <TSLPSIZ> and aspect ratio <TSLPASP> used for all time series line plots may be specified.

Masked area averaging for time series is independent of the general masked area averaging (*HORA) requested with the ICP <MSKFLcn>. However, both averages are computed in the same way (the same Module is used) and both options have the same requirements for access to the Model's distribution of surface types if masking is requested. If a history tape is used as input for Case A <TYPEc>, then the surface type information is automatically extracted from it and used for the masking. Otherwise, it is necessary to specify a surface type Save tape (*SFCT) <SFCTTAP> with the same horizontal resolution as the data being averaged. A fatal error will result if the resolutions do not match.

There are two Surface Type Save Tapes archived on the TBM for use with CCMOA or CCMOB R15 data. They are SR15JA and SR15JL, which contain sea ice distributions for January and July, respectively. A surface type Save tape can be created in any run in which a history tape is input <SFCTCRT>.

The ICP <MSKAVPR> may be used to control printing of masked area averages as they are computed.

The ICPs <MSKAP>, <MSKAPS>, and <MSKAZS> may be used to modify the horizontal area average computations (*HORA), except that only one of these ICPs may be set to 'YES'. The averaging can also be modified by creating user-derived fields (*UDFL) and averaging them.

--->PTZC - Plotting zonal average contours (time vs. latitude)

Primary ICPs: [TSPZCcn]

Secondary ICPs: <TSZCASP> <TSZCSIZ> <DASHLIN> <ZAVRNG> <ZBKFR>

This type of time series plot is a contour plot (*PCON) with time as the abscissa and latitude as the ordinate. The zonal averages for all latitudes are contoured on this two-dimensional grid. Each plot is requested separately by specifying the field name, vertical coordinate value, and (optionally) the contour interval, line label scale factor, and contour dividing value [TSPZCcn].

Although the zonal averaging for time series is independent of the general zonal averaging (*ZONA) requested with the ICP <ZONAVG>, both averaging procedures share the same ICPs for limiting the longitude range <ZAVRNG> and setting the minimum required fraction of unblocked points <ZBKFR>.

Other plot characteristics (such as contour line dashing <DASHLIN> and coloring) are determined by the contour plot ICPs (*PCON).

--->PTMC - Plotting meridional average contours (longitude vs. time)

Primary ICPs: [TSPMCcn]

Secondary ICPs: <TSMCASP> <TSMCSIZ> <DASHLIN> <MAVGRNG> <MBKFR>

This type of time series plot is a contour plot (*PCON) with longitude as the abscissa and time (increasing downwards) as the ordinate. The meridional averages for all longitudes are contoured on this two-dimensional grid. This type of plot is often referred to as a "Hovmoller plot". Each plot is requested separately by specifying the field name, vertical coordinate value, and (optionally) the contour interval, line label scale factor, and contour dividing value [TSPMCcn].

Although the meridional averaging for time series is independent of the general meridional averaging (*MERA) requested with the ICP <MERAVG>, both averaging procedures share the same ICPs for limiting the latitude range <MAVRNG> and setting the minimum required fraction of unblocked points <MBKFR>.

Other plot characteristics (such as contour line dashing <DASHLIN> and coloring) are determined by the contour plot ICPs (*PCON).

-->SPGR - Spectral graphics

Primary ICPs: [SPCcn] [SPSNGRF]

Secondary ICPs: <SPGYINT> <TIMAVGc>

If spectral processing is requested [SPCcn], then the square of the norm may be graphed for all spectral fields [SPSNGRF]. The ordinate on these graphs is the natural log of the sum over Fourier wavenumber of the SHC times its complex conjugate. The abscissa is the natural log of the order of the Legendre polynomial. The range of the ordinate may be controlled with the ICP <SPGYINT>. Nonlinear spectral derived fields (*DFLD) are automatically deleted after graphing, since they have no meaning in gridpoint space.

The graphed fields may also be time averaged <TIMAVGc> before they are graphed.

-->PVAL - Plotting point values

Primary ICPs: [HPPTVAL] [MXPTVAL] [LXPTVAL]

Secondary ICPs: <CLPTVL>

In this type of field value representation, numbers are scaled and printed directly on the plot, centered over the corresponding grid point. Values are not printed for "blocked points". Usually only every second or

third point along the latitude or longitude dimensions can be fit on the plot; the appropriate density is determined automatically. Point value representations can be requested instead of or in addition to contours (on separate frames) for horizontal rectangular projections [HPPTVAL] (*PHRE), and meridional [MXPTVAL] (*PMEX) and latitudinal [LXPTVAL] (*PLAX) cross sections of multilevel fields. On horizontal projections, the point values are superimposed on continental outlines. The color (*PCLR) of the point value characters can be specified <CLRPTVL>.

-->PLIN - Plotting line graphs

Primary ICPs: [LMLFPSL]

Secondary ICPs: <CLCTRGE> <CLTSLPc>

If a plot has only one varying space or time dimension, then field value is used as the coordinate for one of the plot axes, and values are represented as a line (two-dimensional curve). The range of the ordinate is set automatically using "nice" values that enclose the full field value range. "Blocked points" appear as gaps in the curve. This type of plot is used in meridional and latitudinal cross sections (*PMEX) (*PLAX) when the field being plotted has no vertical variation; this type of plot may also be explicitly requested for individual levels of multilevel fields [LMLFPSL]. The color (*PCLR) of these lines is controlled by <CLCTRGE>. Line graphs are also used for some types of time series plots (*PTIM); the colors of these lines are controlled with input parameters <CLTSLPc>.

-->PCON - Plotting contours

Primary ICPs: none

Secondary ICPs: <CLCTRGE> <CLCTRLT> <CLHIGHS> <CLLOWS> <DASHLIN>

If a plot has two varying space or time dimensions, then values can be represented by contouring the two-dimensional field. Major contour lines are labelled with a value, and closed highs and lows are marked and labelled. "Blocked points" appear as holes in the contour pattern. This type of plot is used for horizontal projections (*PHOR), meridional (*PMEX) and latitudinal (*PLAX) cross sections, and for some types of time series plots (*PTIM).

On virtually all contour plots, the contour levels are equally spaced: each level is a multiple of a contour interval. The contour interval, line label scale factor, and contour dividing value may be chosen automatically or specified for each plot individually, with one exception: vertical cross sections of mixing ratio use a special set of contour levels equally spaced for large mixing ratios, and logarithmically spaced for small values (*MIXR). This set of levels can be changed only by modifying the code (*ARCH).

The contour interval, scale factor for numeric values, and a contour dividing value (described below) can all be specified for all contour plots individually. There is a separate ICP for each of these three quantities, for each contour plot dimension type. See the appropriate dimension type topic for a list of ICPs.

If the contour interval is defaulted or set to 0., it is determined automatically, based solely on the minimum and maximum of the values to be contoured. Since the algorithm tends toward a predetermined number of contour levels, it often produces tightly-spaced contours for noisy fields.

Major contour lines are labelled with the contour value. In order to reduce the number of digits printed, this value is usually multiplied by a scale factor before printing. The same scale factor is used for labelling closed highs and lows. If defaulted or set to 0., the scale factor is determined automatically. It is printed at the bottom of the plot, unless its value is one.

If the contour dashing option is enabled <DASHLIN>, then all contour lines for values less than the dividing value are dashed. The color of lines for contour levels greater than or equal to the dividing value can be specified <CLCTRGE> independently of the color for contours less than the dividing value <CLCTRLT>. The colors used for labelling highs and lows can also be controlled <CLHIGHS> <CLLOWS>.

All data points are located exactly with respect to the coordinate axes. Contour lines are located between grid points by performing a one-dimensional interpolation which is linear with respect to distance on the frame (this is not necessarily linear with respect to the physical coordinate).

-->PVEC - Plotting vectors

Primary ICPs: [HPLFPVn]

Secondary ICPs: <HPVOPT> <HPVSCAL> <HPVDIN> <CLVLTD> <CLVGED> <HPVDIV>

On horizontal projection plots (*PHOR), any two fields can be designated as a pair to be combined and plotted as vector arrows [HPLFPVn]. No vectors are plotted at "blocked points". This type of plot can be produced in addition to or instead of contour plots for the scalar fields, or contours of vector magnitude may be superimposed on the arrows <HPVOPT>. The tail of each vector is plotted at the point's location, with the arrow pointing in the vector's direction (defined by the plot axes). The length of the arrow is directly proportional to the vector magnitude at the point's location. The scaling can be determined automatically based on the average vector magnitude, or it can be specified for each vector plot <HPVSCAL>. There is also an option to make all arrows the same length <HPVSCAL>. Individual vector plots may also be skipped entirely <HPVSCAL>. Since plotting arrows

at every grid point could result in overcrowding, grid points may be skipped by specifying a density increment <HPVDIN>.

The color of arrows for vectors whose magnitude is greater than or equal to a dividing value <HPVDIV> can be specified <CLVGED> independently of the color for vectors whose magnitude is less than the dividing value <CLVLTD>.

-->PCLR - Plotting in color

Primary ICPs: none

Secondary ICPs: <CLLABEL> <CLCNTNT> <CLCTRGE> <CLCTRLT> <CLHIGHS>
<CLLOWS> <CLPTVL> <CLVLTD> <CLVGED> <HPVDIV>
<HPCDIV> <MXCDIV> <LXCDIV> <CLTSLPc> <TSPZCan>
<TSPMCan> <DASHLIN>

Since the NCAR metacode system (and the plot utilities) do not currently contain provisions for color filling of areas, the Processor's color capabilities are limited to setting different line colors for different parts of plots. This can be done, however, for all major parts of all plots.

The NCAR standard for color specification uses a three-character numeric Hollerith string to represent each color. The correspondence between these numbers and plotted colors is determined by the translation of the metacode to a particular plot device's instruction set. This is usually set up in a way which allows the user to define or modify a "color table" used to translate the numbers into colors that can be displayed on the plotting device. This means that it does not really matter what numbers are used to specify colors in the Processor, as long as a different number is used for each part of the plot that is to be displayed as a different color; the colors can be reset later without having to modify the plot instruction metafile.

The metafile output by the Processor always contains instructions to change colors, with all colors for all plots set to the default color code '001'. These color change instructions are ignored by the metacode translators for plot devices without color capabilities.

The Processor's color specification options are organized primarily by field value representation types, however some colors may also be specified independently for different plot dimension types (*PLOT). See the Topic Discussion for each plot type to determine how its colors can be controlled.

There is one plot color which is common to all plots. It is the color of plot perimeters, tick marks, and labels (descriptive text) <CLLABEL>. Also, the color specified by <CLCTRGE> is used on both contour and some line plots.

<-> end of Part 2. Descriptions of Processing Options

<-> Part 3. List of Input Control Parameter (ICP) Keywords

All ICP keywords and their default values are listed alphabetically below. Case and field pass suffixes are symbolically compressed (lower case "c" and "n"). To locate the desired keyword, search for the keyword prefixed with a ">".

->BPHSTc Determines whether a History Save Tape (*OHST) can be written when there are blocked points (*DEFS). One value, CHARACTER, 'YES' or 'NO', default is 'NO'. If the value is 'NO', then all field values are checked for the blocked point value (1.E36) as the History Save Tape(s) are written; if this value is encountered, an error abort results. Setting this ICP to 'YES' may cause problems in subsequent runs, since the Modular Processor does not check for blocked points when computing derived fields of types 11, 12, and 13. Also, a history tape containing blocked points cannot be input to the Model.

->CLCNTNT Color (*PCLR) of continental outlines on horizontal projection plots (*PHOR). (one value, CHARACTER, default is '001').

->CLCTRGE Color (*PCLR) of plotted contours (*PCON) greater than or equal to the dividing value <HPCDIV> <MXCDIV> <LXCDIV> <TSZPCcn> <TSPMCcn>. (one value, CHARACTER, default is '001'). This is also the line color for line plots (*PLIN).

->CLCTRLT Color (*PCLR) of plotted contours less than the dividing value <HPCDIV> <MXCDIV> <LXCDIV> <TSZPCcn> <TSPMCcn>. (one value, CHARACTER, default is '001').

->CLHIGHS Color (*PCLR) of highs (H's and values) on all contour plots (*PCON). (one value, CHARACTER, default is '001').

->CLLABEL Color (*PCLR) of labels and borders on all plots (*PLOT). (one value, CHARACTER, default is '001').

->CLLOWS Color (*PCLR) of lows (L's and values) on all contour plots (*PCON). (one value, CHARACTER, default is '001').

->CLPNTVL Color (*PCLR) of point values on point value plots (*PVAL). (one value, CHARACTER, default is '001').

->CLTSLPc Color (*PCLR) of Case c lines for all time series line plots (*PTIM). (one value, CHARACTER, default is '001').

->CLVGED Color (*PCLR) of plotted vectors (*PVEC) greater than or equal to the dividing value <HPVDIV>. (one value, CHARACTER, default is '001').

->CLVLTD Color (*PCLR) of plotted vectors (*PVEC) whose magnitude is less than the dividing value <HPVDIV>. (one value, CHARACTER, default is '001').

->CVFLDcn Triplets of field names for time covariance computations (*TAVG).

(covariance = $\overline{x'y'}$)
The first two names are any requested fields for the same Case and field pass [FIELDcn] (derived fields OK); the third field name is the name by which the covariance is to be known. The covariances are computed only if time averages are also requested for the corresponding Case [TIMAVGc]. The fields needed to compute the covariances must be explicitly requested via the [FIELDcn] ICP, but the computed covariance fields should NOT be requested via the [FIELDcn] ICP. (Cases A and B only, up to 100 triplets for each keyword, up to 8 CHARACTERs per name, default is no covariances computed).

Example:
```
FIELDA1='T','U','V'
TIMAVGA='YES'
CVFLDA1='T','U','COV-TU','T','V','COV-TV'
```

These data cards request two covariance fields: the covariance of T and U, and the covariance of T and V, both for Case A, field pass 1.

->DASHLIN Controls the dashing of certain lines on contour (*PCON) and time series (*PTIM) plots. (one value, CHARACTER, default is 'YES').

='YES' - contour lines less than the dividing value (*PCON), and Case B time series lines (*PTIM) are both dashed
='NO' - no lines are dashed on any plots (most useful for color plotting) (*PCLR).

->DAYSc Model times (days) to process. Each specified "day" refers to a single, instantaneous Model output time, measured in units of days (i.e., number of 24-hour periods, including fraction), relative to the start of Model integration (*DEFS). The list may be specified in either of the following forms:

* Complete list of days, in any order. (up to 1000 days, all REAL). Several special flag values may be used to request days according to their position on the input tapes. These flag values are described below.

-1. - The first day on the first input tape (REAL).
-2. - The next day encountered on the input history tapes (i.e., the day following the day previously processed) (REAL). If this is the first value in the list, it is equivalent to -1. This special value should be avoided when both Cases A and B are sharing the same input tapes, since the next day for Case B may be the day following the day just processed for Case A.
-3. - The last day on the last input tape (REAL).

Volume positioning is most efficient if the days are specified in the order in which they are encountered in the [TAPESc] list. Any day specifier may be repeated; the same day will be processed more than once if so requested.

* First day, last day, interval (do-loop notation). All days in the expanded list are processed. The negative special values described above should not be used with this form of input. (3 values, all REAL, maximum 1000 days when expanded).

Notes:

1. Literal lists containing 3 days are sometimes ambiguous. Any days list of length 3 is interpreted as do loop notation if the third day is less than or equal to the second day, unless the third day is one of the special negative flag values.
2. The day value specified (or computed from do-loop notation) must be accurate to three decimal places (.001 days).
3. This ICP is ignored if Time Average Save Tapes are input for the corresponding Case <TYPEc>, since there is only one day on each of these tapes.
4. The relationship of the requested days to the requested input tapes (i.e., which days are searched for on which tapes) is affected by the value assigned to <ENSMBLc>; see the discussion of case ensemble processing (*ENSB) for details.

Examples:
1. DAYSA=101.,101.5,102.,,102.5,103.,103.5,104.,104.5,105.
2. DAYSA=101.,105.,0.5
3. DAYSA=-1.,-3.,
4. DAYSA=10:-2.

Examples 1 and 2 are equivalent; they both request the processing of nine time samples. Example 3 requests processing of the first and last time samples available on the input tape(s). Example 4 requests the processing of the first ten time samples available on the input tape(s).

->DELPDN List of permanent datasets which are to be deleted from the Cray disk (if they exist there) before beginning processing. All editions of each dataset (up to 100) are deleted. This will force a stage-up from the TMS-4 before the corresponding volume is read. (up to 40 names, CHARACTER, default is no permanent datasets deleted). This option is useful when a bad dataset copy on the Cray disk is suspected (*MSDN).

->DELREL Controls disposition of permanent datasets (on the Cray disk) corresponding to the previous input tape when the search for a day moves to a new tape (one value, INTEGER, default is 1) (*MSDV). See documentation for the Cray Operating System (COS) for definitions of RELEASE and DELETE.

= 0 - previous dataset is neither RELEASEd nor DELETEd
= 1 - previous dataset is RELEASEd, but not DELETEd,
= 2 - previous dataset is RELEASEd and DELETEd.

->DERFLD List of derived field definitions, containing up to 1000 values (mixed mode). See (*DFLD) and (*UDFL) for a discussion of user-defined derived fields. This ICP only defines new derived fields; they must be independently requested if they are to be computed <FIELDcn>. The list consists of a number of consecutive groups of values, with each group defining a different derived field. The maximum number of derived fields which may be defined with this ICP is about 75 or 80. (The exact number depends on the number of code-defined derived fields (*CDFL); the maximum for all types of derived fields combined is 150.) The order of the definition groups may be significant; see Note 1 below. Within each group, the values are interpreted as follows:

1) name of the new derived field (up to 8 CHARACTERs). The name must not begin with a "." or ":", and it must not be the same as any other derived field unless both have a (different) computation type code in the range 11-14. The field name may have the same name as a field on the input tape, in which case the ambiguity is resolved by the ICP <FLDSRCc>.

2) computation type (INTEGER). This is a code number used to describe when the derived field is to be computed relative to other processing steps (*ORDR). See (*DFLD) for additional details. The currently available types are listed below in the order in which they are computed. Note that computations in spectral space are excluded from this list, but see (*VDFL) for a method of computing some new filds in spectral space. For those computation types described as being computed "before" or "after" some processing step, it is not always necessary to request that step in order to compute the derived field. The exceptions are computation types 51, 71, and 81, which can be computed only when time filtering, zonally

averaging, or meridionally averaging, respectively. All types can be computed for Cases A and B; only types 61, 62, 71, and 81 can be computed for Case C (*ORDR).

11 - during input from a CCMOB history tape
12 - during input from a CCMOA history tape
13 - during input from a pressure level history tape
14 - during input from a Time Average or Time Series Save tape
31 - after spectral operations
41 - after vertical interpolation
51 - after time filtering, inside the time loop
61 - before vertical averaging (after time averaging)
62 - after vertical averaging
71 - after zonal averaging (operates on the averages)
81 - after meridional averaging (operates on the averages)

3) vertical placement flag for new derived field (INTEGER).

0 - for all single level fields
1 - for multilevel fields located at layer interfaces
2 - for multilevel fields located at layer midpoints, and all other vertical coordinate levels (e.g., pressure levels)

4) vertical coordinate flag used to determine on which vertical coordinate(s) the new derived field can be computed (INTEGER). Bit 1 is least significant.
bit 1 on - OK to compute on sigma surfaces
bit 2 on - OK to compute on pressure surfaces
For example, a value of 1 means the field can be computed only on sigma surfaces; a value of 3 means the field can be computed on either pressure or sigma surfaces.

5) spectral parity (INTEGER). This flag is used only when spectrally processing output from a hemispheric Model; in most cases it may be set to zero.

0 - parity not applicable
1 - field is odd
2 - field is even

6-?) the remaining values in the definition group must form a valid Reverse Polish Notation (RPN) expression which defines how the derived field is to be computed. Each value must be either an operand or an operator (function). Operands may be either a field name (up to 8 CHARACTERs), or a constant (REAL). Functions are classified as either unary, i.e., operating on a single operand, or binary, i.e., operating on two operands. All function names consist of up to 8

CHARACTERs; the first character is always "." for unary functions, and ":" for binary functions. Valid functions are listed below. The number of values in a defining expression is arbitrary. Each expression must be terminated by the unary function ".END", with the first value of the next definition group (if any) immediately following. A single defining expression may reference at most six different fields, but there is no limit to multiple references to the same field, and there is no limit to the definition nesting depth (i.e., one derived field may be based on another derived field, which may in turn be based on another derived field, etc.). Expressions are interpreted using standard RPN logic with a stack containing four vector registers. See (*UDFL) for a description of the expression evaluation algorithm.

List of Unary Functions

All operands must be fields, except for .CONST, which must have a scalar operand.

.END	expression and definition group terminator
.CONST	create a constant-value multilevel field
.MINUS	negate operand
.ABS	absolute value
.SQRT	square root
.ALOG	natural logarithm
.ALOG10	common (base 10) logarithm
.EXP	e ** operand
.VSUM	vertical sum of all levels
.DSWVSUM	delta sigma weighted vertical sum of all levels
.DSTIMES	delta sigma times each level
.TSINL	multiply operand by sine of latitude
.TCOSL	multiply operand by cosine of latitude
.ZAVDEV	replace operand by deviations from zonal average
.LEVELnn	extract level nn (counting from bottom) (result is the single, extracted level)

List of Binary Functions

OP1 is the leftmost operand, OP2 the rightmost. At least one operand must be a field, unless otherwise noted.

:PLUS	OP1 + OP2	
:MINUS	OP1 - OP2	
:TIMES	OP1 * OP2	
:DIVIDE	OP1 / OP2	
:MIN	minimum(OP1,OP2)	
:MAX	maximum(OP1,OP2)	
:POWER	OP1 ** OP2	
:AMOD	mod(OP1,OP2)	
:L1TIMES	OP1*(bottom level of OP2)	(two fields only)
:EQ	.T. iff OP1.EQ.OP2	
:NE	.T. iff OP1.NE.OP2	
:LT	.T. iff OP1.LT.OP2	
:GT	.T. iff OP1.GT.OP2	
:LE	.T. iff OP1.LE.OP2	
:GE	.T. iff OP1.GE.OP2	
:AND	.T. iff OP1.AND.OP2	(two fields only)
:OR	.T. iff OP1.OR.OP2	(two fields only)

Notes:

1. All field operands may include previously computed derived fields as well as fields from the input tape(s). Derived fields with the same computation type are computed in the order in which they are defined. All code-defined derived fields are defined before all user-defined derived fields. The order of computation for different types can be determined by placing a decimal point to the left of the type code and sorting the resulting fractions in ascending order. Required fields (derived or input) do not need to be explicitly requested, and they are not modified in any way.
2. All functions are applied point-by-point to field operands. Scalar operands are associated with each point of a field operand. At least one operand must always be a field, except for .CONST, which generates a field from a scalar constant.
3. Fields with different numbers of levels may be combined within the same expression. Except when specific levels are referenced, binary functions match levels from the bottom up, and the resulting number of levels is the minimum of the two operands. Extra bottom levels (*XBTL) are not used in any computations; they can only be referenced explicitly as single-level fields. The number of levels resulting from complete expression evaluation must match the number of levels assigned to the derived field by the third value in the group (there are only two possibilities - a single level or all free atmosphere levels).

4. Field values are set to 1. if the result of a logical function is true, and are set to 0. if the result is false. The logical functions :AND and :OR, which require logical field operands, interpret a field value as false if it equals zero, true if it is nonzero.
5. The unary functions .DSWVSUM and .DSTIMES can only be used if the operand is defined on sigma surfaces. Both functions always use the layer midpoint (full level) values for sigma and delta sigma. Even if a subset of sigma levels is being processed <SIGLEVc>, the delta sigma sum over all processed levels is always 1. (The layer interface locations used to compute delta sigma are always midway (in sigma) between the layer midpoint sigma values.)
6. All functions check field operands for blocked points (*DEFS). If a point in either field operand is blocked, the result is also blocked. Scalar operands, however, are not checked for the special blocked point value (1.E36). This means that field values may be set to 1.E36 at a selected set of points, effectively masking those points from subsequent operations. See the example for TSNOW below.
7. The function .ZAVDEV, which computes deviations from a zonal average, computes the zonal average independently of any other zonal averaging operations. The zonal averages (and therefore the deviations), are blocked (*DEFS) if, and only if, all points at a given latitude are blocked.
8. Syntax errors encountered during expression evaluation will cause an error message and an ABORT. Any expression which does not evaluate to a single field is considered invalid. This requirement leads to a few simple rules which can be used to check for some types of expression errors: 1) there must be at least one field operand except when .CONST is referenced, 2) the number of binary functions must be exactly one less than the number of operands, and 3) if there is more than one value preceding the .END, the last value preceding the .END must be a function. For complex expressions, it may be worth generating test fields so that expression evaluations can be verified with known operands. Field values may also be printed (*PRNT), and computations checked by hand.
9. There is no check to make sure operands are within the legal range of applied functions; the FORTRAN library functions provided by the Operating System will generate error messages as appropriate.
10. This ICP is Case independent because derived field definitions are Case independent. Once a derived field is defined, it may be requested for any Case <FIELDcn>, but only if all the fields required to compute it are available for that Case. Ths means that derived field computations cannot combine fields from different Cases. All computation types can be computed for Cases A and B, but Case C does not exist until Case comparison has been done (*ORDR), so only types 61,

62, 71, and 81 can be computed for Case C.

Example:

```
FIELDA1='WIND','TQM','TCELS','FTS','TSNOW','ZC-TV','COSLT','T3','Z5'
 DERFLD=
  'TCELS',61,2,3,0,
      'T',273.15,':MINUS','.END',
  'WIND', 11,2,3,0,
      'U','U',':TIMES','V','V',':TIMES',':PLUS','.SQRT','.END',
  'QM',   11,2,1,0,
      'Q','.DSTIMES',9.80616,':DIVIDE','PS',':L1TIMES','.END',
  'TQM',  11,0,1,0,
      'QM','.VSUM','.END',
  'TQM1', 11,0,1,0,
      'TQM','.END',
  'FTS',  11,0,1,0,
      'TS',280.,':GE','TS',300.,':LE',':AND','.END',
  'TSNOW',11,0,1,0,
      'SNOWH',0.,':EQ',1.E36,':TIMES','TS',':PLUS','.END',
  'ZC-TV',11,2,1,0,
      'T','.ZAVDEV','V','.ZAVDEV',':TIMES','.END',
  'COSLT',11,0,1,0,
     1.0,'.CONST','.LEVEL01','.TCOSL','.END',
  'T3',   41,0,3,0,
      'T','.LEVEL03','.END',
  'Z5',   41,0,3,0,
      'HT0','.LEVEL05','.END'
```

These ICP data cards define eleven new derived fields, and request processing for nine of them. The field TCELS is simply T converted from degrees Kelvin to degree Celsius. It is computed as a multilevel field, after time averaging, on either sigma or pressure surfaces. The field WIND is the velocity magnitude computed from U and V during input from a CCMOB history tape. QM is the water vapor mass in each layer, and TQM is the total water vapor mass in a column. Note that TQM may be requested for processing without explicitly requesting QM, but that the definition of QM must precede the definition of TQM, since they both have the same computation type. TQM1, if requested, would be just a copy of TQM; this is an easy way to rename a field without changing it. The derived field FTS will be 1. if the surface temperature is between 280. and 300., inclusive, and 0. otherwise. Note that averaging FTS (in time or space) results in the fraction (of time or space) for which the expression is true. The next field, TSNOW, is TS masked by the snowcover field. The computation of TSNOW proceeds as follows. SNOWH is the depth of the snowcover, so the result of the logical operation :EQ is true (=1.0) at points where there is no snow, and false (=0.) where there is snow. Multiplying this result by

1.E36 results in a "mask" field which is "blocked" (*DEFS) at points without snow, and zero at all points with snow. Adding TS to this field then gives TSNOW, which is just TS blocked at all points without snowcover. A spatial average of TSNOW is therefore an average of surface temperature over snow only. The next derived field, ZC-TV, when zonally averaged, becomes the zonal covariance of T and V. COSLT is a single-level field containing the cosine of the latitude (same value at all longitudes). Note that .CONST generates a multilevel field, so .LEVEL01 is used to reduce it to a single level before multiplying by the cosine of latitude. T3 is the third level of T, counting only free atmosphere levels, from bottom to top. If vertical interpolation were requested, T3 would contain the temperature on the third interpolated level, since computation of type 41 derived fields follows vertical interpolation. Note that the requested level is part of the function name, and that 2 decimal digits are always required (a leading zero may be necessary). Similarly, Z5 is the fifth level of geopotential height HT0, which is a code-defined derived field computed when reading the input tape(s). Note that HT0 need not be explicitly requested. Forming single level fields in this manner is very useful for computing statistics for mismatched levels. For example, the time covariance of T3 and Z5 could be computed in this step <CVFLDcn>. Note that this requires that T3 and Z5 be explicitly requested on the FIELDA1 card.

->DIFFLDn Array of field name triplets used for requesting Case comparison differences (*COMP). All values are CHARACTER. Within each triplet, the first name is the field from Case A, the second name is the field from Case B, and the third is the difference field (Case A field minus Case B field), which is defined by the user. The fields to be compared must be processed in the corresponding field passes for their respective Cases [FIELDcn]. (up to 100 triplets of names for each keyword, up to 8 CHARACTERs per name, default is no difference fields computed).

Example:
```
 FIELDA1='T','Q'
 FIELDB1='T','MIXRAT'
 DIFFLD1='T','T','T-DIFF','Q','MIXRAT','Q-DIFF'
```

These cards request that differences be computed (Case A minus Case B) for both T and mixing ratio (named Q in Case A, MIXRAT in Case B).

->DPLTCA Specifies the Dicomed camera(s) for the corresponding 'D1' values of <DPLTMF>. Up to 10 CHARACTER values, default is 10:'FICHE'. Specified values are passed to NETDISP with the CAMERA keyword (*PDIS). The Dicomed is currently configured to produce microfiche for a value of 'FICHE', and 35mm roll film for a value

of 'FILM'. There must be a one-to-one correspondence between the values specified for <DPLTCA> and <DPLTMF> up to the last 'D1' value for <DPLTMF>. <DPLTCA> values are ignored unless the corresponding <DPLTMF> value is 'D1'.

Example 1:
```
 DPLTMF='D1','IO','D1'
```
Example 2:
```
 DPLTMF='D1','D1','D1','IO'
 DPLTCA='FICHE','FILM','FICHE'
```

In Example 1, the default camera is used, resulting in two Dicomed fiche copies and one dispose to the IBM 4341. Example 2 requests three Dicomed copies: two on fiche and one on roll film, plus a fourth dispose to the IBM 4341.

->DPLTCT Specifies which Processor Case title <TITLEc> is to be used as the Dicomed plot title if <DPLTIT> is not specified. One value, CHARACTER, 'A', 'B', or 'C', default is 'A'. This ICP is ignored for disposes to <DPLTMF> mainframes other than 'D1'. The maximum length of Dicomed titles depends on the camera type <DPLTCA>; if a title is too long, it is truncated to the maximum length. Titles are prefixed with the string 'ABORT---' if the run aborts due to an error recognized by the Processor code (*PDIS).

->DPLTFN Specifies the filename for the plot file disposed to a mainframe <DPLTMF> other than 'D1'. One value, up to 80 CHARACTERS, default is determined by the mainframe's (Network node's) interface. This filename, especially its length, may be subject to further limitations imposed by the Network node. This ICP is ignored for <DPLTMF> values of 'D1'.

Example 1:
```
 DPLTMF='IO'
 DPLTFN='PLOTS'
```

Example 2:
```
 DPLTMF='AP'
 DPLTFN='Plots.mcd'
```

->DPLTFT Specifies the filetype for the plot file(s) disposed to the IBM 4341. One value, up to 8 CHARACTERS, default is determined by the IBM's Network interface. This ICP is ignored for mainframes <DPLTMF> other than 'IO'.

Example:
```
 DPLTMF='IO'
 DPLTFT='METACODE'
```

->DPLTIT Specifies the Dicomed plot title. One value, up to 72 CHARACTERs, default is to use the title specified by <DPLTCT>. This ICP is ignored for disposes to <DPLTMF> mainframes other than 'D1'. The maximum length of Dicomed titles depends on the camera type <DPLTCA>; if a title is too long, it is truncated to the maximum length. Titles are prefixed with the string 'ABORT---' if the run aborts due to an error recognized by the Processor code (*PDIS).

Example:
DPLTIT='Test Title'

->DPLTMF Specifies the NCAR Local Network node(s) (mainframe(s)) for disposing plot file(s). At plot dispose time, a call to NETDISP is executed for each value specified (*PDIS). Up to 10 CHARACTER values, 2 characters each, default is the single value 'D1'. The following values generate special dispose conditions as noted below. See Computing Division documentation for NETDISP for a complete list of valid mainframe values.

'NO' - no plot file dispose (valid only as the first value specified)
'D1' - Dicomed microfilm output device; the camera type (microfiche or roll film) may be specified with <DPLTCA>, and a plot title may be specified with <DPLTIT> or <DPLTCT>.
'IO' - IBM 4341; the filename and filetype may be specified with <DPLTFN> and <DPLTFT>, respectively.
'AP' - AAP Vax Cluster; the filename (including filetype extension) may be specified with <DPLTFN>. 'WORD=4' is added to the TEXT field.
'nn' - Network Node nn; the filename (including filetype extension) may be specified with <DPLTFN>. MRS=1440 is added to the NETDISP call.

Example:
DPLTMF='D1','IO'

This card requests that two copies of the plot file(s) be disposed: one to the Dicomed fiche unit (default for <DPLTCA>), and one to the IBM 4341. See <DPLTCA> for additional examples.

->ENSMBLc Specifies either a case or time ensemble interpretation of the relationship between the [DAYSc] and [TAPESc] lists. (one value, CHARACTER, default is 'TIME'). Either of the following values may be specified:

'TIME' - the [DAYSc] list is interpreted independently of the [TAPESc] list, i.e., it is assumed that each requested day is unique for the entire [TAPESc] list.

'CASE' - requests case ensemble processing (*ENSB). It is assumed that each requested day [DAYSc] is on each requested tape [TAPESc], and they are all processed. If time averaging is also requested <TIMAVGc>, then the "time average" is actually a case ensemble average.

->FIELDcn List of names for fields to be processed. These fields must be on the input tape(s), or they must be derived fields (*DFLD); other types of computed fields (such as time average statistics (*TAVG)), should NOT be requested via this ICP. (1 to 100 names per keyword, up to 8 CHARACTERs per name).

The Case suffix can be A, B or C; a Case C suffix is used only when requesting derived fields based on differences or ratios (*DFLD). The field pass suffix can range from 1 to 5, and is used to determine which field passes (*DEFS) are processed. If only one field pass is needed, the suffix should be 1. Requesting a field which does not exist results in a printed list of fields on the input tape, followed by an abort.

The number of fields which may be requested and successfully processed in one field pass is limited by the amount of computer memory available. The amount of memory required to process a given list of fields depends not only on the Model resolution, but also on what Processing options have been requested. See the discussion on memory limitations (*MLIM).

Example:
FIELDA1='T','U','V','HTO'

This card requests that T, U and V be read from the input tape(s), and that HTO (geopotential height) be computed and processed as well. The input tapes must be history tapes because HTO was requested.

->FLDSRCc Determines how ambiguities are resolved when a derived field (*DFLD) has the same name as a field on the input tape(s). (one value, CHARACTER, default is 'INPUT'). If a requested field is available from only one source, that source is used regardless of the value of this ICP.

'INPUT' - read ambiguous field from input tape(s)
'DERIVE' - compute the ambiguous field as a derived field

->HEMIS Determines the hemisphere(s) to be plotted when horizontal polar projections are requested <HPROJ>. (one value, CHARACTER, default is 'BOTH'). This ICP is ignored if no polar projections are requested.

'BOTH' - plot both Northern and Southern Hemispheres
'NORTH' - plot Northern Hemisphere only
'SOUTH' - plot Southern Hemisphere only

->HPCDIV Array of triplets used to specify the dividing value(s) for contour lines (*PCON) on horizontal projection plots (*PHOR). The first quantity in each triplet is the field name (CHARACTER), the second quantity is a level specifier (REAL), and the third quantity is the dividing value (REAL). The dividing value is assigned to individual levels (plots) in the same manner as the contour interval for the ICP keyword <HPCINT>. Up to 1000 triplets may be specified (CHARACTER, REAL, REAL). The default is a dividing value of 0. for all field/levels not specified. The ordering of the triplets with respect to field or level specifier has no significance.

Example:
HPCDIV='T',1000.,273.

This data card will cause all contour lines for T values below 273. to be dashed on all horizontal contour plots.

->HPCINT Array of triplets used to specify the contour intervals (*PCON) on horizontal projection plots (*PHOR). The first quantity in each triplet is the field name (CHARACTER), the second quantity is a level specifier (REAL), and the third quantity is the contour interval to be used (REAL). Up to 1000 triplets may be specified (CHARACTER, REAL, REAL); the default is a contour interval of 0., which causes automatic generation of an appropriate interval on a plot by plot basis. Specifying an interval of -1. results in that plot being skipped entirely. The ordering of the triplets with respect to field or level specifier has no significance.

Each triplet may define the contour interval for more than one level of the specified field, since each specified level is interpreted as only one limit of an open-ended vertical range. This is accomplished by applying the following algorithm for each field/level to be plotted. First, the default contour interval of zero is assigned. Then, the level specifier is compared to the value of the vertical coordinate at the level to be contoured (millibars for pressure, 1000*sigma for sigma). If the vertical coordinate value is less than or equal to the level specifier, the contour interval is assigned the third value in the triplet. All specified triplets are checked, however, so that if the vertical coordinate value is less than the level specifier in more than one triplet, the triplet with the smallest level specifier determines the contour interval. Fields defined at only one level have a level value of 1000. (regardless of the actual level placement in the Model), unless the single level is

the result of vertical averaging (*VERA), in which case the level value is always -1. The use of this special value allows plot characteristics to be specified independently for vertical averages; since unaveraged fields always have positive level values, their plots are unaffected.

Example 1:
```
HPCINT='T',1000.,5.,'T',500.,2.
```

This data card will cause all T levels at 500 millibars (or sigma = 0.5) and above to be contoured with an interval of 2., while lower levels (down to 1000.) will be contoured with an interval of 5.

Example 2:
```
HPCINT='U',800.,5.,'U',500.,0.,'U',300.,-1.
```

This data card results in the following contour intervals for U:

* for lev > 800. - contour interval is chosen automatically.
* for 800. >= lev > 500. - contour interval is 5.
* for 500. >= lev > 300. - contour interval is chosen automatically.
* for lev <= 300. - no plots generated.

Example 3:
```
HPCINT='U',100.,5.,'U',-1.,10.
```

This data card results in a contour interval of 5. for all individual (unaveraged) levels of U. If vertical averaging were requested, then the same card would result in a contour interval of 10. for the vertical average of U.

->HPLFPVn Array of triplets (all CHARACTER), listing fields to be plotted as vector arrows on horizontal projections (*PVEC). The first two values are the names (CHARACTER) of the scalar fields to be combined as a vector (eastward and northward components, respectively). The third value is the name of the resulting vector field, as defined by the user. Both of the scalar component fields must be requested for processing in the same field pass [FIELDcn]. (up to 33 triplets for each keyword, up to 8 CHARACTERs per name, default is no vector fields plotted).

Example:
```
FIELDA1='U','V'
HPLFPVn='U','V','WIND'
```

These data cards result in vector arrows for 'WIND', on the horizontal projection requested by <HPVOPT>.

->HPPTVAL Controls point value information on horizontal plots (*PVAL) (one value, CHARACTER, default is 'NO'; 'NO' is always used for polar projection plots)

'NO' - no point value plots are produced
'YES' - point value plots are produced for all horizontal projections; No horizontal projection plots are contoured. Usually only every other grid point value in both dimensions is printed.
'BOTH' - all horizontal projections are plotted as both point values and contours (*PCON), on separate frames.

->HPROJ Projection option for horizontal plots (*PHOR). (one value, CHARACTER, default is 'NONE').

= 'NONE' - no horizontal projection plots are produced
= 'RECT' - an equatorial cylindrical equidistant projection is used for all horizontal plots (single global frame projected as a rectangle).
= 'POLAR' - a polar stereographic projection is used for all horizontal plots. Northern and Southern Hemispheres are plotted on separate frames as circles.

->HPSCAL Array of triplets used to specify the scale factor(s) for horizontal projection (*PHOR) contour plots (*PCON). The first quantity in each triplet is the field name (CHARACTER), the second quantity is a level specifier (REAL), and the third quantity is the scale factor (REAL). Field values are multiplied by the scale factor before being printed on the plot as contour line labels, center values for highs and lows (*PCON), and grid point values (*PVAL). The scale factor (if not equal to 1.) is printed at the bottom of the plot. The scale factor is assigned to individual levels (plots) in the same manner as the contour interval for the ICP keyword <HPCINT>. Up to 1000 triplets may be specified (CHARACTER, REAL, REAL); the default is 0., which causes the scale factor to be determined automatically. The ordering of the triplets with respect to field or level specifier has no significance.

Example:
HPSCAL='Q',1000.,1.E8

This example requests a scale factor of 1.E8 for the horizontal projection plots for all levels of Q.

->HPVDIN Vector density grid increment for all horizontal projection vector plots (*PVEC). This value determines the grid points at which vector arrows are plotted when vector plots are requested <HPLFPVn>. For example, a value of 2 will result in vector arrows at every other grid point in both dimensions (one value, INTEGER, default is 1). Note that this ICP indirectly affects the vector scaling <HPVSCAL> since the scaling is relative to the distance between plotted vectors, not the distance between grid points.

->HPVDIV Array of triplets used to specify the dividing value(s) for vector colors <CLVLTD> <CLVGED> on all horizontal projection vector plots (*PVEC). The first quantity in each triplet is the field name (CHARACTER), the second quantity is a level specifier (REAL), and the third quantity is the dividing value (REAL). The dividing value is assigned to individual levels (plots) in the same manner as the contour interval for the ICP keyword <HPCINT>. Up to 1000 triplets may be specified (CHARACTER, REAL, REAL); the default is 0. for all field/levels not specified. The ordering of the triplets with respect to field or level specifier has no significance.

Example:
```
HPLFPV1='U','V','WIND'
HPVDIV='WIND',1000.,30.
CLVLTD='002'
CLVGED='003'
```

These data cards request that U and V be combined and plotted as the vector 'WIND', and that for all levels of horizontal projections specified by <HPROJ>, 'WIND' vectors whose magnitude is greater than or equal to 30. be colored '003', with all other 'WIND' vectors colored '002'.

->HPVOPT Controls scalar plotting of fields which are requested for plotting as vectors <HPLFPVn>. (One value, CHARACTER, default is 'VECT').

'VECT' - plot only the vector representation of the specified scalar fields. Characteristics of the vector arrows are controlled using the vector name defined by the keyword <HPLFPVn> (*PVEC).

'BOTH' - plot the component fields as scalars (on separate plots) in addition to plotting the vectors. Scalar plotting is still controlled by its own options (*PHOR) (*PCON) (*PVAL).

'VCON' - plot the vector representation of the specified scalar fields, with contours of the vector magnitude superimposed on the same frame. Contour characteristics may be defined (*PCON) using the vector name defined by

the keyword <HPLFPVn>. Characteristics of the vector arrows are controlled as with 'VECT' (*PVEC).

->HPVSCAL Array of triplets used to specify vector scaling on horizontal plots. The first quantity in each triplet is the field name (CHARACTER), the second quantity is a level specifier (REAL), and the third quantity is the vector magnitude scale factor (REAL). The scale factor is assigned to individual levels (plots) in the same manner as the contour interval for the ICP keyword <HPCINT>. Up to 1000 triplets may be specified (CHARACTER, REAL, REAL); the default is 0., which causes the scale factor to be determined automatically. The ordering of the triplets with respect to field or level specifier has no significance.

The lengths of all arrows on a given plot are identically scaled; scaling is NOT a function of latitude, even on polar projection plots. The scale factor (SC) is interpreted as follows:

* SC = 0. - arrow lengths are scaled to vector magnitudes, with the average magnitude assigned an arrow length equal to the distance from one plotted grid point to the next adjacent plotted grid point, at the middle latitude of the plot. Note that this scaling is indirectly affected by the vector density grid point increment <HPVDIN>.
* SC > 0. - arrow lengths are scaled as for SC=0., except that the value of SC is used instead of the average vector magnitude.
* -1. < SC < 0. - all arrows have the same length, which is set to the distance between plot points as for SC=0.
* SC <= -1. - skip the vector plot for the specified field/level.

Example:

```
 HPLFPV1='U','V','WIND'
 HPVSCAL='WIND',800.,-.5,'WIND',500.,20.,'WIND',300.,-1.
```

These data cards request that U and V be combined and plotted as a vector, scaled as follows:

* for lev > 800. - average vector magnitude used to scale the arrow lengths
* 800.>=lev>500. - all arrows are the same length
* 500.>=lev>300. - a vector magnitude of 20. (same units as U and V) is used to scale the arrow lengths
* lev < 300. - no vector plots generated

->ICPECHO Controls the echoing of ICPs to the disposed plot file(s). See (*PDIS). The ICPs are always echoed to the print file ($OUT). One of the following CHARACTER values must be specified; the default is 'BOTH').

'PRINT' - ICPs echoed to print file only
'BOTH' - ICPs echoed to both print and plot files (default)

The plot file text is meaningful only to the Dicomed; it is ignored by the metafile editors on both the AAP VAX Cluster and the IBM 4341s.

->INDEX Controls the printing of an index of plot frames in both the print file ($OUT) and in the disposed plot file(s). The index consists of a one-line description of each plot frame generated and disposed; there is a separate index for each dispose group (*PDIS). Frames are identified by frame numbers consisting of the jobstep number, followed by a period, followed by the plot frame number within the jobstep. These numbers match those generated by <NUMPLT>. One of the following CHARACTER values must be specified; the default is 'BOTH'.

'PRINT' - index is written to print file only
'PLOT' - index is written to plot file only
'BOTH' - index is written to both print and plot files (default)
'NO' - no index written

The plot file text is meaningful only to the Dicomed; it is ignored by the metafile editors on both the AAP VAX Cluster and the IBM 4341s.

->INTDP Default interpolation type code for vertical interpolation to pressure surfaces (*VPRS). One value, INTEGER, default is 2. This interpolation type is used for all fields not named in the exceptions list <PINTXL>. Specifying a value for this ICP nullifies the default for <PINTXL>. See <PINTXL> for examples. The valid interpolation type codes are as follows.

0 - no interpolation
1 - linear in sigma
2 - linear in natural log of sigma (default)
3 - linear in natural log of natural log of (1000*sigma + 2.72)

->LBTDP Default lower boundary treatment code for vertical interpolation to pressure surfaces (*VPRS). One value, INTEGER, default is 4. This lower boundary treatment is used for all fields not named in the exceptions list <PINTXL>. Specifying a value for this ICP nullifies the default for <PINTXL>. See <PINTXL> for examples. The valid lower boundary treatment codes are as follows.

1 - use the surface level if available, extrapolate below the surface if necessary
2 - use the surface level if available, block points below ground elevation
3 - don't use the surface level, extrapolate below the lowest free atmospheric level if necessary
4 - don't use the surface level, block points below the lowest free atmospheric level (default)

->LMLFPSL List of multilevel fields to be plotted as single level line plots one level at a time (*PLIN), in addition to being plotted as a multilevel field on vertical cross sections (*PMEX) (*PLAX) (up to 200 field names, CHARACTER, default is line plots for single level fields only).

->LXASPRT Specifies the aspect ratio for latitudinal cross-section plots (*PLAX). The aspect ratio is defined as the plot height (excluding labels and extra bottom levels, if any) divided by the plot width. See also <LXSIZE> (one value, REAL, default is 0.9).

->LXCDIV Array of value pairs used to specify the dividing value(s) for contour line characteristics (*PCON) for all latitudinal cross-section plots (*PLAX). The first value in each pair is the field name (CHARACTER), and the second is the dividing value (REAL). Up to 200 pairs may be specified; the default is a dividing value of 0.

Example:
LXCDIV='T',273.,'U',20.

This data card requests a dividing value of 273. for T, and a dividing value of 20. for U.

->LXCINT Array of value pairs used to specify the contour interval(s) (*PCON) for all latitudinal cross-section plots (*PLAX). The first value in each pair is the field name (CHARACTER), and the second is the contour interval (REAL). Up to 200 pairs may be specified; the default is 0., which causes the contour interval to be chosen automatically. A negative value causes the plot to be skipped.

Example:
LXCINT='T',5.,'U',3.

This data card requests a contour interval of 5. for T, and a contour interval of 3. for U.

->LXLNSCL Logarithmic scaling option for the ordinate of all latitudinal cross-section plots (*PLAX). (one value, CHARACTER, default is 'YES').

'YES' - The ordinate on latitudinal cross-section plots is logarithmically scaled. Interpolation of contours between data points is linear with respect to the log of the ordinate (*PCON).
'NO' - The ordinate is linearly scaled.
'OLD' - Same as 'YES', except that interpolation of contours between data points is consistent with early versions of the Processor, i.e., logarithmic with respect to the log of the ordinate.

->LXPLOT Controls plotting of latitudinal cross sections (*PLAX) at individual longitudes (one value, CHARACTER, 'YES' or 'NO', default is 'NO'). Meridional averages, when computed (*MERA), are always plotted as latitudinal cross sections; such plots are NOT affected by this ICP. If 'YES' is specified, all fields being processed are plotted at all available latitudes.

->LXPTVAL Controls point value representation (*PVAL) on latitudinal cross-section plots (*PLAX) (one value, CHARACTER, default is 'NO').

'NO' - no point value representations on latitudinal cross sections
'YES' - point value representations are used for all requested latitudinal cross-section plots <MERAVG> <LXPLOT> instead of contours. Usually only every other grid point value in longitude is printed.
'BOTH' - all latitudinal cross sections are plotted with both point values (*PVAL) and contours (*PCON), on separate frames.

->LXSCAL Array of value pairs used to specify the contour plot scale factor(s) (*PCON) for all latitudinal cross-section plots (*PLAX). Field values are multiplied by the scale factor before being printed on the plot as contour line labels, center values for highs and lows (*PCON), and grid point values (*PVAL). The scale factor (if not equal to 1.) is printed at the bottom of the plot. The first value in each pair is the field name (CHARACTER), and the second is the scale factor (REAL). Up to 200 pairs may be specified; the default is 0., which causes an appropriate scale factor to be chosen automatically.

Example:
LXSCAL='T',10.,'Q',1.E8

This data card requests a scale factor of 10. for T and 1.E8 for Q.

->LXSIZE A fraction which specifies the size of latitudinal cross-section plots (*PLAX), relative to the maximum size which will always fit on a frame (vertical extent is variable due to possible extra bottom levels). A value of 1. results in a plot width (not including labels) of .8 times the frame width. This ICP affects both the height and width of the plots proportionately. See also <LXASPRT>. (one value, REAL, default is 1.)

->MAVGPRN Controls option to print all computed meridional averages (*MERA) (one value, CHARACTER, default is 'NO').

'YES' - If meridional averages are computed <MERAVG>, they are printed in the output file as well as being plotted. For each field/level, the average of the meridional averages is also computed and printed.
'NO' - Meridional averages are not printed.

->MASKSc Surface type mask pair used to request surface type masking (*MASK). If this keyword is not specified, there is no general field masking (but there may still be surface type masking associated with other operations). Specifying two CHARACTER values is interpreted as a request to mask all levels of all fields, affecting all subsequent operations (*ORDR). Two values must always be specified, using any combination of the following:

'LAND' - include all points over Model land,
'OCEAN' - include all points over Model ocean,
'SICE' - include all points over Model sea ice. (The distribution of sea ice is currently assumed to be constant in time.)

All masks are "positive", and are combined with a logical OR. For example, specifying the pair 'LAND','SICE' will cause all points above either land or sea ice to retain their field values, while all other points are "blocked". Both mask strings must be specified even if only one mask is desired; for example, the pair 'LAND','LAND' must be used in order to request a land-only mask.

->MAVRNG Range of latitudes for meridional averaging (*MERA), specified as the minimum and maximum value, repectively, in degrees (south latitude is negative, north latitude is positive). Both values are REAL and must be in the range -90. to +90., inclusive. They need not correspond to actual grid point values; all grid points which lie between them (inclusive) are averaged. The default is to average over all latitudes. This ICP is ignored if no meridional averages are requested.

Example:
MERAVG='YES'
MAVRNG=-90.,0.

These two data cards request meridional averages of all southern hemisphere latitudes.

->MBKFR The fraction of points in the averaging interval which must be defined (i.e., not blocked) (*DEFS) before the meridional average (*MERA) of the points is defined (one value, REAL, default is 0.667). Using the default value and an R15 Model resolution, for example, if the average for a particular longitude is not based on at least 27 defined field values (.667*40=26.7), then the average for that longitude is undefined. This ICP is ignored if no meridional averages are requested.

->MEMCON Controls the use of several special memory conservation techniques. See (*CMEM) for a discussion of this parameter. One value, 'YES', or 'NO', default is 'NO'.

->MEMORY Number of words of memory to be pre-allocated to the dynamic memory manager at the beginning of the run (one value, INTEGER, default is no memory preallocated). See (*CMEM) for a discussion of this parameter. A special character value of 'MAX' may also be specified; this will cause all but a small amount of the available memory to be allocated.

->MERAVG Controls the computation of meridional averages (*MERA) (one value, CHARACTER, default is 'NO').

'YES' - Meridional averages are computed and plotted as latitudinal cross sections (*PLAX) for all fields being processed.
'NO' - No meridional averages are computed.

->MNFRMS Minimum number of plot frames produced (per dispose group) (*PDIS), before an attempt to dispose the plot file at a logical breakpoint (e.g., between zonal average and horizontal projection plotting, or between fields within horizontal projection plotting). (one value, INTEGER, default is 10000). This ICP is ignored if <DPLTMF>='NO'

->MSKAP Controls how masked area averages are computed (*HORA). One value, 'YES' or 'NO', default is 'YES'. 'YES' specifies that the field values themselves are to be averaged when masked area averages are requested <MSKFLc>. 'NO' specifies that this type of average is not to be computed (in which case either or both <MSKAPS> and <MSKAZS> would normally be set to 'YES'). This keyword also controls time series plots of masked area averages <TSPALcn>, but in this case only one of the three keywords

<MSKAP>, <MSKAPS>, and <MSKAZS> may be set to 'YES'. If no time series plots are requested, all three keywords may be set to 'YES'.

->MSKAPS Controls how masked area averages are computed (*HORA). One value, 'YES' or 'NO', default is 'NO'. 'YES' specifies that the squares of the field values are to be averaged when masked area averages are requested <MSKFLc>. 'NO' specifies that this type of average is not to be computed (in which case either or both <MSKAP> and <MSKAZS> would normally be set to 'YES'). This keyword also controls time series plots of masked area averages <TSPALcn>, but in this case only one of the three keywords <MSKAP>, <MSKAPS>, and <MSKAZS> may be set to 'YES'. If no time series plots are requested, all three keywords may be set to 'YES'.

->MSKAVPR Controls printing of masked area averages computed when time series plots are produced. One value, 'YES' or 'NO', default is 'NO'. Other masked area averages <MSKFLcn> are always printed.

->MSKAZS Controls how masked area averages are computed (*HORA). One value, 'YES' or 'NO', default is 'NO'. 'YES' specifies that the squares of the zonal averages are to be averaged (meridionally) when masked area averages are requested <MSKFLc>. 'NO' specifies that this type of average is not to be computed (in which case either or both <MSKAP> and <MSKAPS> would normally be set to 'YES'). This keyword also controls time series plots of masked area averages <TSPALcn>, but in this case only one of the three keywords <MSKAP>, <MSKAPS>, and <MSKAZS> may be set to 'YES'. If no time series plots are requested, all three keywords may be set to 'YES'.

->MSKFLcn Array of specification groups for horizontal area averages (*HORA) for a given field, area, and surface type. Each field to be averaged must also be explicitly requested for processing for the corresponding Case and field pass <FIELDcn>, or it must have been generated by a previous processing step (*ORDR). All levels of each field are averaged individually, using the same mask. Each group in the list consists of eight values, as follows:

1) name of input field (CHARACTER),
2) name of output (averaged) field (up to 8 CHARACTERs)
3,4) surface type mask pair; any combination of the following masks as two separate CHARACTER strings:

'LAND' - include all points over Model land,
'OCEAN' - include all points over Model ocean,
'SICE' - include all points over Model sea ice. (The distribution of sea ice is assumed to be constant in time.)

'ALL' - include all points without regard to surface type.

All masks are "positive", and are combined with a logical OR. For example, specifying the pair 'LAND','SICE' will cause all points above either land or sea ice to be averaged within the specified area (see values 5-8 below). Both mask strings must be specified even if only one mask is desired. Note that a pair like 'LAND','ALL' is equivalent to 'ALL','ALL', and results in the averaging of all points within the specified rectangle. The pair 'LAND','LAND', for example, must be used in order to request a land-only mask.

5) latitude defining the southern boundary of the averaging rectangle, specified in degrees, in the range -90. to +90. (REAL)
6) latitude defining the northern boundary of the averaging rectangle, specified in degrees, in the range -90. to +90. (REAL)
7) longitude defining the western boundary of the averaging rectangle, specified in degrees, in the range -180. to the last grid point before (west of) +180. (REAL)
8) longitude defining the eastern boundary of the averaging rectangle, specified in degrees, in the range -180. to the last grid point before (west of) +180. (REAL)

The latitudes and longitudes defining the area do not have to correspond exactly to Model gridpoint values. The actual averaging area consists of a number of complete Model grid boxes whose centers are contained within (or on the perimeter of) the specified rectangle. The western boundary longitude may be greater than the eastern boundary longitude, implying that the desired area includes the +180. to -180. discontinuity (these two values both identify the same meridian). If the two boundary longitudes are equal, the longitude range is 360. degrees, i.e., all longitudes are averaged. Similarly, if the latitude boundaries are equal, all latitudes are averaged. Note that specifying all zeros for the four latitude-longitude parameters is a simple way to request a global average. Gaussian latitude weights are used in order to obtain a true area average.

Up to 200 groups of 8 values each may be specified. The total number of values input must always be a multiple of 8, i.e., each of the 8 values must be specified within each group. If no groups are input, no masked area averages are produced.

Example:

```
 FIELDA1='T'
 MSKFLA1='T','T-NH'  ,'ALL' ,'ALL' ,0.   ,90. ,0. ,0.
        ,'T','T-NA-L','LAND','LAND',15.,75.,-135.,-60.
```

These input cards request two separate masked area averages for T. The first is an area average for the northern hemisphere without any surface type masking; the second average is over the North American continent, land only. The masked-area specification is also printed along with each area average, with the spatial parameters adjusted to reflect the actual locations of the Model grid points used in the average.

->MXASPRT Specifies the aspect ratio for meridional cross-section plots (*PMEX). The aspect ratio is defined as the plot height (excluding labels and extra bottom levels, if any) divided by the plot width. See also <MXSIZE> (one value, REAL, default is 0.9).

->MXCDIV Array of value pairs used to specify the dividing value(s) for contour line characteristics (*PCON) for all meridional cross-section plots (*PLAX). The first value in each pair is the field name (CHARACTER), and the second is the dividing value (REAL). Up to 200 pairs may be specified; the default is a dividing value of 0.

Example:
MXCDIV='T',273.,'U',20.

This data card requests a dividing value of 273. for T, and a dividing value of 20. for U.

->MXCINT Array of value pairs used to specify the contour interval(s) (*PCON) for all meridional cross-section plots (*PMEX). The first value in each pair is the field name (CHARACTER), and the second is the contour interval (REAL). Up to 200 pairs may be specified; the default is 0., which causes the contour interval to be chosen automatically. A negative value causes the plot to be skipped.

Example:
MXCINT='T',5.,'U',3.

This data card requests a contour interval of 5. for T, and a contour interval of 3. for U.

->MXFRMS Maximum number of plot frames produced (per dispose group) (*PDIS), before an attempt to dispose the plot file. (one value, INTEGER, default is 10000). This ICP is ignored if <DPLTMF>='NO'

->MXLATRV Controls latitude reversal on meridional cross-sections (*PMEX) (one value, CHARACTER, default is 'YES')

'YES' - The abscissa on meridional cross sections (including single level line plots) (*PMEX) will have latitudes decreasing from left to right (North Pole on left).
'NO' - Latitudes will increase from left to right (North Pole on right).

->MXLNSCL Logarithmic scaling option for the ordinate of all meridional cross-section plots (*PMEX). (one value, CHARACTER, default is 'YES').

'YES' - The ordinate on meridional cross-section plots is logarithmically scaled. Interpolation of contours between data points is linear with respect to the log of the ordinate (*PCON).
'NO' - The ordinate is linearly scaled.
'OLD' - Same as 'YES', except that interpolation of contours between data points is consistent with early versions of the Processor, i.e., logarithmic with respect to the log of the ordinate.

->MXLTRNG Latitude range to appear on all meridional cross-section plots (*PMEX). Two REAL latitude values, in degrees, southernmost latitude first. Default is -90. to +90. Order of values is independent of <MXLATRV>. The left and right plot borders are placed at the exact range limits, but plotting extends only to the outermost data points within the borders. The latitude labelling increment is chosen as the largest "nice" value which produces between 5 and 10 intervals between labelled tick marks. ("Nice" is defined as 1, 2, 4, or 5 times a power of ten.) Latitude tick marks are always positioned exactly (even if the specified range limits are not integral values), but the tick mark labels are rounded to the nearest degree. A labelled tick mark always appears at the left border of the plot; if the specified range is not a multiple of the computed labelling increment, then the rightmost labelled tick mark will not coincide with the right border. This situation can always be avoided by slight adjustments in the specified range limits.

Example:
MXLTRNG=0.,90.

This data card requests that all meridional cross-section plots cover the northern hemisphere only.

->MXPLOT Controls plotting of meridional cross sections (*PMEX) at individual longitudes (one value, CHARACTER, 'YES' or 'NO', default is 'NO'). Zonal (latitudinal) averages, when computed (*ZONA), are always plotted as meridional cross sections; such plots are NOT affected by this ICP. If 'YES' is specified, all fields being processed are plotted at all available longitudes.

->MXPTVAL Controls point value representation (*PVAL) on meridional cross-section plots (*PMEX) (one value, CHARACTER, default is 'NO').

'NO' - no point value representations on meridional cross sections
'YES' - point value representations are used for all requested meridional cross-section plots <ZONAVG> <MXPLOT> instead of contours. Usually only every other grid point value in latitude is printed.
'BOTH' - all meridional cross sections are plotted with both point values (*PVAL) and contours (*PCON), on separate frames.

->MXSCAL Array of value pairs used to specify the contour plot scale factor(s) (*PCON) for all meridional cross-section plots (*PMEX). Field values are multiplied by the scale factor before being printed on the plot as contour line labels, center values for highs and lows (*PCON), and grid point values (*PVAL). The scale factor is printed at the bottom of the plot if it is not equal to 1. The first value in each pair is the field name (CHARACTER), and the second is the scale factor (REAL). Up to 200 pairs may be specified; the default is 0., which causes an appropriate scale factor to be chosen automatically.

Example:
MXSCAL='T',10.,'Q',1.E8

This data card requests a scale factor of 10. for T and 1.E8 for Q.

->MXSIZE A fraction which specifies the size of meridional cross-section plots (*PMEX), relative to the maximum size which will always fit on a frame (vertical extent is variable due to possible extra bottom levels). A value of 1. results in a plot width (not including labels) of .8 times the frame width. This ICP affects both the height and width of the plots proportionately. See also <MXASPRT>. (one value, REAL, default is 1.)

->NDYHSTc Maximum number of days to place on a single History Save Tape. (One value, INTEGER, default is 30). All Save Tapes in the list <SAVHSTc>, with the possible exception of the last tape generated, will contain this number of days. It is the user's responsibility to limit the amount of data written to a single tape by appropriate specification of this parameter.

->NDYTSRc Maximum number of days to place on a single Time Series Save Tape. (One value, INTEGER, default is 30). All Save Tapes in the list <SAVTSRc>, with the possible exception of the last tape generated, will contain this number of days. It is the user's

responsibility to limit the amount of data written to a single tape by appropriate specification of this parameter.

->NLCDP Default number of Planetary Boundary Layer levels to copy for vertical interpolation to pressure surfaces (*VPRS). One value, INTEGER, default is 0. This number is used for all fields not named in the exceptions list <PINTXL>. Specifying a value for this ICP nullifies the default for <PINTXL>. See <PINTXL> for examples.

->NSDPRNT Number of decimal digits to consider when printing "significant" differences between Cases A and B (*COMP). (One value, INTEGER, default is no printout of significant differences). Only differences requested with the ICP <DIFFLDn> are considered. If the number of decimal digits to consider is specified as "n", then a Case A field is significantly different from the Case B field (at a corresponding point is space) if, and only if, the first "n" digits (excluding leading zeros) are not the same. Printout of significant differences is limited to 10 per latitude, and a total of 50 per field.

->NUMPLT Controls the printing of a frame number at the bottom of each plot frame generated. The frame number consists of the jobstep (*DEFS) number followed by a period, followed by the plot frame count within the jobstep (*PDIS). These numbers match those generated for the plot index <INDEX>. (One value, CHARACTER, 'YES' or 'NO', default is 'YES').

->ORIGFLD Controls processing of original fields (Cases A and B) after Case Comparison (*COMP) (one value, 'YES' or 'NO', CHARACTER, default is 'YES').

'YES' - Processing of the original fields continues after comparison.
'NO' - The Case A and B data are discarded immediately after comparison. No plots are produced for Cases A and B, even if they would otherwise be produced before Case Comparison (*ORDR).

->PINTXL Exceptions list for vertical interpolation to pressure surfaces (*VPRS). The list consists of up to 100 groups of values, with exactly 4 values per group. The values in each group are as follows.

1) field name (up to 8 CHARACTERS)

2) interpolation type code (INTEGER)

    0 - no interpolation
    1 - linear in sigma
    2 - linear in natural log of sigma (default)
    3 - linear in natural log of natural log of (1000*sigma + 2.72)

3) lower boundary treatment code (INTEGER)

    1 - use the surface level if available, extrapolate below the surface if necessary
    2 - use the surface level if available, block points below ground elevation
    3 - don't use the surface level, extrapolate below the lowest free atmospheric level if necessary
    4 - don't use the surface level, block points below the lowest free atmospheric level (default)

4) number of Planetary Boundary Layer levels to copy without interpolating (*VPRS) (INTEGER).

The default exceptions list, which is used if, and only if, all three of the ICPs <INTDP>, <LBTDP>, and <NLCDP> are defaulted, is given below. If a value is input for any of these ICPs, the default exceptions list is null (i.e., there are no exceptions).

```
PINTXL='Z'      ,2,1,0
      ,'HTO'    ,2,1,0
      ,'HT1'    ,2,1,0
      ,'Q'      ,3,4,0
      ,'MIXRAT',3,4,0
      ,'MOIST'  ,0,4,0
      ,'CLD'    ,0,4,0
      ,'CLOUD'  ,0,4,0
      ,'QC'     ,0,4,0
      ,'QRS'    ,0,4,0
      ,'QRL'    ,0,4,0
      ,'DTCONV',0,4,0
```

Example 1:

```
 LBTDP=3
 PINTXL='Z',2,1,0,'MIXRAT',3,4,0
```

In this example, all fields except Z and MIXRAT are assigned an interpolation type of 2, a lower boundary treatment code of 3, and no free atmospheric levels are left uninterpolated. The values assigned with the ICP PINTXL are used for Z and MIXRAT.

Example 2:
INTDP=1

In this example, all fields are assigned an interpolation type of 1, a lower boundary treatment code of 4, and no free atmospheric levels are left uninterpolated. There are no exceptions.

->PKHSTc Packing density for output History Save Tapes (*OHST). Fields may be packed in the same manner as for history tapes output by the Model, except that all fields are packed with the same density. One value, INTEGER, default is 1 (i.e., no packing). Allowable range of values is 1 to 4.

->PRESSLE List of pressure levels for vertical interpolation, (up to 100 levels in millibars, ordered with largest (closest to ground) values first, REAL, default is no interpolation). The specification of at least one value for this keyword is interpreted as a request for interpolation to pressure surface(s).

Example:
PRESSLE=900.,850.,700.,600.,500.,400.,300.,200.,100.,10.

This data card requests that all fields be interpolated to the 10 pressure levels specified.

->PRFLDcn Triplets of field names for time averages of products (*TAVG).

(product = $\overline{xy}$)
The first two names are any requested fields for the same Case and field pass [FIELDcn] (derived fields OK); the third field name is the name by which the time average of the product is to be known. The products are computed only if time averages are also requested for the corresponding Case [TIMAVGc]. The fields needed to compute the products must be explicitly requested via the [FIELDcn] ICP, but the computed product fields should NOT be requested via the [FIELDcn] ICP. (Cases A and B only, up to 100 triplets for each keyword, up to 8 CHARACTERs per name, default is no computation of products).

Example:
FIELDA1='T','U','V'
TIMAVGA='YES'
PRFLDA1='T','U','PRO-TU','T','V','PRO-TV'

These data cards request two time average product fields: the product of T and U, and the product of T and V, both for Case A, field pass 1.

->PRFNMc List of fields whose values are to be printed to the output file (*PRNT). The fields to be printed must be explicitly requested via the [FIELDcn] ICP. This keyword is ignored unless PRINTc = 'YES'. Up to 50 field names, up to 8 CHARACTERs each, default is all fields being processed.

->PRINTc Controls the printing of field values for Case c (*PRNT). One CHARACTER value, 'YES' or 'NO', default is 'NO'. See <PRLATc>, <PRLEVc>, and <PRLONc> before running with this ICP set to 'YES'.

->PRLATc List of latitudes at which field values are to be printed (*PRNT). This keyword is ignored unless PRINTc = 'YES'. Up to 50 REAL values in degrees latitude, in any order. Default is all latitudes being processed. The nearest existing latitude for each specified value is printed.

->PRLIMR Range of field values for limiting printout of field values (*PRNT). If this ICP is input, exactly two REAL values must be specified, and printing is limited to only those field values within a range determined by the input values. If the first value (V1) is less than or equal to the second (V2), then a field value X is printed only if V1 <= X <= V2. If V1 is greater than V2, then X is printed only if X > V1 or X < V2 (think of the print range as wrapping around infinity, plus to minus, from V1 to V2, and excluding the limits). This print limiting is applied to all field values requested by the specification lists, regardless of Case. If no limiting range is specified, there is no print limiting based on field value. This keyword is ignored unless PRINTc = 'YES'.

->PRLEVc List of levels at which field values are to be printed (*PRNT), in millibars for pressure, 1000*SIGMA for sigma surfaces. This keyword is ignored unless PRINTc = 'YES'. Up to 50 REAL values, in any order. Default is all levels being processed. The nearest existing level for each specified value is printed.

->PRLONc List of longitudes at which field values are to be printed (*PRNT). Up to 50 REAL values in degrees longitude, in the range -180. to the last grid point before (west of) +180., in any order. Default is all longitudes being processed. The nearest existing longitude for each specified value is printed.

->PWDHSTc TMS-4 write password, and dedicated TBM reel for all History Save Tapes (*OHST) <SAVHSTc> for the corresponding Case. (One or two values, both CHARACTER). The first value is the write password, which defaults to no password protection. The second value is the dedicated TBM reel name, which defaults to the system default reel. The write password must be specified in order to specify a dedicated reel; 0 (INTEGER) may be used to indicate no password

protection. Specifying a write password of 'NOTBM' will prevent the Save Tape from being disposed to the TBM; it will still be saved on the Cray disk, without a write password. If a Save Tape is being written to a dedicated reel, the generic resource declaration *TD must appear on the Cray JOB card.

Example:
PWDHSTA='PASSWD'

->PWDTSRc TMS-4 write password, and dedicated TBM reel for all Time Series Save Tapes (*STSR) <SAVTSRc> for the corresponding Case. (One or two values, both CHARACTER). The first value is the write password, which defaults to no password protection. The second value is the dedicated TBM reel name, which defaults to the system default reel. The write password must be specified in order to specify a dedicated reel; 0 (INTEGER) may be used to indicate no password protection. Specifying a write password of 'NOTBM' will prevent the Save Tape from being disposed to the TBM; it will still be saved on the Cray disk, without a write password. If a Save Tape is being written to a dedicated reel, the generic resource declaration *TD must appear on the Cray JOB card.

Example:
PWDTSRA='PASSWD'

->PTOPc Pressure (in millibars) at the top of the Model, used for computing the code-defined derived field DELPRES (*CDFL). (One value, REAL, default is undefined; see DELPRES description).

->RATFLDn Array of field name triplets used for requesting Case comparison ratios (*COMP). All values are CHARACTER. Within each triplet, the first name is the field from Case A, the second name is the field from Case B, and the third is the ratio field (Case A field divided by the Case B field), which is defined by the user. The fields to be compared must be processed in the corresponding field passes for their respective Cases [FIELDcn]. (up to 100 triplets of names for each keyword, up to 8 CHARACTERs per name, default is no ratio fields computed).

Example:
```
FIELDA1='T','U'
FIELDB1='T','U'
RATFLD1='T','T','T-RAT','U','U','U-RAT'
```

These cards request that ratios be computed (Case A divided by Case B), for both T and U.

->SAVHSLc Volume name (VSN), write password, and dedicated TBM reel for the Horizontal Slice Save Tape (*SHOR) to be written for the corresponding Case (up to 3 values, all CHARACTER, default is no Horizontal Slice Save Tape written). Specification of one or more values for this keyword is interpreted as a request to produce a Horizontal Slice Save Tape. All fields being processed for the Case are written to the Save Tape (*ORDR). By default, the Save Tape is SAVEd as a permanent dataset on the Cray disk, and also DISPOSEd to the TBM.

Only the first value need be specified, in which case there is no password protection and the system default reel is used. If the second and/or third values are specified, all previous values must also be specified; an INTEGER 0 can be used to indicate that no write password is desired. Specifying a write password of 'NOTBM' will prevent the save tape from being disposed to the TBM; it will still be saved on the Cray disk, without a write password. If a save tape is being written to a dedicated reel, the generic resource declaration *TD must appear on the Cray JOB card.

Example:
```
SAVHSLA='HSLTAP','PASSWD'
```

->SAVHSTc List of volume names (VSNs) comprising a set of History Save Tapes (*OHST) to be written (up to 20 VSNs, 6 CHARACTERs each, default is no History Save Tapes output for the corresponding Case). Specification of one or more values is interpreted as a request to produce the History Save Tape(s); all fields being processed (*ORDR) for the specified Case are saved at each day being processed (but ONLY for FIELD PASS 1). All tapes are SAVEd as permanent datasets on the Cray disk, and also DISPOSEd to the TBM. See <PWDHSTc> for specifying write passwords and TBM reels for these save tapes, and <NDYHSTc> for specifying the maximum number of time samples which may be written to a single tape.

Example:
```
SAVHSTA='HISTP1','HISTP2'
PWDHSTA='PASSWD'
NDYHSTA=15
```

->SAVTAVc Volume name (VSN), write password, and dedicated TBM reel for the Time Average Save Tape (*STAV) to be written for the corresponding Case (up to 3 values, all CHARACTER, default is no Time Average Save Tape produced). Specification of one or more values for this keyword is interpreted as a request to produce a Time Average Save Tape, but only if time average statistics have also been requested [TIMAVGc] for the Case. All fields being processed for the Case are written to the save tape (*ORDR). The Save Tape is SAVEd as a permanent dataset on the Cray disk, and

also DISPOSEd to the TBM.

Only the first value need be specified, in which case there is no password protection and the system default reel is used. If the second and/or third values are specified, all previous values must also be specified; an INTEGER 0 can be used to indicate that no write password is desired. Specifying a write password of 'NOTBM' will prevent the Save Tape from being disposed to the TBM; it will still be saved on the Cray disk, without a write password. If a save tape is being written to a dedicated reel, the generic resource declaration *TD must appear on the Cray JOB card. The keyword [SAVTAPc] is synonymous with [SAVTAVc]; either keyword may be used with the same result.

Example:
SAVTAVA='TAVTAP','PASSWD'

->SAVTSPR Volume name (VSN) for a Time Series Plot Save Tape (produced in a previous run) which is to be read and used as the data source for producing time series plots (*STSP). (one value, CHARACTER, default is no Time Series Plot Save Tape input). If a Time Series Plot Save Tape is specified for input using this keyword, then all of the keywords used to originally request the time series (in the run which wrote the tape) must also be specified with the same request groups. Although the time series data read from this save tape cannot be modified in any way, the plot characteristics can be changed. These characteristics include: ordinate ranges and Case pairings for line plots; contour intervals, scale factors, and dividing values for contour plots; aspect ratios, sizes, and colors for all plots. Except for the values of these plot parameters, all of the time series plot keywords and value groups should be identical to those used in the run which created the save tape. The order of the keywords is not significant, but the order of the groups of values specified for each keyword is significant: this ordering is what determines which plot parameters are used for which time series. ONLY plot characteristic parameters are examined; the parameters specifying the time series data are ignored, and so may be set to any value (but not skipped entirely).

Example:
SAVTSPR='TSPTAP'

->SAVTSPW Volume name (VSN), write password, and dedicated TBM reel for the Time Series Plot Save Tape (*STSP) to be written (up to 3 values, all CHARACTER, default is no Time Series Plot Save Tape produced). The Save Tape is SAVEd as a permanent dataset on the Cray disk, and also DISPOSEd to the TBM. Specification of one or more values for this keyword is interpreted as a request to produce a Time Series Plot Save Tape; all requested time series

(*PTIM) are saved, for all Cases, but ONLY for field pass 1.

Only the first value need be specified, in which case there is no password protection, and the Save Tape is written to the system default reel. If the second and/or third values are specified, all previous values must also be specified (an INTEGER 0 can be used to indicate that no password protection is desired). Specifying a write password of 'NOTBM' will prevent the Save Tape from being disposed to the TBM; it will still be saved on the Cray disk, without a write password. All time series produced for plotting are written to the Save Tape. If a save tape is being written to a dedicated reel, the generic resource declaration *TD must appear on the Cray JOB card.

Example:
SAVTSPW='TSPTAP','PASSWD'

->SAVTSRc List of volume names (VSNs) comprising a set of Time Series Save Tapes (*STSR) to be written (up to 20 VSNs, 6 CHARACTERs each, default is no Time Series Save Tapes for the corresponding Case). Specification of one or more values is interpreted as a request to produce the Time Series Save Tape(s); all fields being processed (*ORDR) for the specified Case are saved at each day being processed (but ONLY for FIELD PASS 1). All tapes are SAVEd as permanent datasets on the Cray disk, and also DISPOSEd to the TBM. See <PWDTSRc> for specifying write passwords and TBM reels for these save tapes, and <NDYTSRc> for specifying the maximum number of time samples which may be written to a single tape.

Example:
SAVTSRA='TSRTP1','TSRTP2'
PWDTSRA='PASSWD'
NDYTSRA=15

->SDFLDcn Pairs of field names for time standard deviation computations (*TAVG).

(standard deviation = SQRT($\overline{x'x'}$))
The first name is any requested field for the same Case and field pass [FIELDcn] (derived fields OK); the second field name in each pair is the name by which the standard deviation is to be known. The standard deviations are computed only if time averages are also requested for the corresponding Case [TIMAVGc]. The fields needed to compute the standard deviations must be explicitly requested via the [FIELDcn] ICP, but the computed standard deviations should NOT be requested via the [FIELDcn] ICP. (Cases A and B only, up to 100 pairs for each keyword, up to 8 CHARACTERs per name, default is no standard deviations computed).

Example:
FIELDA1='T','U'
TIMAVGA='YES'
SDFLDA1='T','SD-T','U','SD-U'

These data cards request the computation of standard deviations for both T and U, for Case A, field pass 1.

->SFCTCRT Volume name (VSN), write password, and dedicated TBM volume for creating (writing) a Surface Type Save Tape (*SFCT) (up to 3 values, all CHARACTER, default is no Surface Type Save Tape written). There are limitations on the form of the input data if such a save tape is to be output; see (*SFCT) for further details. The save tape is SAVEd as a permanent dataset on the Cray disk, and also DISPOSEd to the TBM. Specification of one or more values for this keyword is interpreted as a request to produce a Surface Type Save Tape; the first Case processed (normally Case A) is used as the source for the data.

Only the first value need be specified, in which case there is no password protection, and the save tape is written to the system default reel. If the second and/or third values are specified, all previous values must also be specified (an INTEGER 0 can be used to indicate that no password protection is desired). Specifying a write password of 'NOTBM' will prevent the save tape from being disposed to the TBM; it will still be saved on the Cray disk, without a write password. If a save tape is being written to a dedicated reel, the generic resource declaration *TD must appear on the Cray JOB card.

Example:
SFCTCRT='SFCTAP','PASSWD'

->SFCTTAP Volume name (VSN) for a Surface Type Save Tape to be used as input (*SFCT) (1 value, CHARACTER, default is no Surface Type Save Tape input). This tape must have been created in a previous run using the ICP keyword <SFCTCRT>.

->SHSLZAV Controls whether zonal averages or point values are output to the Horizontal Slice Save Tape when such output is requested <SAVHSLc>. This zonal averaging is independent of that requested with <ZONAVG>, but the same zonal average qualifiers such as longitude range <ZAVRNG> are used. One CHARACTER value, 'YES' or 'NO', default is 'NO'.

'YES' - Only zonal averages are written to the tape.
'NO' - Only point values are written to the tape.

->SIGLEVc List of indices for Model sigma levels to be processed (*LSIG) when input is from Model history tapes (Cases A and B only). The indices refer to Model layer numbers, with a value of one indicating the lowest layer, and indices increasing upward. The list must be ordered with the indices increasing, but they need not be consecutive (i.e., individual layers may be skipped). Indices larger than the number of Model layers are ignored. Up to 100 indices, all INTEGER, default is processing of all Model layers on the history tape.

Example:
```
SIGLEVA=3,4,6
```

This set of values requests that only the third, fourth and sixth sigma levels be processed for Case A.

->SPCBPcn List of four spectral truncation parameters used to request spectral bandpassing (*SBND) for a particular Case (A or B only) and field pass. Also see (*SPEC) and Model documentation for a discussion of the spectral truncation parameters and the spherical harmonic coefficients (SHC). All four values are INTEGER:

1) Smallest value of m for which SHC are to be retained
2) Largest value of m for which SHC are to be retained
3) Smallest value of n for which SHC are to be retained
4) Largest value of n for which SHC are to be retained

Spectral processing must be explicitly requested <SPCcn> in order to enable spectral bandpassing.

Example:
```
SPCA1='YES'
SPCBPA1=5,10,3,12
```

These data cards request spectral bandpass filtering for all Case A fields in pass 1. The spherical harmonic coefficients are set to zero for m<5, n>10, n<3, and n>12. The data are then returned to grid point space and processing continues.

->SPCcn Controls the transformation of grid point data into spherical harmonic coefficients (SHC), and back again into grid point space, for a particular Case (A or B only) and field pass. (one value, CHARACTER, 'YES' or 'NO', default is 'NO'. A value of 'YES' must be specified in order to activate any options for processing in spectral space. Also see (*SPEC) and Model documentation for a discussion of the spectral truncation parameters and the spherical harmonic coefficients (SHC).

->SPCDFcn List of fields to be deleted immediately after spectral graphics are completed. This list should include all fields not needed in grid point space. Cases A and B only, up to 100 field names containing up to 8 CHARACTERs each. Default is no fields deleted.

->SPCEFcn List of fields to be excluded from spectral processing. Cases A and B only, up to 100 field names containing up to 8 CHARACTERs each. Default is no fields excluded, i.e., all fields being processed are transformed into spectral space (if spectral processing is requested).

->SPCINTc List of five spectral truncation parameters used to request spectral interpolation (*SINT) for a particular Case (A or B only), for all field passes. Also see (*SPEC) and Model documentation for a discussion of the spectral truncation parameters and the spherical harmonic coefficients (SHC). All five values are INTEGER:

1) New value for spectral truncation parameter M
2) New value for spectral truncation parameter N
3) New value for spectral truncation parameter K
4) New value for the number of longitude points (including two overlap points) in grid point space. This value should be consistent with the new values for M, N, and K.
5) New value for the number of latitude points in grid point space. This value should be consistent with the new values for M, N, and K.

Example:
```
 SPCA1='YES'
 SPCINTA=15,15,30,50,40
```

These data cards request spectral interpolation to an R15 resolution. The data is returned to grid point space at the new resolution (50 by 40), and processing continues.

->SPCVP Defines fields as vector pairs for the purpose of computing vector-pair derived fields in spectral space (*VDFL). All values are field names (up to 8 CHARACTERs each), organized into 1 to 10 groups with exactly 10 field names in each group. Within each group, the first two names define the gridpoint fields comprising the vector pair, and the following eight names define the associated spectral derived fields. The vector pair fields may be any combination of fields read from the input tape(s), code-defined derived fields (*CDFL), and user-defined derived fields (*UDFL), as long as they are both available during the transformation into spectral space (*ORDR). This ICP only defines the new spectral derived fields; in order for them to be computed, they must be explicitly requested with the appropriate

[FIELDcn] ICP, and the appropriate [SPCcn] ICP must be set to 'YES'. The vector pair need not be explicitly requested with the [FIELDcn] ICP. Each field name specified must be unique within the entire list, and the new derived field names must be different than any other derived field names.

For each vector pair, the 8 derived fields are analogous to the 8 code-defined derived fields related to the horizontal wind velocity (U,V) pair. The 10 names in each group are interpreted as follows. The U,V analog field name is given first for each value, and the computation type is given last in parentheses:

1) U - east-west component of gridpoint vector pair
2) V - north-south component of gridpoint vector pair
3) DIV - divergence of vector pair (type 21)
4) VOR - vorticity of vector pair (type 21)
5) CHI - velocity potential of vector pair (type 211)
6) PSI - streamfunction of vector pair (type 211)
7) UD - east-west gridpoint "velocity" from (3). (type 24)
8) UZ - east-west gridpoint "velocity" from (4). (type 24)
9) VD - north-south gridpoint "velocity" from (3). (type 24)
10) VZ - north-south gridpoint "velocity" from (4). (type 24)

Example:

```
 FIELDA1='TAUX','VOR-TAU','DIV-QFX'
 SPCVP='TAUX','TAUY','DIV-TAU','VOR-TAU','CHI-TAU','PSI-TAU',
                     'UD-TAU','UZ-TAU','VD-TAU','VZ-TAU'
       ,'U*Q','V*Q','DIV-QFX','VOR-QFX','CHI-QFX','PSI-QFX',
                     'UD-QFX','UZ-QFX','VD-QFX','VZ-QFX'
 DERFLD='U*Q',11,2,3,0,'U','Q',':TIMES','.END'
       ,'V*Q',11,2,3,0,'V','Q',':TIMES','.END'
```

This example defines 16 vector-pair derived fields, and two user-defined derived fields (*UDFL). The FIELDA1 ICP requests processing for the input field TAUX, plus two of the vector-pair derived fields. Note that DIV-QFX (divergence of moisture flux) is computed from the user-defined derived fields U*Q and V*Q, which do not need to be explicitly requested - they are automatically computed and then deleted as soon as they are no longer needed.

->SPGYINT Sets the ordinate interval for graphs produced by <SPSNGRF>. The upper limit of the ordinate range is set automatically; the lower limit is the upper limit minus the value specified for this keyword. One value, REAL, default is to determine both limits of the ordinate range automatically.

->SPSNGRF Controls the production of a particular type of graph in spectral space. The graph abscissa is the sum over Fourier wavenumber of the natural log of the SHC times its complex conjugate, and the

ordinate is the natural log of the order of the Legendre polynomial. See (*SGRF). All fields which exist in spectral space are graphed. This keyword is ignored unless spectral processing is requested <SPCc>. One value, 'YES' or 'NO', default is 'NO'.

->SUBPc List of indices for pressure levels to be processed when input is from pressure level history tapes (Cases A and B only). This keyword is analogous to <SIGLEVc> for Model history tapes (*LSIG). The indices refer to layer numbers, with a value of one indicating the lowest layer, and indices increasing upward. The list must be ordered with the indices increasing, but they need not be consecutive (i.e., individual layers may be skipped). Indices larger than the actual number of layers are ignored. Up to 100 indices, all INTEGER, default is processing of all levels on the history tape.

->SURFLEV Controls the processing of surface levels (if they exist) along with the free atmospheric levels of requested fields [FIELDcn], for input from history tapes only. (one value, CHARACTER, 'YES' or 'NO', default is 'NO'). Since the automatic processing of surface levels can only occur if input is from a Model history tape, this ICP is ignored if save tapes (*SAVT) are input. This ICP also affects the computation of derived fields (*DERF) from history tapes (including geopotential height Z).

'YES' - If a requested field has a surface level (same field name with an "S" suffix), then it is processed along with the other levels of the field.
'NO' - Surface levels are not added automatically; they are processed only if explicitly requested by their full name, and are then treated as separate fields.

->TAPESc List of TMS-4 volume names for the input data (Case A or B only) (*DATA), ordered with time increasing (up to 300 VSNs, 6 CHARACTERs each)

->TCFLDcn Triplets of field names for total eddy covariance computations (*TAVG).

(total eddy covariance = $\overline{x*y*}$)
where the stars represent deviations from a zonal average, and the overbar represents a time average.

The first two names are any requested fields for the same Case and field pass [FIELDcn] (derived fields OK); the third field name is the name by which the total eddy covariance is to be known. The covariances are computed only if time averages are also requested for the corresponding Case [TIMAVGc]. The fields needed to compute the covariances must be explicitly requested

via the [FIELDcn] ICP, but the computed covariance fields should NOT be requested via the [FIELDcn] ICP. (Cases A and B only, up to 100 triplets for each keyword, up to 8 CHARACTERs per name, default is no total eddy covariances computed).

Example:
FIELDA1='T','U','V'
TIMAVGA='YES'
TCFLDA1='T','U','ECV-TU','T','V','ECV-TV'

These data cards request two eddy covariances: the combination of T and U, and also the combination of T and V, both for Case A, field pass 1.

->TFWTSc Array of time filtering weights used iff 'FULL' or 'HALF' is specified for <TIMFILc>. See (*TFIL). Up to about 75 weights, all REAL.

Example:
TIMFILA='FULL'
TFWTSA=-1.,1.

These data cards request that all fields be replaced by the time differences for consecutive, processed time samples.

->TIMAVGc Controls the computation of time averages and other statistics (one value, CHARACTER, default is 'NO').

'YES' - the time average of all fields being processed (*ORDR) is computed for all requested days [DAYSc], and the computation of other separately-requested time average statistics is enabled.
'NO' - the model days are processed individually. No time average statistics are computed.
'SPEC' - time averages of the SHC are computed in spectral space, provided that spectral processing is also requested <SPCcn>.

->TIMFILc Controls the time filtering option (*TFIL) (Case A or B only) (one or two values, both CHARACTER with defaults 'NONE')

first value:

'NONE' - no time filtering is performed (default)
'FULL' - a full set of filter weights must be specified for the keyword <TFWTSc>. See (*TFIL) for a description of how the filter is applied.
'HALF' - the first half of a set of symmetric filter weights must be specified for the keyword <TFWTSc>. The second half of the filter is generated by appending the first NADD

weights in reverse order. If n is the number of weights input for TFWTSc, then NADD equals n-1 if n is odd, but equals n if n is even. The resulting set of weights is applied as with the 'FULL' option.

'LOWP' - a low-pass filter with a width of 31 points is used (the keyword <TFWTSc> is ignored). Only frequencies in the range 0.0 to 0.1 days**-1. are passed when a series with a 0.5 day increment is filtered. The internally-generated weights are applied as with the 'FULL' option.
NOTE: For additional details on this plus the two following filters, see Blackmon, M.L., "A Climatological Spectral Study of the 500 mb Geopotential Height of the Northern Hemisphere", J. Atmos. Sci., Vol. 33, No. 8, August, 1976, p. 1609.

'BANP' - a band-pass filter with a width of 31 points is used (the keyword <TFWTSc> is ignored). Only frequencies in the range 0.17 to 0.45 days**-1. are passed when a series with a 0.5 day increment is filtered. The internally-generated weights are applied as with the 'FULL' option.

'HIHP' - a high-pass filter with a width of 31 points is used (the keyword <TFWTSc> is ignored). Only frequencies in the range 0.5 to 1.0 days**-1. are passed when a series with a 0.5 day increment is filtered. The internally-generated weights are applied as with the 'FULL' option.

second value:
(used iff first value is 'FULL' or 'HALF')

'NORM' - each of the weights in the full set is divided by the sum of the full set. Note that the sum of the weights is the filter gain at a frequency of zero. This option is useful primarily for computing running means.

any other character string - the weights are not normalized (default)

Example 1:
```
TIMFILA='LOWP'
```

Example 2:
```
TIMFILA='FULL','NORM'
TFWTSA=1.,2.,1.
```

Example 1 requests a 31-point low-pass filter, and Example 2 requests a 3-point time smoothing.

->TITLEc Case description used to identify a particular Case on all plots and printout. The description can be any sequence of printable characters (up to 64 CHARACTERs, default is blanks).

Example:
TITLEA='PROCESSOR TEST RUN'

->TSLPASP Aspect ratio for all time series line plots (*PTIM) (one value, REAL, default is 1.). The aspect ratio is defined as the plot height divided by the plot width. See also <TSLPSIZ>.

->TSLPSIZ A fraction which specifies the ratio of actual plot size to the largest plot which will fit on a frame, for all time series line plots (*PTIM). Both of the plot dimensions are affected proportionately. See also <TSLPASP>. (one value, REAL, default is 1.)

->TSMCASP Aspect ratio for all meridional average time series contour plots (*PTMC) (one value, REAL, default is 1.). The aspect ratio is defined as the plot height divided by the plot width. See also <TSMCSIZ>.

->TSMCSIZ A fraction which specifies the ratio of actual plot size to the largest plot which will fit on a frame, for all meridional average time series contour plots (*PTMC). Both of the plot dimensions are affected proportionately. See also <TSMCASP>. (one value, REAL, default is 1.)

->TSPALcn Specification list used to request time series line plots of masked area averages for a given field, level, area, and surface type mask (*PTAL). These plots may be requested for Cases A, B, and C, field pass 1 only. The list consists of groups of ten parameters, as follows:

1) name of field to be averaged and plotted (CHARACTER). This field must be explicitly requested via the [FIELDcn] ICP.
2,3) surface type mask pair, specifying any combination of the following masks as two separate CHARACTER strings:

'LAND' - include all points over Model land,
'OCEAN' - include all points over Model ocean,
'SICE' - include all points over Model sea ice. The distribution of sea ice is assumed to be constant in time.
'ALL' - include all points without regard to surface type.

All masks are "positive", and are combined with a logical OR. For example, specifying the pair 'LAND','SICE' will cause all points above either land or sea ice to be averaged within the specified area (see parameters 5-8 below). Both mask strings must be specified even if only one mask is desired.

Note that a pair like 'LAND','ALL' is equivalent to 'ALL','ALL', and results in no masking; the pair 'LAND','LAND' must be used in order to request a land-only mask. See (*HORA) and (*SFCT) for a description of the options for input and output of the field which determines the land-ocean-sea ice distribution.

4) level value for the level to be plotted - millibars for pressure levels, 1000.*sigma for Model sigma levels (REAL). This value need not be exact - the nearest available level is plotted.
5) latitude defining the southern boundary of the averaging rectangle, specified in degrees, in the range -90. to +90. (REAL)
6) latitude defining the northern boundary of the averaging rectangle, specified in degrees, in the range -90. to +90. (REAL)
7) longitude defining the western boundary of the averaging rectangle, specified in degrees, in the range -180. to the last grid point before (west of) +180. (REAL)
8) longitude defining the eastern boundary of the averaging rectangle, specified in degrees, in the range -180. to the last grid point before (west of) +180. (REAL)
9,10) minimum and maximum values, respectively, for the range of the plot's ordinate (REAL). If the minimum is greater than or equal to the maximum, the ordinate range is selected automatically.

The latitudes and longitudes defining the area do not have to correspond exactly to Model gridpoint values. The actual averaging area consists of a number of complete Model grid boxes whose centers are contained within (or on the perimeter of) the specified rectangle. The western boundary longitude may be greater than the eastern boundary longitude, implying that the desired area includes the +180. to -180. discontinuity (these two values both identify the same meridian). If the two boundary longitudes are equal, the longitude range is 360. degrees, i.e., all longitudes are averaged. Similarly, if the latitude boundaries are equal, all latitudes are averaged. Note that specifying all zeros for the four latitude-longitude parameters is a simple way to request a global average. Gaussian latitude weights are used in order to obtain a true area average.

Up to 200 groups of 10 values each may be specified. The total number of values input must always be a multiple of 10, i.e., each of the 10 values must be specified within each group. If no groups are input, no plots of this type are produced. This ICP is completely independent of <MSKFLcn>, but the same averaging algorithm is used.

```
Example:
 FIELDA1='T'
 PRESSLEV=850.
 TSPALA1='T','ALL','ALL',850.,0.,0.,0.,230.,290.,
         'T','LAND','LAND',850.,15.,90.,-170.,-60.,230.,290.
```

These input cards request two separate time series plots for T at 850 millibars. The first plot is a global average without any masking; the second plot is an average over the North American continent (land only). The ordinate for both plots will range from 230. to 290. degrees Kelvin. The masked-area specification will also be printed at the top of each plot, with the spatial parameters adjusted to reflect the actual locations of Model grid points.

->TSPFNPn List of names of Case A fields which, if plotted as time series line plots (*PTIM), are to be paired with Case B fields (up to 50 field names pairs, all CHARACTER). This keyword is valid for field pass 1 only. Paired time series are plotted on the same axes and frame. Only the Case A fields are listed for this keyword; the Case B fields paired with them are those which appear in the same relative group position for the corresponding Case B time series keywords (same field pass and time series plot type). For example, if a Case A area average is requested by the second group of 10 values specified for the <TSPALA1> keyword, then the paired Case B average is the one which is requested by the second group of 10 values assigned to the <TSPALB1> keyword. Such a correspondence mechanism is necessary in order to resolve the ambiguities that arise when more than one time series plot is requested for the same field, within a given plot type (e.g., different areas or masks for the same field). All such plots are paired according to relative group position within each plot type, and for all plot types requested for the fields. The specification parameters within the corresponding groups may be different. The paired plots are produced either instead of or in addition to the individual plots, depending on the value assigned to the keyword <TSPFPH>.

The Case A time series for paired plots is always a solid line. If <DASHLIN> is given a value of 'YES' (the default), then the standard dashed line pattern is used for Case B. If <DASHLIN> is 'NO', then the Case B time series is also solid, and Cases A and B are indistinguishable unless different colors are specified for the two Cases (*PCLR).

If the paired plots have different ordinate range specifications, then an expanded range (determined by overlaying the two ranges) is used.

Contour plots are never paired.

```
Example:
 FIELDA1='T'
 TSPPLA1='T',600.,-80.,-120.,200.,240.,
         'T',500.,-80.,-120.,0.,0.
 FIELDB1='T'
 TSPPLB1='T',600.,-78.,+107.,190.,230.,
         'T',500.,-78.,+107.,0.,0.
 TSPFNP1='T'
```

This set of input cards results in two frames. The first frame plots time vs. 600. level temperature near both Byrd Station and Vostok, Antarctica, with an ordinate range of 190. to 240. The second frame plots time vs. 500. level temperature near both Byrd Station and Vostok, with an ordinate range determined by the minimum and maximum of all 500. level temperatures.

->TSPFPH Specifies how paired fields requested by the <TSPFNPn> keywords are handled (one of the following two values, CHARACTER, default is 'PAIR').

'PAIR' - paired fields are plotted together only
'BOTH' - paired fields are plotted both together and on separate frames

->TSPMCcn Specification list used to request time series contour plots of meridional averages for a given field and level (*PTMC). This is a Hovmoller plot with longitude the abscissa and time the ordinate (increasing downwards). These plots may be requested for Cases A, B, and C, field pass 1 only. The list consists of groups of five parameters, as follows:

1) name of field to be plotted (CHARACTER). This field must be explicitly requested via the [FIELDcn] ICP.
2) level value for the level to be plotted - millibars for pressure levels, 1000.*sigma for Model sigma levels (REAL). This value need not be exact - the nearest available level is plotted.
3) contour interval (REAL). Specifying 0. will cause the contour interval to be chosen automatically.
4) scale factor used for all numbers printed on the plot (highs, lows, and line labels) (REAL). Specifying 0. will cause the scale factor to be chosen automatically.
5) dividing value for contour plot characteristics (REAL) (*PCON).

Up to 200 groups of 5 values each may be specified. The total number of values input must always be a multiple of 5, i.e., each of the 5 values must be specified within each group. If no

groups are input, no plots of this type are produced.

The keywords <MAVRNG> and <MBKFR> are used to determine the latitude range and minimum fraction of unblocked points for the meridional average computations; otherwise this option is completely independent of meridional averaging <MERAVG>.

Example:
```
FIELDA1='V'
TSPMCA1='V',500.,0.,0.,0.
```

These input cards result in a time series plot for the meridional average of south-north wind at (or near) 500 millibars (or .5 sigma). The contour interval and scale factor are chosen automatically, and all negative contour lines are dashed.

->TSPPLcn Specification list used to request time series line plots of point values for a given field, level, latitude, and longitude (*PTFL). These plots may be requested for Cases A, B, and C, field pass 1 only. The list consists of groups of six parameters, as follows:

1) name of field to be plotted (CHARACTER). This field must be explicitly requested via the [FIELDcn] ICP.
2) level value for the level to be plotted - millibars for pressure levels, 1000.*sigma for Model sigma levels (REAL). This value need not be exact - the nearest available level is plotted.
3) latitude to be plotted, in degrees, in the range -90. to +90. (REAL). This value need not be exact - the nearest available latitude is plotted.
4) longitude to be plotted, in degrees, in the range -180. to the last grid point before (west of) +180. (REAL). This value need not be exact - the nearest available longitude is plotted.
5,6) minimum and maximum values, respectively, for the range of the plot's ordinate (REAL). If the minimum is greater than or equal to the maximum, the ordinate range is selected automatically.

Up to 200 groups of 6 values each may be specified. The total number of values input must always be a multiple of 6, i.e., each of the 6 values must be specified within each group. If no groups are input, no plots of this type are produced.

Example:
```
FIELDA1='Z'
TSPPLA1='Z',500.,82.,-62.,0.,0.
```

These input cards request a time series plot for the 500. millibar (or .5 sigma) geopotential height near Alert, NWT, Canada. The ordinate scale is automatically chosen so that the full range of values is plotted with maximum resolution.

->TSPZCcn Specification list used to request time series contour plots of zonal averages for a given field and level (*PTZC). Time is the abscissa and latitude the ordinate. These plots may be requested for Cases A, B, and C, field pass 1 only. The list consists of groups of five parameters, as follows:

1) name of field to be plotted (CHARACTER). This field must be explicitly requested via the [FIELDcn] ICP.
2) level value for the level to be plotted - millibars for pressure levels, 1000.*sigma for Model sigma levels (REAL). This value need not be exact - the nearest available level is plotted.
3) contour interval (REAL). Specifying 0. will cause the contour interval to be chosen automatically.
4) scale factor used for all numbers printed on the plot (highs, lows, and line labels) (REAL). Specifying 0. will cause the scale factor to be chosen automatically.
5) dividing value for contour plot characteristics (REAL) (*PCON)

Up to 200 groups of 5 values each may be specified. The total number of values input must always be a multiple of 5, i.e., each of the 5 values must be specified within each group. If no groups are input, no plots of this type are produced.

The keywords <ZAVRNG> and <ZBKFR> are used to determine the longitude range and minimum fraction of unblocked points for the zonal average computations; otherwise this option is completely independent of zonal averaging <ZONAVG>.

Example:
```
 FIELDA1='V'
 TSPZCA1='V',500.,0.,0.,0.
```

These input cards result in a time series plot for the zonal average of south-north wind at (or near) 500 millibars (or .5 sigma). The contour interval and scale factor are chosen automatically, and all negative contour lines are dashed.

->TSPZLcn Specification list used to request time series line plots of zonal averages for a given field, level, and latitude (*PTZL). These plots may be requested for Cases A, B, and C, field pass 1 only. The list consists of groups of five parameters, as follows:

1) name of field to be plotted (CHARACTER). This field must be explicitly requested via the [FIELDcn] ICP.
2) level value for the level to be plotted - millibars for pressure levels, 1000.*sigma for Model sigma levels (REAL). This value need not be exact - the nearest available level is plotted.
3) latitude to be plotted, in degrees, in the range -90. to +90. (REAL). This value need not be exact - the nearest available latitude is plotted.
4,5) minimum and maximum values, respectively, for the range of the plot's ordinate (REAL). If the minimum is greater than or equal to the maximum, the ordinate range is selected automatically.

Up to 200 groups of 5 values each may be specified. The total number of values input must always be a multiple of 5, i.e., each of the 5 values must be specified within each group. If no groups are input, no plots of this type are produced.

The keywords <ZAVRNG> and <ZBKFR> are used to determine the longitude range and minimum fraction of unblocked points for the zonal average computations; otherwise this option is completely independent of zonal averaging <ZONAVG>.

Example:
```
 FIELDA1='U','V'
 TSPZLA1='U',500.,45.,0.,0.,
         'V',500.,45.,0.,0.
```

These input cards result in a time series plot for the zonal average of east-west wind at (or near) 500 millibars (or .5 sigma), plus a similar plot for south-north wind. The ordinate scales are automatically chosen so that the full value ranges are plotted with maximum resolution.

->TSZCASP Aspect ratio for all zonal average time series contour plots (*PTZC) (one value, REAL, default is 1.). The aspect ratio is defined as the plot height divided by the plot width. See also <TSZCSIZ>.

->TSZCSIZ A fraction which specifies the ratio of actual plot size to the largest plot which will fit on a frame, for all zonal average time series contour plots (*PTZC). Both of the plot dimensions are affected proportionately. See also <TSZCASP>. (one value, REAL, default is 1.)

->TYPEc Type of tape(s) input for the corresponding [TAPESc] list (Case A or B only) (one value, CHARACTER, default is 'CCMOB').

='CCMOA' - Model CCMOA format history tape(s)
='CCMOB' - Model CCMOB format history tape(s)
='PTP1' - Pressure level history tape(s)
='SAVTAV' - Processor Time Average Save Tape(s). The value 'SAVE' is synonymous. When this type of save tape is input, the <DAYSc> ICP is ignored.
='SAVTSR' - Processor Time Series Save Tape(s)

NOTE: Both "CCMOA" and "CCMOB" contain the numeral "0", NOT the letter "O".

When reading history format Save Tapes written by the Processor (*OHST), either 'CCMOB' or 'PTP1' should be specified, depending on whether the Save Tape data are on sigma or pressure surfaces, respectively.

->VAVRNG Range of levels to be averaged when vertical averaging is requested (*VERA) <VERAVG>, specified as the lowest (closest to the ground) and highest levels, respectively, in either millibars or 1000 times the sigma value. All levels which lie within the specified range (inclusive) are averaged. The specified levels need not correspond to actual level values, but a full layer weight is always used (*VERA). (two values, real, default is to average all levels). This ICP is ignored if no vertical averages are requested.

Example:
VAVRNG=1000.,500.

->VBKFR The fraction of points in the averaging interval which must be defined (i.e., not blocked) before the vertical average (*VERA) of the points is defined (one value, REAL, default is 0.0). If any average is based on this fraction of points or less, that average is blocked. This ICP is ignored if no vertical averages are requested.

->VERAVG Controls the computation of vertical averages (*VERA) (one value, CHARACTER, default is 'NO').

'YES' - All fields being processed are vertically averaged.
'NO' - The fields are not vertically averaged.

->ZAVGPRN Controls printing of zonal averages (*ZONA) requested with the ICP <ZONAVG> (one value, CHARACTER, default is 'NO').

'YES' - If zonal averages are computed <ZONAVG> then the averages are printed in the output file as well as being plotted.
'NO' - Zonal averages are not printed.

->ZAVRNG Range of longitudes to be averaged when zonal averaging is requested (*ZONA), specified as the westernmost and easternmost longitudes, repectively, in degrees (west longitude is negative, east longitude is positive, valid range of longitudes is -180. to the last grid point before (west of) +180.) The specified longitudes need not correspond to actual grid point values; all grid points which lie between the specified limits (inclusive) are averaged. The western boundary longitude may be greater than the eastern boundary longitude, implying that the desired range includes the +180. to -180. discontinuity (these two values both identify the same meridian). If the two boundary longitudes are equal, the longitude range is 360. degrees, i.e., all longitudes are averaged. This keyword is ignored if no zonal averages are requested. (Two values, REAL, default is to average all longitudes).

Example:
ZAVRNG=-130.,-60.

->ZBKFR The fraction of points in the averaging interval which must be defined (i.e., not blocked) (*DEFS) before the zonal average (*ZONA) of the points is defined (one value, REAL, default is 0.667). If the average for any latitude is based on this fraction of points or less, that average is blocked. This ICP is ignored if no zonal averages are requested.

->ZCVFLcn Triplets of field names used to request time average zonal covariance computations (*ZEST).

$$(\text{zonal covariance}) = \left[\overline{(x)^{*}\,(y)^{*}}\right]$$

The first two names are any processed fields for the same Case and field pass (*ORDR); the third field name is the name by which the covariance is to be known. The covariances are computed only if time averages are also requested for the corresponding Case [TIMAVGc]. The fields needed to compute the covariances must be explicitly requested via the [FIELDcn] ICP, but the computed covariance fields should NOT be requested via the [FIELDcn] ICP. (Cases A and B only, up to 100 triplets for each keyword, up to 8 CHARACTERs per name, default is no zonal covariances computed). Case B is temporarily limited to field pass 1.

Example:
FIELDA1='T','U','V'
TIMAVGA='YES'
ZCVFLA1='T','U','ZCV-TU','T','V','ZCV-TV'

These data cards request two zonal covariance fields: the covariance of T and U, and the covariance of T and V, both for Case A, field pass 1.

->ZONAVG Controls the computation of zonal averages (*ZONA) (one value, CHARACTER, default is 'NO').

'YES' - Zonal averages are computed and plotted as meridional cross-sections (*PMEX) for all fields in all Cases.
'NO' - No zonal averages are computed.

->ZORD Controls the addition of a height scale (in km) along the ordinate (outside right) of all vertical cross-section plots (*PMEX) (*PLAX). (one value, CHARACTER, 'YES' or 'NO', default is 'YES'). A 7 km scale height is used to relate height to the plot ordinate, and a surface pressure of 1013.2 millibars is assumed when the ordinate is pressure.

->ZSBKFR The fraction of points in the averaging interval which must be defined (i.e., not blocked) (*DEFS) before zonal average statistics (*ZEST) are defined (one value, REAL, default is 0.667). If statistics for any latitude are based on this fraction of points or less, the statistics are blocked. This ICP is ignored if no zonal average statistics are requested.

->ZSTFLcn Pairs of field names used to request time average zonal standard deviation computations (*ZEST).

$$\text{zonal standard deviation} = \text{SQRT}[\overline{(x)^* (x)^*}]$$

The first name is any processed field for the Case and field pass (*ORDR); the second field name is the name by which the zonal standard deviation is to be known. The standard deviations are computed only if time averages are also requested for the corresponding Case [TIMAVGc]. The fields needed to compute the standard deviations must be explicitly requested via the [FIELDcn] ICP, but the computed standard deviation fields should NOT be requested via the [FIELDcn] ICP. (Cases A and B only, up to 100 pairs for each keyword, up to 8 CHARACTERs per name, default is no zonal standard deviations computed). Case B is temporarily limited to field pass 1.

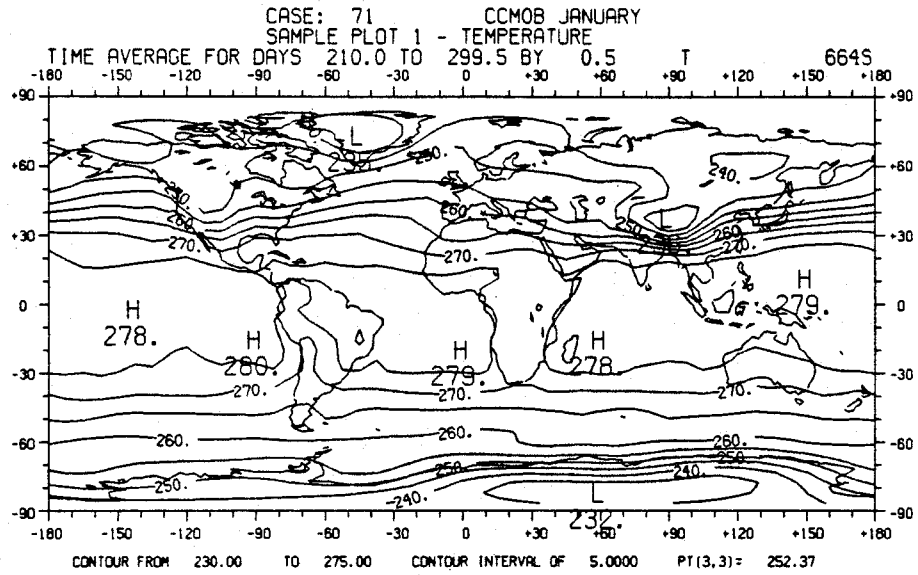
Example:
```
FIELDA1='T','U'
TIMAVGA='YES'
ZCVFLA1='T','ZSD-T','V','ZSD-V'
```

These data cards request the zonal standard deviations for both T and U for Case A, field pass 1.

<-> end of Part 3. List of ICP Keywords

Appendix A

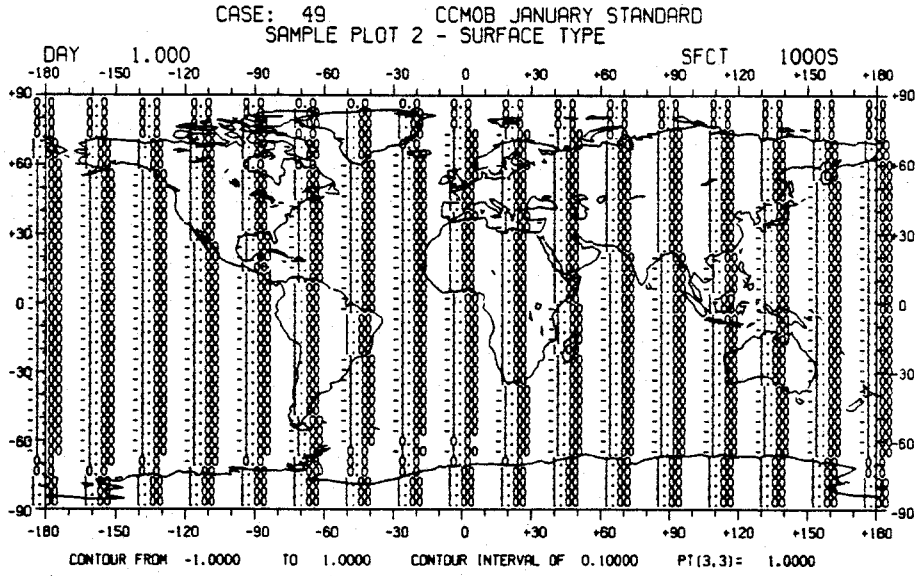
Sample Plots



FRAME 1.5

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 1 - TEMPERATURE'
 TAPESA='S71Z02'
 TYPEA='SAVTAV'
 FIELDA1='T'
 HPROJ='RECT'
 HPCINT='T',1000.,5.
 ENDOFDATA
```

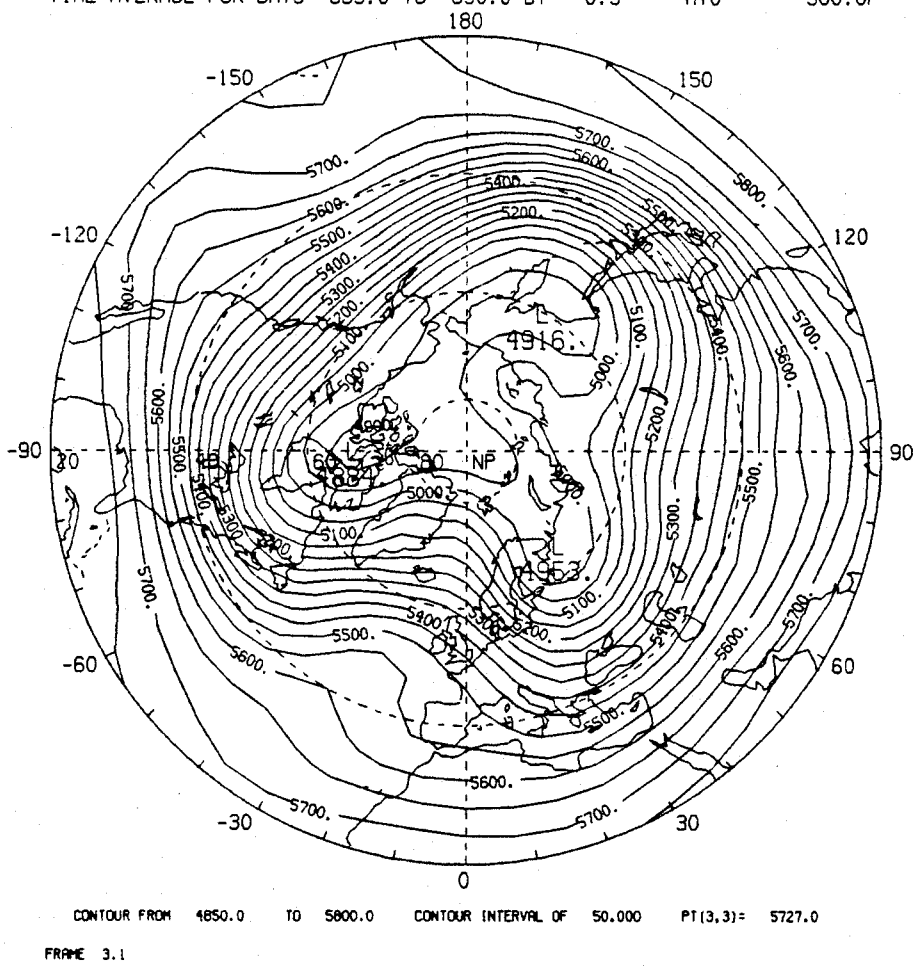
Fig. 1. Temperature contours on a sigma surface (*PHRE) (*PCON). A separate plot is generated for each sigma level on the input Time Average Save Tape (*STAV).



FRAME 2.1

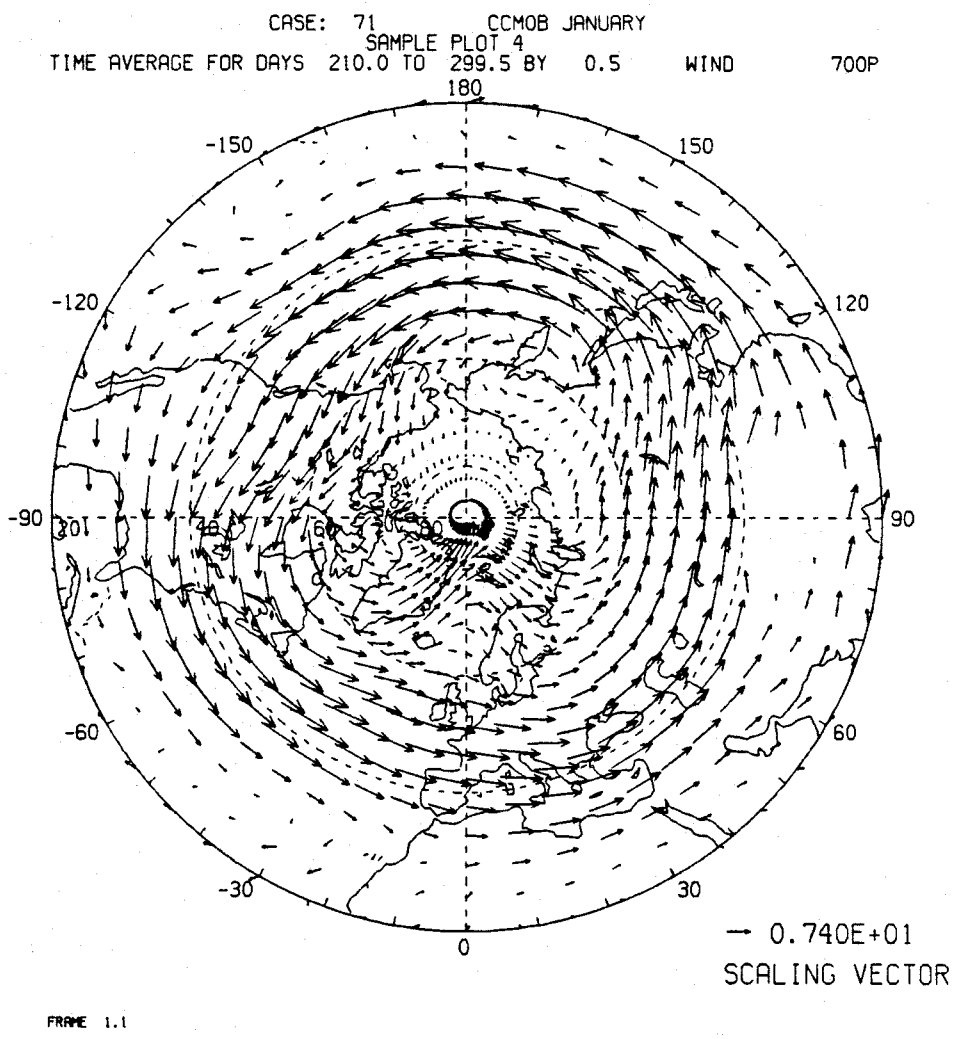
```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 2 - SURFACE TYPE'
 TAPESA='SR15JA'
 TYPEA='SAVTAV'
 FIELDA1='SFCT'
 HPROJ='RECT'
 HPPTVAL='YES'
 ENDOFDATA
```

Fig. 2. Point values for surface type (land, ocean, sea ice) (*PHRE) (*PVAL). Note that the Surface Type Save Tape SR15JA (*SFCT) is processed as if it were a Time Average Save Tape (*ITAV).



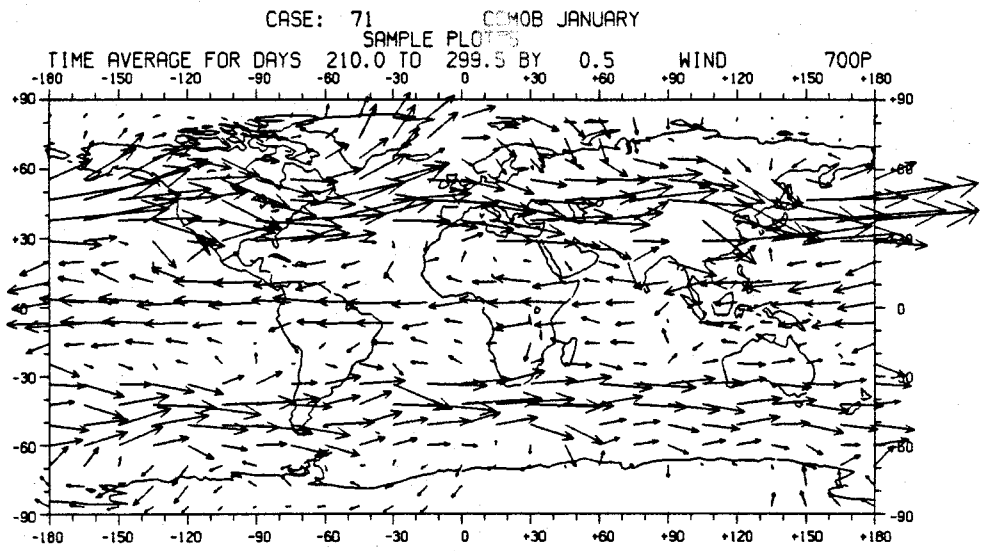

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 3 - GEOPOTENTIAL HEIGHT'
 TAPESA='X71058','X71059','X71060'
 DAYSA=865.,890.,0.5
 FIELDA1='HTO'
 PRESSLE=500.
 TIMAVGA='YES'
 HPROJ='POLAR'
 HEMIS='NORTH'
 ENDOFDATA
```

Fig. 3. Geopotential height at 500 millibars contoured on a polar projection (*PHPO) (*PCON). The derived field HTO (*CDFL) is computed (on sigma surfaces) from fields on the input history tape, then interpolated to 500 millibars (*VPRS).



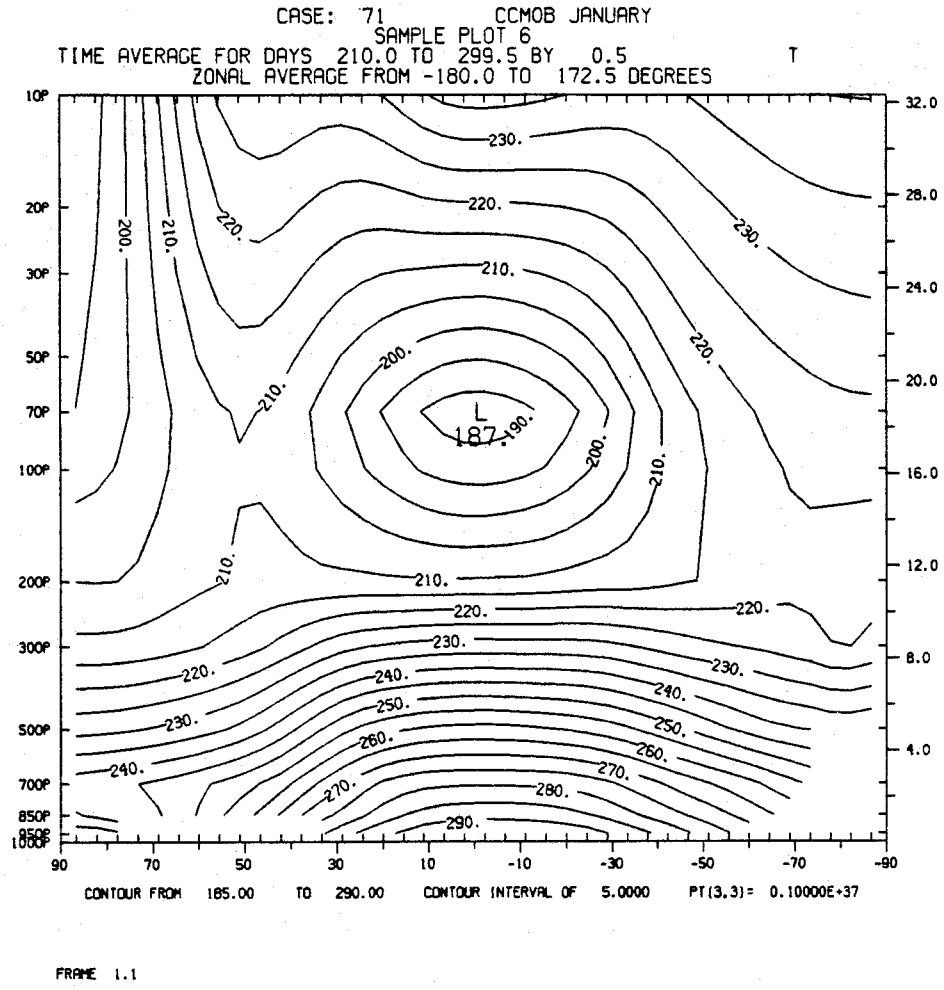

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 4'
 TAPESA='S71P02'
 TYPEA='SAVTAV'
 FIELDA1='U','V'
 HPROJ='POLAR'
 HEMIS='NORTH'
 HPLFPV1='U','V','WIND'
 HPVOPT='VECT'
 HPVSCAL='WIND',1000.,-1.,'WIND',700.,0.,'WIND',699.,-1.
 ENDOFDATA
```

Fig. 4. Horizontal wind vectors at 700 millibars on a polar projection (*PHPO) (*PVEC). The input Time Average Save Tape contains data on a number of pressure levels (*STAV), but plotting for levels other than 700 millibars is supressed by the ICP HPVSCAL.



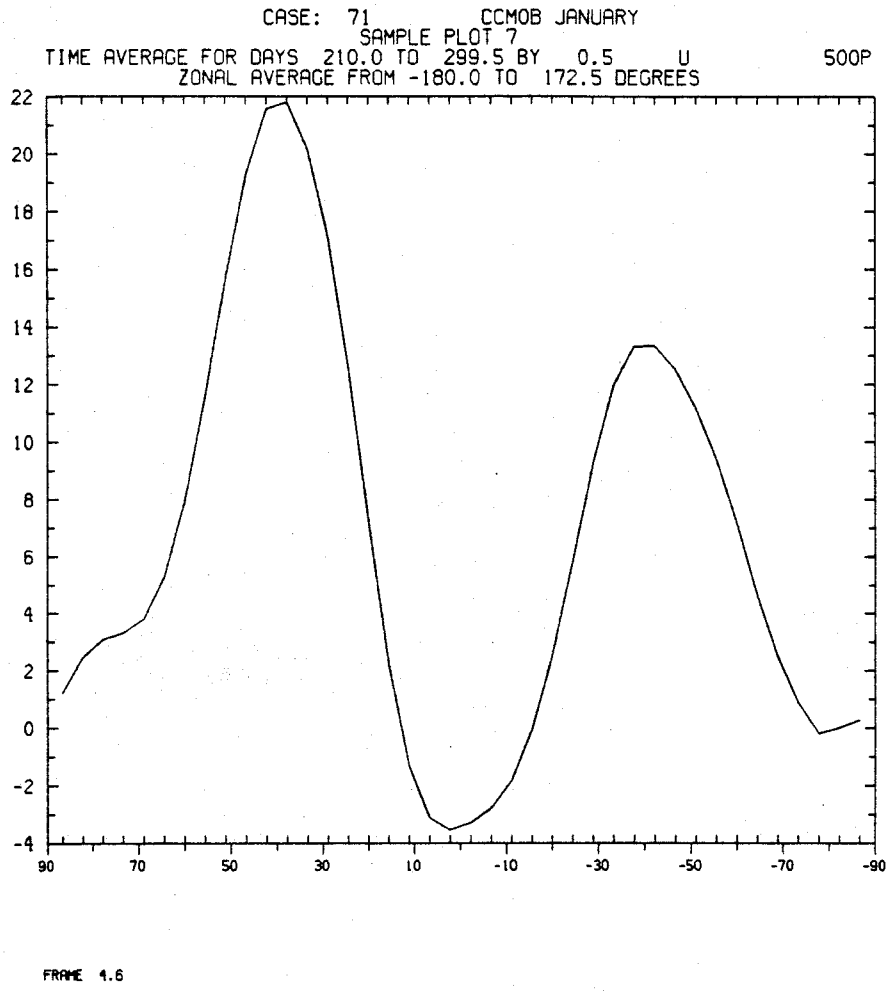

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 5'
 TAPESA='S71P02'
 TYPEA='SAVTAV'
 FIELDA1='U','V'
 HPROJ='RECT'
 HPLFPV1='U','V','WIND'
 HPVOPT='VECT'
 HPVDIN=2
 HPVSCAL='WIND',1000.,-1.,'WIND',700.,0.0,'WIND',699.,-1.
 HPCINT='WIND',1000.,5.
 ENDOFDATA
```

Fig. 5. Horizontal wind vectors at 700 millibars on a rectangular projection (*PHRE) (*PVEC). The input Time Average Save Tape contains data on a number of pressure levels (*STAV), but plotting for levels other than 700 millibars is supressed by the ICP HPVSCAL.



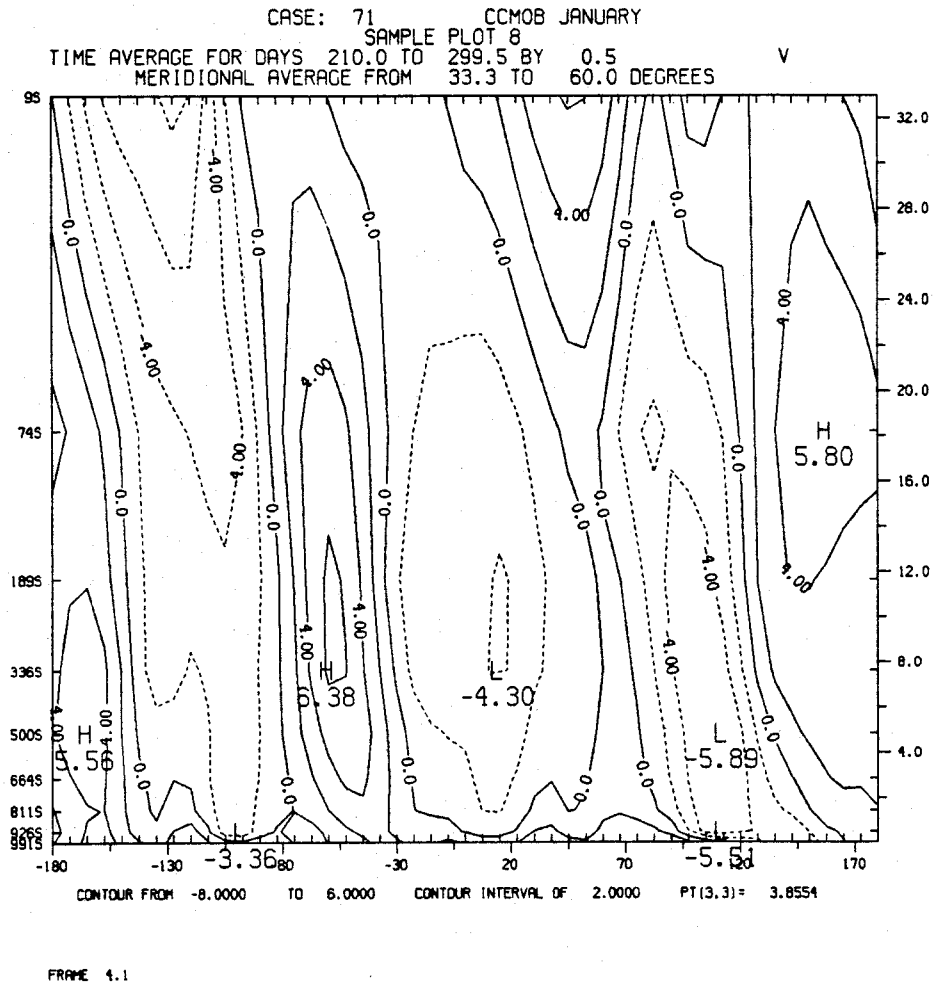

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 6'
 TAPESA='S71P02'
 TYPEA='SAVTAV'
 FIELDA1='T'
 ZONAVG='YES'
 MXCINT='T',5.
 ENDOFDATA
```

Fig. 6. Zonally averaged temperature contoured as a meridional cross section (*ZONA) (*PMEX) (*PCON). All pressure levels contained on the input Time Average Save Tape (*STAV) are contoured.



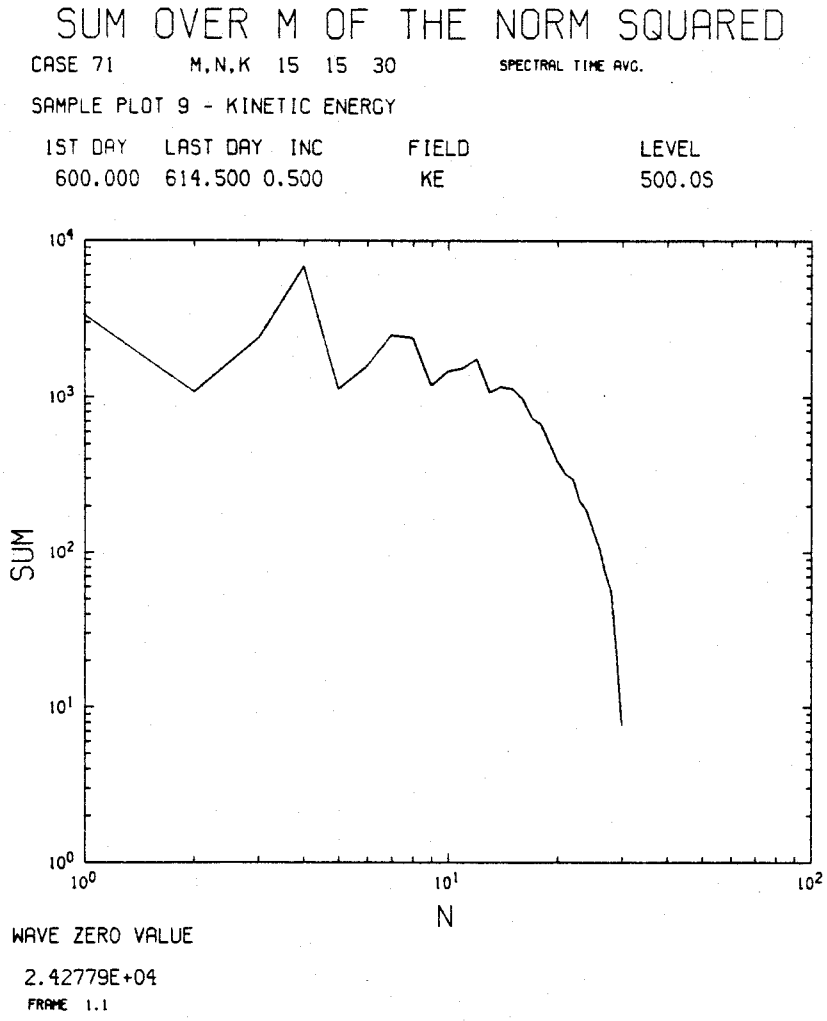

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 7'
 TAPESA='S71P02'
 TYPEA='SAVTAV'
 FIELDA1='U'
 ZONAVG='YES'
 LMLFPSL='U'
 ENDOFDATA
```

Fig. 7. Line plot for zonally averaged zonal wind at 500 millibars (*ZONA) (*PLIN). A separate plot is produced for each pressure level contained on the input Time Average Save Tape (*STAV).



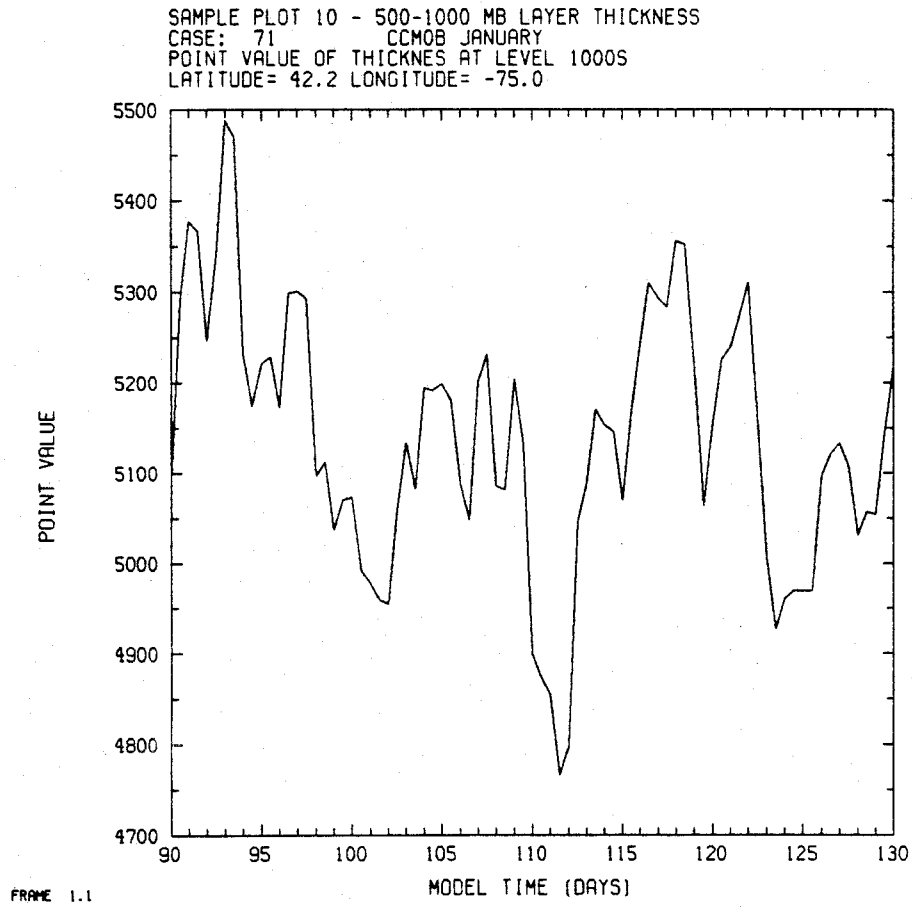

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 8'
 TAPESA='S71ZO2'
 TYPEA='SAVTAV'
 FIELDA1='V'
 MERAVG='YES'
 MAVRNG=30.,60.
 LXCINT='V',2.
 ENDOFDATA
```

Fig. 8. Meridionally averaged meridional wind contoured as a latitudinal cross section (*MERA) (*PLAX) (*PCON). All sigma levels contained on the input Time Average Save Tape (*STAV) are contoured.



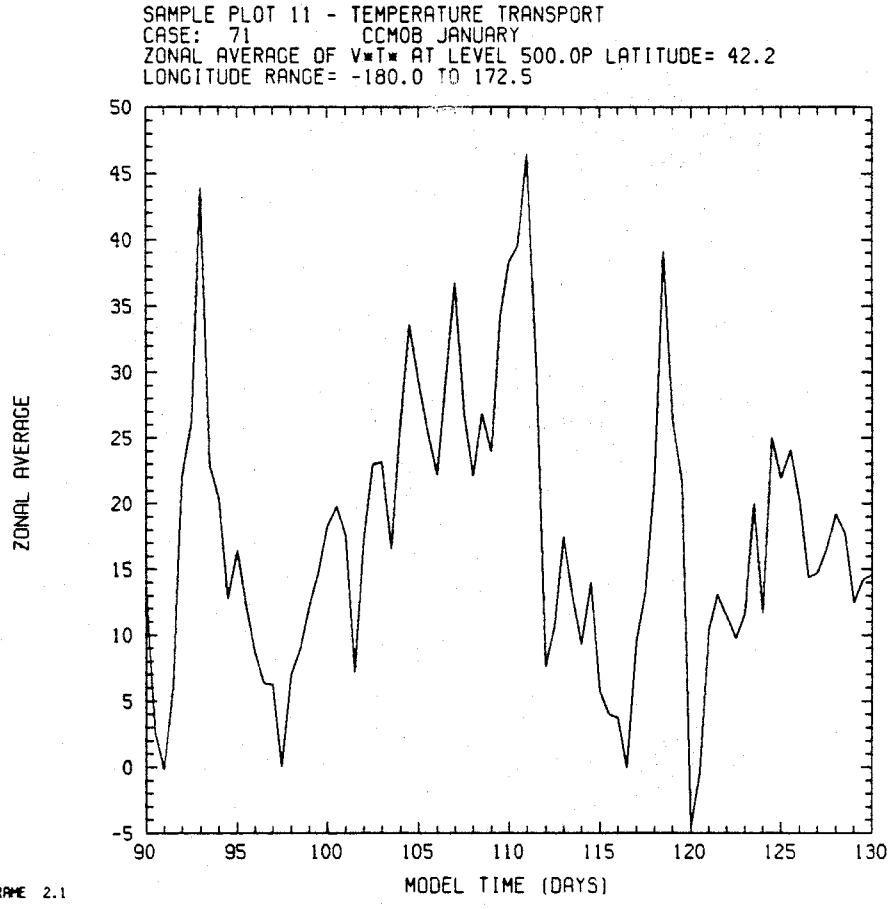

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 9 - KINETIC ENERGY'
 TAPESA='X71041'
 DAYSA=600.,614.5,0.5
 FIELDA1='KE'
 SIGLEVA=5
 SPCA1='YES'
 TIMAVG='SPEC'
 SPSNGRF='YES'
 ENDOFDATA
```

Fig. 9. Spectral plot of kinetic energy, time averaged in spectral space (*SPGR). Kinetic energy is computed from U and V on the input history tape (*CDFL).



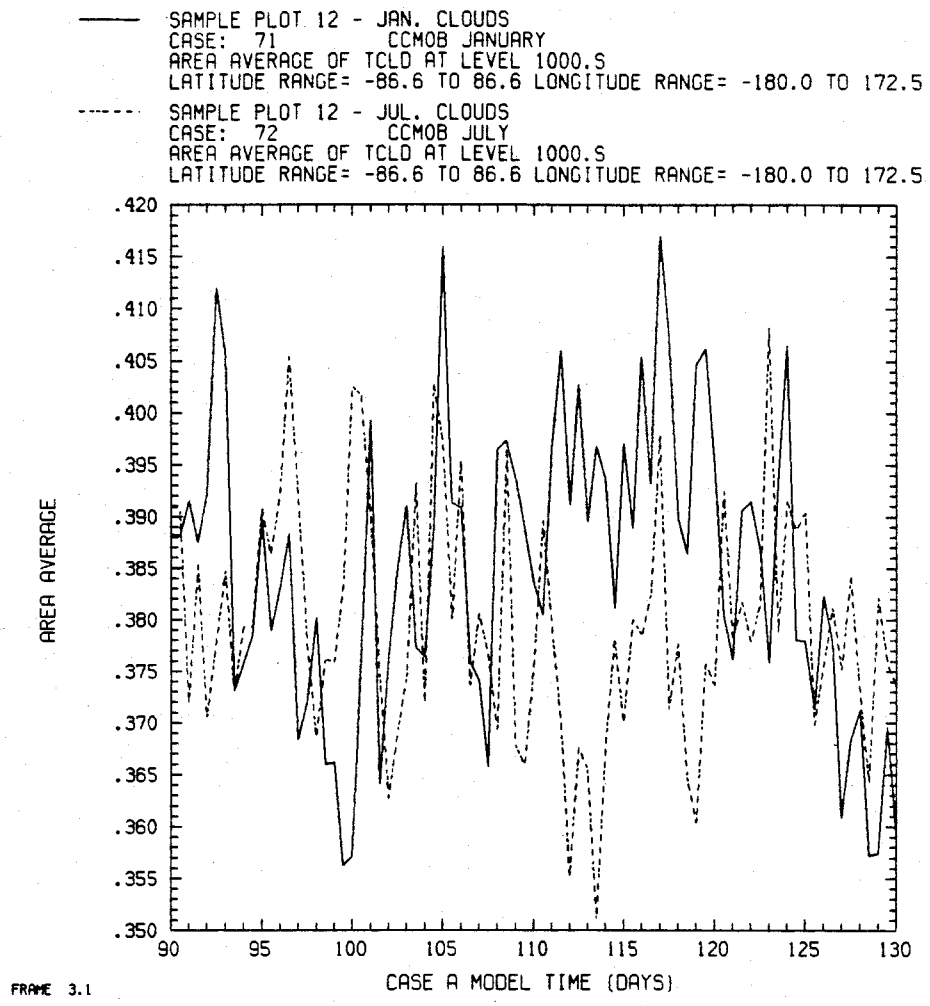

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 10 - 500-1000 MB LAYER THICKNESS'
 TAPESA='X71007','X71008','X71009'
 DAYSA=90.,130.,0.5
 FIELDA1='THICKNES'
 PRESSLE=1000.,500.
 DERFLD='THICKNES',41,0,2,0,
        'HT0','.LEVEL02','HT0','.LEVEL01',':MINUS','.END'
 TSPPLA1='THICKNES',500.,40.,-75.,2:0.
 ENDOFDATA
```

Fig. 10. Thickness of 500-1000 millibar layer at (42.2N, 75.W), plotted as a time series (*PTFL). The geopotential height is computed from fields on the input history tapes (*CDFL), and then interpolated to 1000 and 500 millibars (*VPRS). The algebraic expression for computing thickness from geopotential height is specified with the ICP DERFLD (*UDFL).



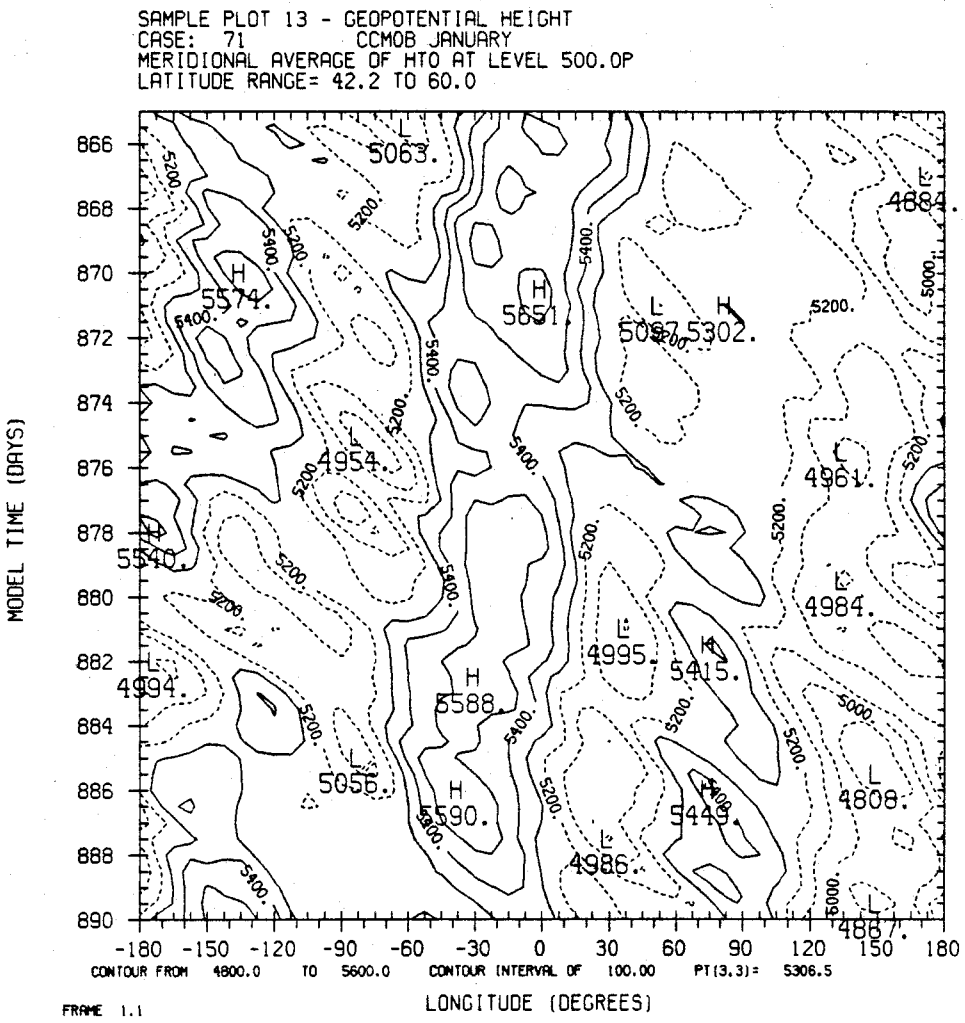

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 11 - TEMPERATURE TRANSPORT'
 TAPESA='X71007','X71008','X71009'
 DAYSA=90.,130.,0.5
 PRESSLE=500.
 FIELDA1='V*T*'
 DERFLD='V*T*',11,2,3,0,
        'V','.ZAVDEV','T','.ZAVDEV',':TIMES','.END'
 TSPZLA1='V*T*',500.,40.,2:0.
 ENDOFDATA
```

Fig. 11. Zonal average of 500 millibar temperature transport across 42.2N, plotted as a time series (*PTZL). V and T are interpolated to 500 millibars immediately after being read from the input history tapes (*VPRS). The algebraic expression for computing V*T* from V and T is specified with the ICP DERFLD (*UDFL).



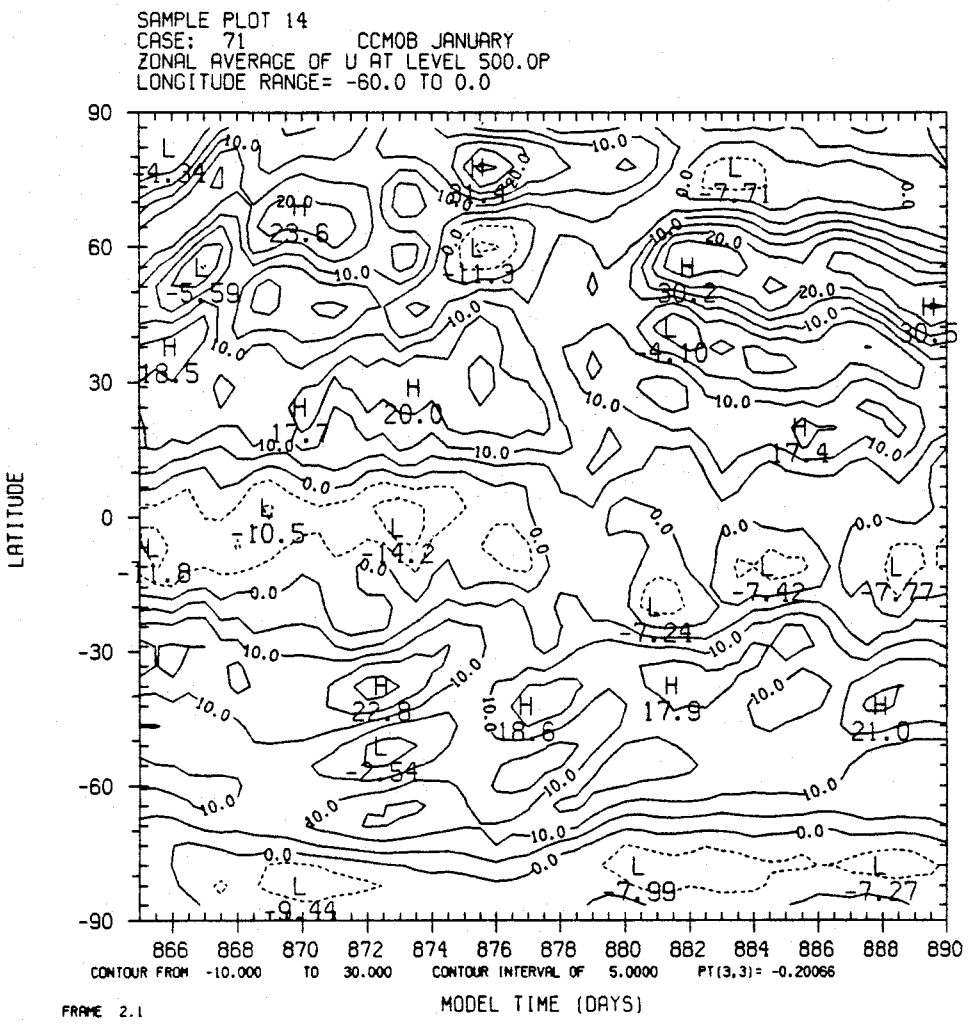

```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 12 - JAN. CLOUDS'
 TAPESA='X71007','X71008','X71009'
 DAYSA=90.,130.,0.5
 FIELDA1='TCLD'
 TSPALA1='TCLD','ALL','ALL',1000.,6:0.
C
 TITLEB='SAMPLE PLOT 12 - JUL. CLOUDS'
 TAPESB='X72007','X72008','X72009'
 DAYSB=90.,130.,0.5
 FIELDB1='TCLD'
 TSPALB1='TCLD','ALL','ALL',1000.,6:0.
 TSPFNP1='TCLD'
 ENDOFDATA
```

Fig. 12. Global averages of January and July total cloudiness plotted as two time series (*PTAL). Total cloudiness is computed from the multilevel cloud field on the input history tapes (*CDFL).




```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 13 - GEOPOTENTIAL HEIGHT'
 TAPESA='X71058','X71059','X71060'
 DAYSA=865.,890.,0.5
 FIELDA1='HTO'
 PRESSLE=500.
 TSPMCA1='HTO',500.,100.,0.,5300.
 LAVRNG=40.,60.
 ENDOFDATA
```

Fig. 13. Hovmoller plot of 500 millibar geopotential height averaged between 42.2N and 60.0N latitude (*PTMC) (*PCON). Geopotential height is computed from fields on the input history tapes (*CDFL).




```
C
C INPUT CONTROL PARAMETERS
C
 TITLEA='SAMPLE PLOT 14'
 TAPESA='X71058','X71059','X71060'
 DAYSA=865.,890.,0.5
 FIELDA1='U'
 PRESSLE=500.
 TSPZCA1='U',500.,5.,2:0.
 ZAVRNG=-60.,0.
 ENDOFDATA
```

Fig. 14. Zonally averaged 500 millibar zonal wind contoured as a time series for all latitudes (*PTZC). Note that the zonal averaging is performed over a limited range of longitudes (*ZONA).

Index for Topic Discussions

Index for ICP Keywords

Index for ICP Keywords (continued)